

Christian Schuster

Novel Schemes for Dynamic Monte Carlo Weighting

**IPP 2019-15
August 2019**



DEGREE PROJECT IN ENGINEERING PHYSICS,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2019

MASTER'S THESIS IN NUCLEAR-, PARTICLE-
AND ASTROPHYSICS, 30 CREDITS
GARCHING, GERMANY 2019



Novel Schemes for Dynamic Monte Carlo Weighting

CHRISTIAN SCHUSTER

**KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ENGINEERING SCIENCES**

**TECHNICAL UNIVERSITY OF MUNICH
DEPARTMENT OF PHYSICS**

MAX PLANCK INSTITUTE FOR PLASMA PHYSICS

Supervisors: Roberto Bilato (IPP)
Sibylle Günter (IPP/TUM)
Thomas Jonsson (KTH)
Gergely Papp (IPP)

Examiner at KTH: Ulrich Vogt
1st Examiner at TUM: Sibylle Günter
2nd Examiner at TUM: Thomas Jonsson

Collaborators: Seppo Sipilä (Aalto SCI)
Jari Varje (Aalto SCI)

Abstract

Ion Cyclotron Resonance Heating (ICRH) is a technique used for heating magnetically confined plasmas [1]. The goal of these plasma experiments is to achieve controlled thermonuclear fusion. Under the influence of ICRH the ion distribution function is no longer Maxwellian, it develops a high energy tail [2]. Most ions still have energies comparable to the thermal energy of the ion distribution. The energetic particles are however important for understanding the heating process [3], for the loss of particles [4, 5] and for the excitation of instabilities [6].

The distribution function evolves according to diffusion-advection equations [2, 7], which can generally be solved with Monte Carlo methods. For this purpose, one uses a discrete set of markers that evolve according to the Langevin equation. However only inaccurate information can be obtained about regions where the distribution function is small, because they are populated only sparsely by markers. An example for such a region is the high energy tail of the ion distribution function in a plasma.

This thesis proposes and investigates reweighting algorithms that permit us to have differently weighted markers for different regions of phase space. This allows better resolution of regions with small densities. When markers travel between regions we split them or remove some of them to avoid mixing of markers of different weight. This leads to correlated markers. These correlations can reduce the accuracy of the results.

Simple diffusion-advection equations serve as test cases for characterizing our new methods. Without reweighting, the square of the relative error of density calculations is inversely proportional to the density. We establish that with reweighting the squared relative error can be kept in the same order of magnitude even when the density varies over several orders of magnitude. After suggesting a general procedure for determining the reweighting parameters, we implement our algorithms in the code ASCOT-RFOF [8, 9], which calculates the ion distribution function under the influence of ICRH. As one result of our thesis, the computational time required for resolving the high energy tail at $E > 1.3 \text{ MeV}$ can be reduced by a factor of 100. Simultaneously the number of markers required by the simulation for developing the high energy tail is reduced by a factor of approximately 50. Our methods can be included in any Monte Carlo code for solving diffusion-advection equations, provided it can handle a fluctuating number of markers with nonuniform weight.

Kurzfassung

Die Ionen-Zyklotronresonanz-Heizung (ICRH) ist eine Technik zum Heizen von magnetisch eingeschlossenen Plasmen [1]. Das Ziel dieser Plasmaexperimente ist die kontrollierte thermonukleare Fusion. Mit ICRH ist der Hochenergiebereich der Ionenverteilungsfunktion des Plasmas stärker ausgeprägt als bei einer Maxwellverteilung. Die meisten Ionen besitzen immer noch Energien, welche mit der Ionentemperatur vergleichbar sind. Allerdings sind die hochenergetischen Ionen wichtig für die Heizung selbst [3], für Teilchenverluste [4, 5] und für die Anregung von Instabilitäten [6].

Die zeitliche Entwicklung der Verteilungsfunktion wird durch Diffusions-Advektions-Gleichungen vorgegeben [2, 7]. Diese können im Allgemeinen mit Monte Carlo Methoden gelöst werden. Hierfür werden diskrete Marker verwendet, deren Bewegung durch die Langevin-Gleichung vorgegeben wird. Wir erhalten aber nur ungenaue Informationen über Bereiche in denen die Verteilungsfunktion klein ist, da sich nur wenige Marker in ihnen befinden. Ein Beispiel für einen solchen Bereich ist der Hochenergieanteil der Ionenverteilung im Plasma.

In dieser Arbeit stellen wir Algorithmen zur Neugewichtung der Marker vor und untersuchen deren Eigenschaften. Sie erlauben uns, unterschiedlich gewichtete Marker für unterschiedliche Bereiche des Phasenraums zu verwenden. Dadurch lässt sich die Auflösung von Bereichen mit geringer Dichte verbessern. Wenn Marker den Bereich wechseln, werden sie entweder aufgeteilt oder einige von ihnen entfernt, um ein Mischen von Markern mit unterschiedlichem Gewicht zu vermeiden. Das wiederum führt zu korrelierten Markern. Diese Korrelationen können die Genauigkeit der Ergebnisse verringern.

Wir verwenden einfache Diffusions-Advektionsgleichungen um unsere Methoden zu charakterisieren. Ohne Neugewichtung ist das Quadrat des relativen Fehlers umgekehrt proportional zur Dichte. Wir zeigen, dass mit Neugewichtungen der quadratische relative Fehler der Dichte konstant gehalten werden kann, auch wenn sich die Dichte um mehrere Größenordnungen ändert. Zum Festlegen der Parameter der Neugewichtungen stellen wir eine allgemeine Vorgehensweise vor. Schlussendlich implementieren wir unsere Algorithmen in den Simulationscode ASCOT-RFOF [8, 9], welcher die Ionenverteilungsfunktion unter Einfluss von ICRH berechnet. Ein Ergebnis der Masterarbeit ist, dass die benötigte Rechenzeit für die Ionenverteilungsfunktion bei $E > 1.3 \text{ MeV}$ um einen Faktor von 100 verringert werden kann. Gleichzeitig wird die Anzahl von Markern, welche benötigt werden um einen ausgeprägten Hochenergiebereich zu entwickeln, um einen Faktor von etwa 50 verringert. Unsere Methoden können in jedem Monte Carlo Code, welcher Diffusions-Advektions-Gleichungen löst, verwendet werden. Voraussetzung ist jedoch, dass der Code eine fluktuierende Anzahl von Markern mit unterschiedlichen Gewichten erlaubt.

Sammanfattning

Joncyklotronresonansuppvärmning (ICRH) är en metod som används för att värma upp magnetiskt inneslutna plasman [1]. Med dessa plasmaexperiment vill man uppnå kontrollerad termonukleär fusion. När man använder ICRH upphör jonerna att vara Maxwellfördelade, eftersom fördelningsfunktionen får en högenergetisk svans. De flesta joner har dock en energi som är jämförbar med plasmats temperatur [2]. De högenergetiska partiklarna är viktiga för uppvärmningen [3], förluster av partiklar [4, 5] och excitation av instabiliteter [6].

Fördelningsfunktionen lyder under diffusion-advektions-ekvationer [2, 7] som i allmänhet kan lösas med Monte Carlo-metoder. För detta används en diskret uppsättning av markörer som följer Langevinekvationen. I regioner där fördelningsfunktionens värde är litet kan dock enbart inexact information erhållas, eftersom det finns få markörer där. Ett exempel är högenergidelen av jonernas fördelningsfunktion i ett plasma.

I den här avhandlingen föreslår och undersöker vi algoritmer för omviktning av markörer som medger markörer med olika vikter i olika regioner av fasrummet. Det tillåter bättre upplösning av regioner med låga tätheter. När markörer rör sig mellan regioner klyver vi respektive tar vi bort markörer för att förhindra att markörer med olika vikt blandas. Detta leder till att markörerna blir korrelerade vilket kan minska noggrannheten av metoden.

Vi använder enklare diffusion-advektions-ekvationer som testfall för att karakterisera våra nya metoder. Utan omviktning är kvadraten av den relativa osäkerheten av tätheten omvänt proportionell mot tätheten. Vi fastställer att vi med omviktningen kan behålla kvadraten av den relativa osäkerheten konstant, även om tätheten varierar med flera storleksordningar. Vi föreslår en allmän metod för att välja parameter för omviktning. Vi implementerar också våra algoritmer i koden ASCOT-RFOF [8, 9], som beräknar jonernas fördelningsfunktion i ett ICRH-scenario. Som ett resultat av detta arbete kan vi minska beräkningstiden för högenergidelen $E > 1.3 \text{ MeV}$ med faktor 100. Samtidigt minskas antalet markörer som krävs för att lösa upp den högenergetiska delen av fördelningsfunktionen med ca en faktor 50. Våra metoder kan användas med vilken Monte Carlo-kod som helst för att lösa diffusion-advektions-ekvationer, förutsatt att koden kan hantera ett fluktuerande antal markörer med olika vikt.

Acknowledgements

Over the course of the last year I was in Stockholm with the fusion plasma physics group at KTH, in Helsinki with the Fusion and Plasma Physics group at Aalto University, and in Garching at the Max Planck Institute for Plasma Physics. I was always welcomed cordially and received help whenever I needed it. There are some people who I especially would like to thank. The countless long discussions with Thomas Jonsson gave invaluable input for the development of the reweighting schemes. Gergely Papp's tutoring on how to actually write a thesis was essential. Thanks to Roberto Bilato, and again Thomas Jonsson and Gergely Papp, for sharing their expertise on plasma physics with me and for always generously dedicating me their time. And to Seppo Sipilä and Jari Varje for teaching me how to use and modify ASCOT. I would also like to thank Sibylle Günter for her help and feedback. Furthermore I will not forget all the interesting conversations with Björn Ljungberg that also improved my Swedish skills. I am grateful to my family and friends for always being there for me. Last but not least I owe Giulia a lot for all her support.

Contents

1. Introduction	1
1.1. Controlled Thermonuclear Fusion	1
1.2. Heating of Fusion Plasmas	1
1.3. Reweighting Methods in Other Fields	3
2. Theory	5
2.1. Coulomb Collisions	5
2.2. Particle-Wave Interactions	6
2.3. Evolution of the Particle Distribution Function	7
2.3.1. Ion Distribution Function with ICRH	9
2.3.1.1. Anisotropy	9
2.4. Monte Carlo Simulations	10
2.4.1. Monte Carlo Simulations for Integrals	11
2.4.2. Importance Sampling	12
2.4.3. Monte Carlo Simulations for PDEs	13
2.4.3.1. Wiener Process	13
2.4.3.2. Langevin Equation	14
2.5. Conclusion	15
3. Reweighting	17
3.1. Introduction	17
3.2. Classic Reweighting	17
3.2.1. Correlated Marker Positions	18
3.2.1.1. Decorrelation Length	18
3.2.2. Multiple Crossings	19
3.2.3. Non-Constant Total Weight	20
3.3. Alternative Schemes	20
3.3.1. Definition of Weight Regions	22
3.3.2. New Roulette Schemes	23
3.3.2.1. Correlated Roulette	23
3.3.2.2. Deterministic Roulette	25
3.3.2.3. Correlated Blue-Green Roulette	25
3.3.3. New Splitting Schemes	26
3.3.3.1. Correlated Random Splitting	26
3.3.3.2. Deterministic Splitting	27
3.3.4. Avoiding Unnecessary Crossings	27
3.3.4.1. Discrete Time Reweighting	27

3.3.4.2. Hysteresis	28
3.3.5. Summary of Novel Reweighting Schemes	28
3.4. A Simple Model for Analytical Calculations	28
3.5. Noisy Fluxes	29
3.5.1. Flux-Bin Model	31
3.6. Numerical Estimations of Errors	33
4. Performance of Novel Schemes	35
4.1. Overview	35
4.2. Wiener Process	36
4.2.1. Introduction	36
4.2.2. Required Number of Markers	36
4.2.3. Hysteresis Region in 1D	37
4.2.4. Numerical Investigation	39
4.2.4.1. Unchanged Mean	40
4.2.4.2. Roulette without further Measures	40
4.2.4.3. Comparison to Flux-Bin Model	44
4.2.4.4. Hysteresis and Time-Discrete Reweighting	44
4.2.4.5. Dependency of Decorrelation on Weight Ratio	49
4.2.4.6. Two Dimensional	52
4.3. Flow	54
4.3.1. Introduction	54
4.3.2. Analytical Calculations	54
4.3.2.1. Discontinuous Flow	54
4.3.2.2. Density Function	55
4.3.2.3. Mean and Variance of Binned Density Measurement	55
4.3.3. Single Boundary	56
4.3.4. Multiple Boundaries	58
4.3.4.1. Weight Ratio	58
4.3.4.2. Unchanged Mean	58
4.4. Parameter Choices	62
4.4.1. Weighting Function	62
4.4.1.1. Automatic Weighting Function	65
4.4.1.2. Mean Number of Markers	65
4.4.2. Hysteresis, Time between Reweighting, and Weight Steps	67
4.5. Initialization	68
4.6. Implementation in ASCOT-RFOF	70
4.6.1. Setup for Reweighting	71
4.6.2. Simulation Results	72
5. Conclusion	77
A. Decorrelation Time	81
A.1. Definitions and Setup	81

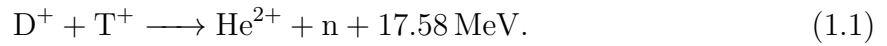
A.2. Mean Value	82
A.3. Limiting Cases	82
A.3.1. $t = 0$	82
A.3.2. $t \rightarrow \infty$	83
A.4. Variance for $t > 0$	83
A.5. Comparison to Simulation	84
B. Probability Density with the Deterministic Roulette	87
B.1. Definitions and Setup	87
B.2. Probabilities and Probability Distributions for Single Particles	87
B.3. Combined Probability Distribution	88
C. Influence of Correlated Deletion on Variance	91
D. Wiener Process with Multiple Boundaries	93
D.1. Influence of Boundary Position	93
D.2. Influence of Number of Boundaries	93

1. Introduction

1.1. Controlled Thermonuclear Fusion

Controlled nuclear fusion might be a practically inexhaustible source of clean energy [1]. The technical requirements are high: For two nuclei to fuse, the Coulomb potential has to be overcome, requiring kinetic energies much larger than thermal energies in nature on Earth. By heating the fuel some portion of the nuclei can be fast enough to undergo fusion. The required temperatures are in the order of 2×10^8 K, or 20 keV which is a more common unit in plasma physics ¹[11]. At this temperature the fuel is ionized and forms a plasma. Because of the high temperatures, confining and heating the fuel is a critical topic. Most research on fusion energy concentrates on magnetic confinement devices, where magnetic fields guide the plasma and confine the energy in the plasma [12, Chapter 10].

Because of a, compared to other possible isotope combinations, high reaction cross section at comparably low temperatures the most promising fuel is a mixture of deuterium and tritium. Their fusion reaction is



The charged helium nucleus (α -particle) is confined by the magnetic fields, while the neutron leaves the plasma unhindered and is stopped in the reactor wall. Here its kinetic energy is converted to heat. [12, Chapter 10]

1.2. Heating of Fusion Plasmas

In a fusion reactor energy losses from the plasma, that would reduce the temperature of the plasma, are planned to be balanced mostly by the α -particles originating in fusion reactions following equation (1.1) [1]. To reach these temperatures one however has to rely on other heating mechanisms. Also, a small portion of external heating in a reactor scenario is beneficial for controlling purposes [13]. For research purposes it is also necessary to heat plasmas externally.

One method of auxiliary heating is by injecting radio frequency (RF) waves into the reactor vessel. The waves may be absorbed by different mechanisms at different positions in the plasma [14]. One absorption mechanism exploits the periodic motion (gyromotion)

¹Temperatures in Kelvin can be related to energies with Boltzmann's constant: $E = k_B T$, $1 \text{ K} \approx 8.6 \times 10^{-5} \text{ eV}$ [10]

1. Introduction

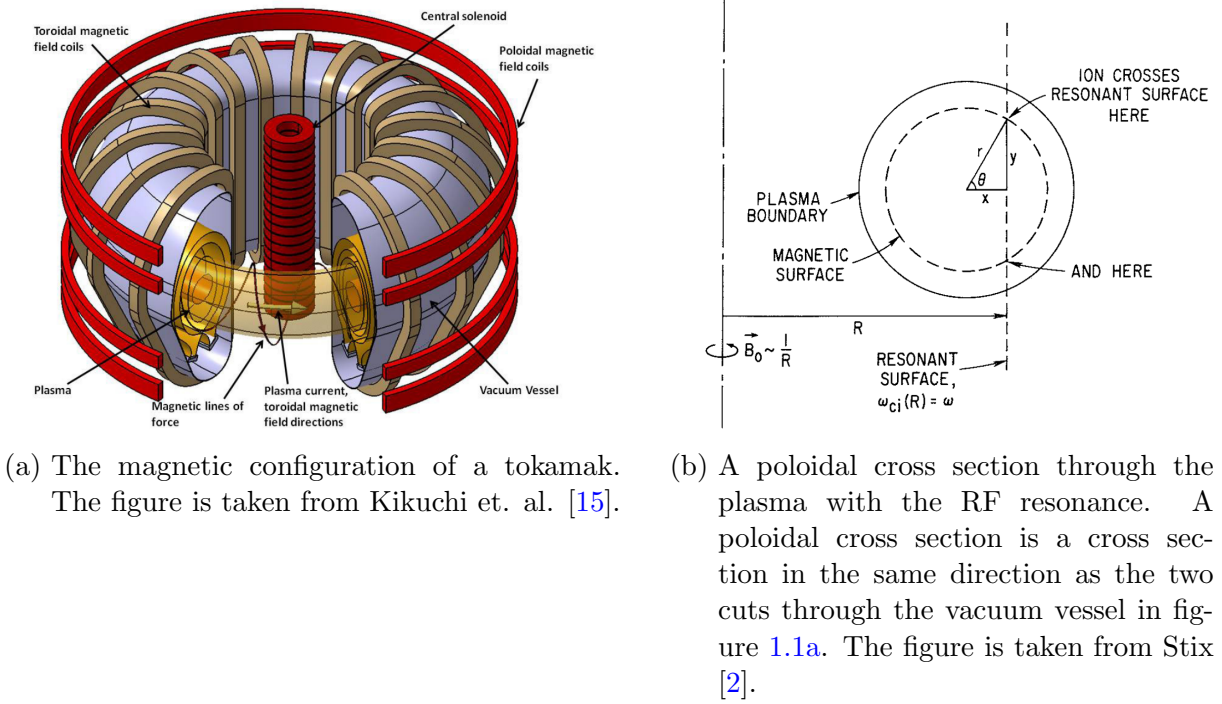


Figure 1.1.: A schematic overview of a tokamak and a radio frequency (RF) resonance.

of charged particles around the field lines of the confining magnetic field. The frequency of this gyration is proportional to the magnetic field strength [12]:

$$f_c = \frac{Bq}{2\pi m}. \quad (1.2)$$

q and m are the electric charge and mass of the gyrating particle. If this gyration of an ion is in resonance with the RF wave, i.e. if the wave frequency is an integer multiple of f_c , it may be accelerated to high energies [1]. Heating based on this resonance with the ion cyclotron frequency is called Ion Cyclotron Resonance Heating (ICRH).

The most promising designs for fusion reactors use a toroidal geometry to confine the plasma [1]. An example of the tokamak configuration, the most studied magnetic fusion device, is shown in figure 1.1a. We will now explain the function of the parts shown in figure 1.1a based on the book from Wesson [14]. The plasma is separated from the outside by the vacuum vessel. The toroidal field coils generate most of the magnetic field confining the plasma, while the poloidal field coils counteract forces that would propel the plasma towards the walls of the vacuum vessel. A current is induced in the plasma, as shown with the yellow arrow in figure 1.1a, because the plasma would be unstable otherwise. This current is induced by the central solenoid.

Inside the vacuum vessel the magnetic field B is inversely proportional to R , where R is the distance to the symmetry axis of the torus. The symmetry axis is the line the central solenoid wraps around in figure 1.1a. By choosing the frequency of the injected RF wave one can select where the power is absorbed [2, Chapter 17].

Unless otherwise indicated we base our explanation of the wave particle interaction on chapter 17 in the book of Stix [2]. In the plasma the magnetic field lines organize themselves in nested surfaces [1], shown in figure 1.1a in yellow. In 0th order the particle motion follows the field lines [12], resulting in particles confined to the magnetic surfaces. Figure 1.1b shows the radial position of the resonance and where the trajectory of an ion intersects this position. The magnetic field amplitude decreases from left to right. Ions in the plasma move on the magnetic surface shown as dashed circle. Whenever they are at the indicated distance to the symmetry axis they are in resonance with the RF field.

Depending on the phase of the gyration of the ion with respect to the wave field the ion is accelerated or decelerated by the wave. Neglecting a phenomena known as superadiabaticity the phase of an ion at one passing depends chaotically on the phase of the passing before. We can approximate the chaotic dependency by a stochastic one. The phases can be treated as uncorrelated, and the resonant heating as a stochastic process. Another source for stochastic behavior are collisions between particles. In total the equation of motion consists of a deterministic part due to the Lorentz force and the mean influence of collisions and heating, and two stochastic terms related to heating and collisions.

The effects described above lead to a ion distribution function that consists of two parts: a thermal Maxwellian at comparably low energies and a non-Maxwellian tail that declines much slower than a thermal tail would. Those ions in the tail are of great importance for heating and the plasma dynamics on one hand because they can drive instabilities in the plasma [5, 6] and are important for heating the plasma with ICRH [5]. On the other hand they can be used to study the effects of highly energetic fusion-born α -particles in a reactor [5]. As they are few in numbers it is difficult to resolve them with good statistics using Monte Carlo methods. Ideally we would like to use more markers of less weight in the tail region and fewer markers of higher weight in the bulk region. In this work we investigate how dynamic reweighting can be used to achieve this.

1.3. Reweighting Methods in Other Fields

The problem of resolving low probabilities with Monte Carlo methods arises also in fields other than plasma physics, and splitting methods for efficient solutions of this problem already exist [16] under the name 'Rare Event Sampling'. To my knowledge these methods require that the distribution function is already partially known to be able to set parameters, or that the markers can be advanced independent of each other [16]. This is problematic for applying these methods to ICRH because one does not know in advance how the high energy tail will develop. And due to nonlinear coupling, e.g. for determining the heating power and the amplitude of the wave field, all particles have to be advanced on short time scales compared to the time span of the simulation.

2. Theory

In this chapter we first introduce Coulomb collisions and wave-particle interaction. The wave-particle interaction leads to particle distributions that are not in local thermal equilibrium, we therefore need to describe the plasma with kinetic theory. After introducing kinetic theory we discuss how collisions and wave interactions lead to the distribution of ions in the plasma. The kinetic equations we obtain are of the diffusion-advection type for the solution of which we introduce Monte Carlo methods.

2.1. Coulomb Collisions

A fusion plasma mainly consists of electrons and fully ionized hydrogen isotopes. This means that when two particles collide they undergo Coulomb collisions, apart from a few nuclear interactions. The differential cross section for scattering by the Coulomb potential is the Rutherford cross section:

$$\frac{d\sigma}{d\Omega} = \left(\frac{Z_1 Z_2 e^2}{8\pi\epsilon_0 m v^2 \sin^2 \Theta/2} \right)^2. \quad (2.1)$$

The total cross section diverges because of the slow decrease of the Coulomb potential for large radii, but not in a plasma [12]. Charges of opposite polarity as the one creating the scattering potential accumulate around it, shielding the charge on long distance. This is known as **Debye shielding** with a characteristic length scale of λ_D [14]. In fusion plasmas much more particles undergo scattering at small angles than large angles. This is expressed in the **Coulom logarithm** $\ln\Lambda \approx 18$ which is defined as natural logarithm of the ratio of collisions that occur at an impact parameter of λ_D compared to collisions that lead to a deflection of 90° [12].

The cross section (2.1) decreases strongly for higher particle speeds. This is very different from neutral gases. Hard sphere potentials for example do not show a velocity dependency at all. We will discuss the effect Coulomb collisions have on ions based on Helander [7, Chapter 3]. For ion-ion collisions the strongly decreasing cross section together with a distribution function close to a Maxwellian results in a strongly decreasing collision operator for high velocities. For ion-electron collisions the collision operator depends only little on the ion velocity because the width of the electron distribution is typically larger than the speed of the ion.

2.2. Particle-Wave Interactions

As described in the introduction we can heat a fusion plasma by injecting electromagnetic waves into it. There are several possible mechanisms for absorption of these waves [14], we will focus on the absorption at the ion cyclotron resonance frequency.

Due to the Lorentz force charged particles, in our fusion plasma ions and electrons, move along a spiraling trajectory when a magnetic field is present. This motion can be separated into a component parallel to the magnetic field, which is mostly unaffected by the magnetic field, and a perpendicular, circular motion which is called gyration. As already stated in the introduction the frequency of gyration 1.2 is proportional to the magnetic field $B \propto 1/R$. As stated in the introduction the particle motion is 0th order confined to magnetic surfaces. A particle moving on a magnetic surface encounters in general various B fields because its distance to the symmetry axis (R) changes. This means their gyration frequency varies over time. By tuning the frequency of the incident wave one can select the position where they are in resonance with the gyromotion of the ions [12].

The electric field of the so-called X mode waves one uses for ICRH [1] rotates in the same plane as the gyrating particles [12]. At the resonance the electric field rotates with a frequency that is an integer multiple of the gyrofrequency. Discussing the absorption mechanism is easiest when the wave frequency is identical to the gyrofrequency and the direction of rotation is identical for the electric wave field and the particle. If the particle and the wave are then in phase the electric force acting on the particle is at all times parallel to the velocity component of the particle perpendicular to the magnetic field. If the particle and the wave are out of phase the electric force acting on the particle is at all times antiparallel to the velocity component of the particle perpendicular to the magnetic field. If particle and wave are in phase energy is transferred from the wave to the particle, the wave is damped and the particle accelerated. If they are out of phase energy is transferred from the particle to the wave, the wave is amplified and the particle decelerated. This means that if we send particles with same energy but different phase through the resonance they will exit with different energies.

Classical mechanics and electrodynamics are deterministic. The phase of the particle when passing through the resonance the second time is therefore already determined by its properties of the first passing. This dependency is however chaotic in nature, and we can employ an argument similar to molecular chaos: Like with collisions, also a deterministic process, it is a good approximation to assume the individual events as uncorrelated because of the mixing dynamics of the process.

Independent from the initial situation crossing the resonance leads to a broadening of the particle distribution. It is therefore a diffusive process in velocity space. When the distribution in velocity space broadens, the mean energy increases. By diffusing the particle distribution energy was transferred from the wave field to the particles.

2.3. Evolution of the Particle Distribution Function

In plasma physics, as in other fields of physics concerned with ensembles of many particles, exist at least two different ways of looking at the same situation: On one hand one considers the individual particles following their respective laws of motion, on the other one describes the particles as continuum.

In the particle description we have Newton's second law:

$$\frac{d^2}{dt^2} \mathbf{x}_i = \frac{1}{m_i} \mathbf{F}_{\text{ext}}(\mathbf{x}_i, \dot{\mathbf{x}}_i, t) + \mathbf{F}_{\text{interact}}(\{\mathbf{x}_j\}),$$

where t is the time parameter, \mathbf{x}_i is the position of the i -th particle, $\dot{\mathbf{x}}_i$ is the velocity of it, m_i the particle mass, \mathbf{F}_{ext} the external force acting on the particle, and $\mathbf{F}_{\text{interact}}(\{\mathbf{x}_j\})$ the force all particles exert on the i -th particle.

In the continuum description this becomes the Boltzmann equation for the distribution function:

$$\frac{\partial}{\partial t} f + \mathbf{v} \cdot \nabla_{\mathbf{v}} f + \mathbf{F}_{\text{macro}}(\mathbf{x}, \mathbf{v}, t) \cdot \nabla_{\mathbf{x}} f = \left(\frac{\partial f}{\partial t} \right)_{\text{col}}. \quad (2.2)$$

$f d^3x d^3v$ is the expectation value for the number of particles in the phase space region $d^3x d^3v$. In magnetized plasmas $\mathbf{F}_{\text{macro}}$ is primarily given by the Lorentz force $\mathbf{F}_{\text{macro}} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B})$, where \mathbf{E} and \mathbf{B} are the macroscopic fields consisting of external fields and collective contributions from the particles. The short scale interactions are covered by the nonlinear Boltzmann collision operator $\left(\frac{\partial f}{\partial t} \right)_{\text{col}}$. Evaluating the full Boltzmann collision operator is computationally expensive, and often not necessary. When most collisions result in small angle deflections, i.e. when the $\ln \Lambda \gg 1$, and f is close to a Maxwellian, we can derive a collision operator of the Fokker-Planck type [7]:

$$\left(\frac{\partial f}{\partial t} \right)_{\text{col}} = \nabla_{\mathbf{v}} \cdot \left[\mathbf{A} f + \nabla_{\mathbf{v}} \cdot \left(\vec{D} f \right) \right]. \quad (2.3)$$

\mathbf{A} is a vector-valued and \vec{D} is a tensor-valued coefficient. We recognize the first term as an advection, or friction, term and the second as a diffusion in velocity space.

In general particles at the spatial position \mathbf{x} can, and will, collide with all particles whose spatial distance is less than λ_D . But because the cross section decreases strongly with the relative velocity of the colliding particles (equation (2.1)) it is a good approximation to only consider particles that are also close to the test particle in velocity space, as long as the distribution is close to a Maxwellian [7]. As stated before we require the distribution function f to be close to a Maxwellian. At high energies the distribution function is therefore small, and there are only few particles between which collisions are probable. Therefore both \mathbf{A} and \vec{D} become smaller for higher energies.

2. Theory

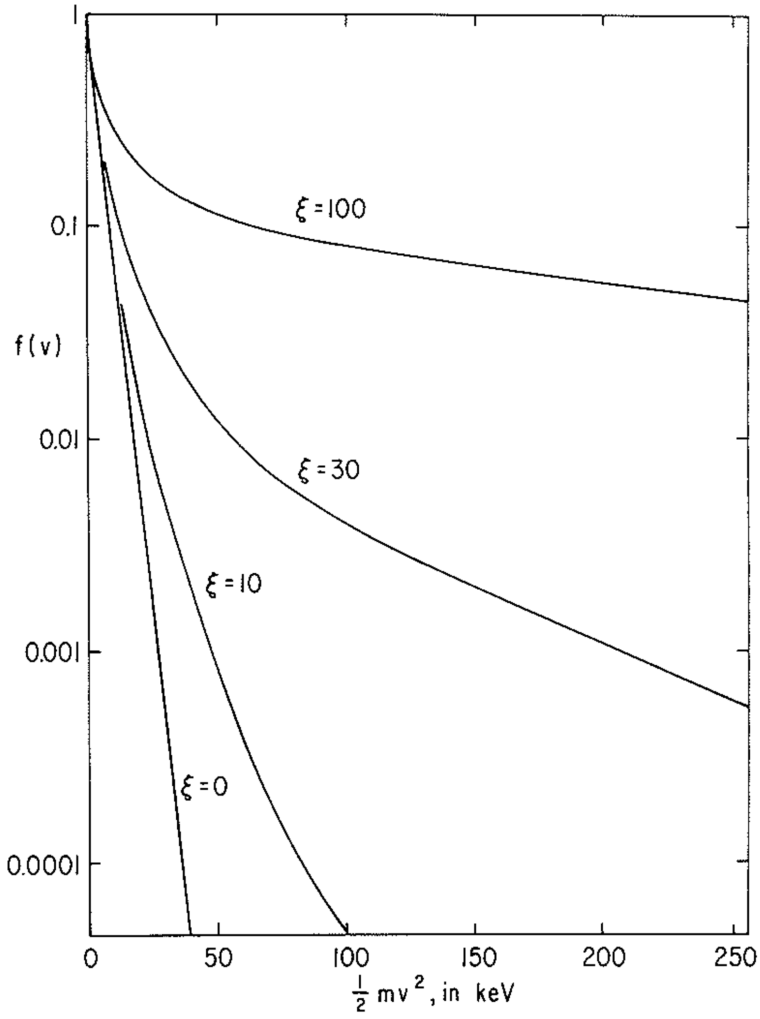


Figure 2.1.: Examples for deuterium distribution functions with different amounts of heating, parameterized by ξ . We see that heating leads to a strongly populated tail of the distribution, while leaving the thermal bulk mainly unaffected. The figure is taken and modified from Stix [17].

2.3.1. Ion Distribution Function with ICRH

The diffusive contribution of the wave-particle interactions is included with an additional term in the Boltzmann equation (2.2):

$$\frac{\partial f}{\partial t} f + \mathbf{v} \cdot \nabla_{\mathbf{v}} f + q(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_{\mathbf{x}} f = \left(\frac{\partial f}{\partial t} \right)_{\text{col}} + \left(\frac{\partial f}{\partial t} \right)_{\text{rf}}.$$

Here \mathbf{E} and \mathbf{B} are the macroscopic fields consisting of external fields and collective contributions from the particles. We saw that the collision operator (2.3) has a diffusive and an advective contribution. To deduct the shape of the ion distribution function with ICRH we first consider a thermal equilibrium without heating $\left(\frac{\partial f}{\partial t} \right)_{\text{rf}} = 0$. Then, the time derivative vanishes, and all other terms have to balance each other such that we have a Maxwellian distribution. For thermal ions collisions with other ions are the dominant contribution [1]. In section 2.1 we discussed that the contribution from the ion-ion collisions should be much weaker for energies much larger than thermal energies. The heating term on the other hand does not in general become weaker for more energetic particles, the RF resonance is localized in real-space but not in energy. We again consider the equilibrium, now with heating. For low energies the heating term is small compared to the advection due to the collisions, and the distribution does not differ considerably from the thermal Maxwellian. But for high energies the advection from collisions is small, and the diffusive nature of the heating term flattens the distribution. The only advection that works against this is the advection due to the ion-electron collisions. The effects from the RF interaction and the collisions leads to a high energy tail and a higher total energy in the plasma. Example distributions are shown in figure 2.1.

2.3.1.1. Anisotropy

ICRH not only changes the shape of the ion distribution function as function of energy, but also causes an anisotropy of the distribution function. As already stated in the introduction the particles in a tokamak plasma move, to 0th order, on magnetic surfaces. A magnetic surface is shown as dashed circle in figure 1.1b. We can separate the particles in the plasma into two categories [12]: Particles that move all around the magnetic surface, called passing particles. And particles that are confined to the side of the magnetic surface at large R , they cannot pass below a certain value of R and are called trapped particles. A trapped particle is characterized by a, compared to a passing particle, large ratio of the perpendicular to the parallel velocity component v_{\perp}/v_{\parallel} (with respect to the magnetic field). When such a trapped particle moves towards lower R it is eventually reflected. The wave field influences mainly the perpendicular velocity v_{\perp} . ICRH therefore increases the number of trapped particles. When the radial position where reflection occurs, called turning point, is at the RF resonance the trapped particle is affected more by the wave field than other particles are [2, 3, 18]. ICRH leads to an accumulation of ions in regions of phase space where the turning point is close to the RF resonance [3]. These ions have larger energies than ions in the thermal bulk [18].

2.4. Monte Carlo Simulations

Monte Carlo simulations are a very intuitive and for high dimensions efficient method to calculate properties of particle ensembles. The most direct calculation of the properties of a plasma would be to solve the coupled equations of motion of all particles in the plasma. In practice this is not possible because of the vast number of particles. But what if we use fewer particles, make them heavier and give them higher charges such that all the densities remain the same? Because the advection terms in equation (2.2) are linear all the flows in the plasma and the wave interactions should be treated correctly. But collisions between two heavy particles are very different from collisions between four ones with half the mass. In this section we will discuss how to go from the Boltzmann equation (2.2) to an equation of motion for the particles, circumventing the issue that we have to treat collisions differently for few large particles than for many small particles. This will however only work with Boltzmann equations of the diffusion-advection type.

Monte Carlo methods are able to solve general diffusion-advection equations, we will therefore summarize the \mathbf{x} and \mathbf{v} dependency of f as x . Also we are not restricted to even-numbered dimensions of x . To express the difference between the distribution function of particles in phase space and the more general solution of a diffusion-advection we denote the latter as ρ . With Monte Carlo simulations we do not solve for f in the Boltzmann equation directly. Instead we can evaluate derived quantities s defined as an integral over the distribution function f :

$$\langle s \rangle = \int_D s(x) \rho(x) dx. \quad (2.4)$$

Here D is the domain in which the diffusion-advection equation for f is defined. Examples for s are the total number of particles

$$s = 1,$$

the total energy

$$s = \frac{mv^2}{2}$$

and the average distribution function in a region of phase space $V \subset D$

$$s = \frac{1}{\text{vol}(V)} \mathbb{1}_V(x) = \frac{1}{\text{vol}(V)} \begin{cases} 1 & \text{if } x \in V \\ 0 & \text{if } x \notin V \end{cases}. \quad (2.5)$$

$\text{vol}(V) = \int_V dx$ is the (hyper)volume of the region.

By dividing the domain into small regions, named bins, we can recover the distribution function using equation (2.5).

We will see later in this chapter that solving diffusion-advection equations with Monte Carlo methods is an extension of solving integrals. We will therefore discuss solving integrals with Monte Carlo methods first.

2.4.1. Monte Carlo Simulations for Integrals

Monte Carlo integration is a technique that uses random numbers to evaluate integrals. The simplest approach is to randomly draw positions from a uniform distribution inside the domain D , with volume V , of the integral. Then evaluate the integrand at those points and take the average of the integrand values. We multiply this by the volume of the domain and get an estimate of the integral. This is easy to show, we will consider a single sampling point. s is the function we want to integrate, I_s is our estimate for the integral:

$$I_s(x) = s(x)V$$

$$p(x) = \frac{1}{V}.$$

p is the uniform probability distribution for the sampling point. Because the sampling point is a stochastic variable also our estimate is stochastic. It is straightforward to show that the expectation value, also called mean, of our estimate is as desired the integral of the function

$$\langle I_s \rangle = V \langle s(x) \rangle = V \frac{\int_D s(x)p(x) dx}{\int_D p(x) dx} = V \frac{1/V}{V/V} \int_D s(x) dx = \int_D s(x) dx$$

where D is the domain over which we are integrating, with the volume V . The standard deviation on the other hand can be interpreted as typical inaccuracy of the estimate.

When we use more sampling points the integral is defined with the average, where it is irrelevant whether we average the estimates or the function values at the different sampling points:

$$I_s(\{x_i\}) = \frac{\sum_i f(s_i)}{N} V = \frac{\sum_i I_s(x_i)}{N}.$$

The mean and the variance of two uncorrelated stochastic variables add linearly. For the mean this follows directly from the linearity property of the integral, and for the variance this is known as Bienaymé's formula [19]. Also, from the definition of the variance it is clear that

$$\text{Var}(aI_s) = a^2 \text{Var}(I_s), \quad (2.6)$$

where a is a constant. We then have

$$\langle I_s \rangle = \left\langle \frac{\sum_i I_s(x_i)}{N} \right\rangle = \frac{\sum_i \langle I_s(x_i) \rangle}{N} = N \frac{\langle I_s \rangle}{N} = \langle I_s \rangle$$

$$\text{Var}(I_s) = \text{Var}\left(\frac{\sum_i I_s(x_i)}{N}\right) = \frac{\sum_i \text{Var}(I_s(x_i))}{N^2} = \frac{N \text{Var}(I_s)}{N^2} = \frac{\text{Var}(I_s)}{N}. \quad (2.7)$$

The mean is unaffected by using more sampling points, and the variance is inversely proportional to the number of sampling points. This means that the standard deviation

2. Theory

$\sigma = \sqrt{\text{Var}(I_s)}$, or error, of the calculation scales as $\mathcal{O}(N^{-1/2})$. This scaling is completely independent of the number of dimensions the integral is defined, or how often we can differentiate the integrand. The usual integration methods like Simpson or Gauss quadrature do not have these properties, they are suited for low dimensional problems and smooth integrands. Monte Carlo integration is therefore better for high dimensional problems and 'rough' integrands.

In the next section we will improve the accuracy of the method presented in this section. This is well known as **Importance Sampling** and closely related to reweighting. In both cases one wants to increase the accuracy of the estimate by choosing different probability distributions for the sampling positions.

2.4.2. Importance Sampling

The integration method discussed in the previous section has one aspect that we can improve on: If we draw positions where the integrand is small in absolute value, it contributes only little to the result. This means we waste a function evaluation for an uninteresting position. If we would draw random numbers not from a uniform distribution but one that samples important regions more often the result could be more accurate. But of course we have to take some additional measures to not change the mean of the result. This method is called importance sampling [20].

We start with the integral we want to compute and insert unity:

$$\begin{aligned}\langle s \rangle &:= \int_D s(x) \, dx \\ &= \int_D \frac{s(x)}{p(x)} p(x) \, dx.\end{aligned}$$

We can interpret the last line as the mean of the function $f' = \frac{s(x)}{p(x)}$ that is sampled according to the probability density p . We can choose p freely as long as it is a valid probability density: It has to be normalized and non-negative. And p may only be 0 where s is zero as well. This is only for a single sampling position, but we already know that the error scales as $N^{-1/2}$.

The variance is then

$$\begin{aligned}\text{Var}(s) &= \frac{\text{Var}_{\text{single}}(s)}{N} \\ &= \frac{1}{N} \left[\int_D f'^2(x) p(x) \, dx - \left(\int_D f'(x) p(x) \, dx \right)^2 \right].\end{aligned}$$

Where we simply use the definition of the variance. If we choose $f'(x) = \text{const}$ the variance becomes 0. The probability distribution for the sampling positions should therefore be proportional to the function one wants to integrate.

2.4.3. Monte Carlo Simulations for PDEs

So far we have talked about how we can evaluate integrals using Monte Carlo methods. We can also use them to solve partial differential equations of the diffusion-advection type.

Equation (2.4) tells us that we want to evaluate integrals over the particle distribution function. In this equation we know $s(x)$, but not the particle distribution function $\rho(x)$. As we will discuss now it is possible to get sampling positions that are distributed according to ρ even though we do not know ρ yet.

We want to find a random experiment whose probability distribution for the outcome is the solution of the diffusion-advection equation we want to solve. As we will see shortly **stochastic differential equations (SDE)**, which we can consider as limit of a random walk, fulfill this requirement. We illustrate the connection between diffusion-advection equations and SDEs with an example, the **Wiener process**. We will also use the Wiener process as model system.

2.4.3.1. Wiener Process

We start with a random walk in one continuous x dimension and discrete time:

$$x_{n+1} = x_n + b\xi_n\sqrt{\Delta t}, \quad (2.8)$$

where ξ is a Gaussian random number with zero mean and a variance of 1. For every step ξ is independent of all previous and following ξ . We have

$$x_n = x_0 + b\sqrt{\Delta t} \sum_{i=0}^{n-1} \xi_i.$$

For large n the central limit theorem [21] dictates

$$x_n = x_0 + b\sqrt{n\Delta t}\xi.$$

$b\sqrt{n\Delta t}\xi = \sqrt{b^2n\Delta t}\xi$ is a normally distributed random number with zero mean and a standard deviation of $\sqrt{b^2n\Delta t}$. We can introduce the time t as

$$t = n\Delta t$$

When we take the limit $\Delta t \rightarrow 0$ equation (2.8) becomes the SDE

$$dx(t) = \xi(t)b\sqrt{dt} \quad (2.9)$$

with $\xi(t)$ a Gaussian distributed random number with 0 mean and a variance of 1. Simultaneously the probability distribution for the position remains a Gaussian and becomes

$$\rho(x, t) = \frac{1}{\sqrt{2\pi b^2 t}} \exp \left[-\frac{(x - x_0)^2}{2b^2 t} \right]. \quad (2.10)$$

2. Theory

Equation (2.10) also is a solution to the diffusion equation

$$\partial_t \rho = D \Delta \rho, \quad (2.11)$$

where Δ is the Laplace operator. We can therefore obtain sampling positions distributed according to equation (2.11) by solving equation (2.9), or its time-discrete counterpart equation (2.8), for every sampling position. The random numbers ξ have to be independent for each sampling position.

2.4.3.2. Langevin Equation

It is possible to find the SDE that yields a probability distribution equivalent to the solution of a diffusion-advection equation without solving the diffusion-advection equation beforehand. A derivation is for example given by Ichimaru [22, p. 294ff]. Particles following the stochastic differential equation, SDE, also known as Langevin equation,

$$d\mathbf{x} = \mathbf{a} dt + \vec{b} \cdot \boldsymbol{\xi}(t) \sqrt{dt}, \quad (2.12)$$

are distributed according to the solution of the Fokker-Plank equation

$$\frac{\partial f}{\partial t} = -\mathbf{a} \nabla_{\mathbf{x}} f + \nabla_{\mathbf{x}} \cdot \left(\nabla_{\mathbf{x}} \cdot \frac{\vec{b} \cdot \vec{b}}{2} f \right). \quad (2.13)$$

\vec{b} is a tensor describing the typical step width in different directions, and $\frac{\vec{b} \cdot \vec{b}}{2} = \vec{D}$ is the diffusion coefficient where $\vec{b} \cdot \vec{b}$ is the matrix product of \vec{b} with itself. In one dimension or when the diffusion is isotropic we can replace \vec{b} and \vec{D} with scalars. ξ are normally distributed random numbers with zero mean and a variance of 1. $\xi(t_1)$ and $\xi(t_2)$ are uncorrelated except if $t_1 = t_2$.

We now interpret the initial condition of the diffusion-advection equation (2.13) as probability distribution and draw sampling positions according to it. Then we advance the sampling positions according to the Langevin equation (2.12). The sampling positions are then distributed according to the solution of the diffusion-advection equation. We can interpret each sampling position as position of a hypothetical particle, called marker, whose equation of motion is the Langevin equation.

The equation for quantities derived from the distribution function (2.4)

$$\langle s \rangle = \int_D s(x) f(x) dx \quad (2.14)$$

then becomes

$$I_s = \frac{N_{\text{phys}}}{N} \sum_{i=1}^N s(x_i),$$

where N is the number of markers, I_s the estimate for $\langle s \rangle$ and

$$N_{\text{phys}} = \int_D f(x) dx$$

the number of physical particles. We define the weight of a marker as

$$w_i = \frac{N_{\text{phys}}}{N}$$

and finally arrive at

$$I_s = \sum_{i=1}^N s(x_i) w_i. \quad (2.15)$$

As presented here the weights of the markers w_i are identical, making the index redundant. But in general the weight can be different for each marker.

2.5. Conclusion

After discussing effects influencing the ion distribution function, collisions (section 2.1) and wave interactions (section 2.2), we showed examples for the ion distribution function in figure 2.1 and qualitatively discussed why a high energy tail develops. We can calculate the ion distribution function by solving the Vlasov equation with Monte Carlo methods. For this we have to solve many instances of the Langevin equation that corresponds to the Vlasov equation, as presented here this requires a collision operator that only consists of diffusion and advection terms.

When investigating the high energy tail the low energy part of the distribution is of little interest. Regardless of what exactly we want to calculate, for example the total energy or momentum in the tail, a histogram of the distribution function in the tail, or the number of energetic ions that leave the plasma, the function s in equation (2.4) will be large in the tail and small at low energies. As discussed in section 2.4.2 our sampling points, and therefore markers, should be concentrated in the high energy tail.

However, as we can see in figure 2.1 the low energy Maxwellian part of the distribution contains most of the physical particles. When we solve for the distribution function with Monte Carlo methods also most markers will be concentrated at low energies and high densities. The topic of this thesis is how we can modify the Monte Carlo method for diffusion-advection equations to use sampling positions that are not proportional to the density function. For this we will duplicate and delete markers when they propagate to different regions of phase space and adjust their weights.

3. Reweighting

3.1. Introduction

In section 2.5 we discussed that we have to modify the marker distribution we get from solving the SDE such that we can weight the markers to get less variance. To solve diffusion-advection equations we have to use markers following the Langevin equation due to construction of the method. The weighting function w depends on the position x , and the markers will move across different regions of configuration space. This means the marker has to change weight during runtime, but conservation of particle number, represented by weight, should not be violated. More generally, the distribution function itself should not be affected. This won't be achieved perfectly, but to good approximation: The mean is unchanged by the methods we propose, while fluctuations in normally conserved quantities are kept small and do not grow with time. We will do this by duplicating and deleting markers.

3.2. Classic Reweighting

There are different approaches on how to duplicate and delete markers to keep the weights of the markers identical to the weight prescribed by the weighting function $w(x)$. We will start with a simple algorithm, which we will call **classic reweighting**.

We specify discrete regions and a weight for each of those regions, $w(x)$ is a discrete function. We show in section 3.3.3 that having markers of different weight in the same region of phase space increases the variance compared to markers of equal weight. Therefore all the ratios of the weights of the different regions shall be integers, otherwise splitting one marker into several of equal weight is not trivial.

We denote the weight in region A with $w(A)$.

When a marker moves from region A to region B there are three possibilities:

$$w(A) = w(B)$$

We do not take further actions.

$$w(A) > w(B)$$

The weight of the marker will be reduced by a factor of $\frac{w(B)}{w(A)}$. To conserve weight we create $\frac{w(A)}{w(B)} - 1$ new identical markers at the same position. We call this operation 'split'.

$$w(A) < w(B)$$

The weight of the marker will be increased by a factor of $\frac{w(B)}{w(A)}$. As we only consider

3. Reweighting

one marker at a time we cannot merge it with other markers. We want to conserve the distribution function on average, so that when many markers cross this boundary the relative error will be small. We achieve this by deleting the marker with a probability of $1 - \frac{w(A)}{w(B)}$. We call this operation 'shoot'. The simple algorithm we use when shooting a marker is called Roulette, and we will introduce more advanced versions of the roulette with our new schemes.

This method has several shortcomings which we will discuss in the following sections. Later we will propose methods to avoid some of them.

3.2.1. Correlated Marker Positions

To understand this fundamental behavior let us consider a simplified situation, the Wiener process we defined in 2.4.3.1. One stationary solution is the constant probability distribution, and we initialize markers with this distribution. Evaluating a measure, for example the density using a histogram, will give fluctuating results. The standard deviation of the result depends among others on which quantity we are measuring. But it always scales as $\mathcal{O}(1/\sqrt{n})$ if the markers are uncorrelated, where n is the number of markers.

When one marker is split in two identical markers the markers are of course correlated. Thus, the $\mathcal{O}(1/\sqrt{n})$ scaling does not hold anymore. Because they are at the same position and have the same properties the splitting does not decrease the error. If the markers would follow a deterministic ODE they would have the same position forever and we would never gain accuracy. Instead, the stochastic term in the SDE causes the markers to become more and more independent as time progresses, restoring the $\mathcal{O}(1/\sqrt{n})$ scaling.

To get a better understanding how this decorrelation occurs we consider the following situation: Consider two markers that start at the same time at the same random position, after this both independently perform a random walk. The markers follow the Wiener process on a periodic domain. We then calculate the standard deviation of a histogram based density measurement. The calculation and the result is shown in the appendix A. In figure A.1 we see that the standard deviation of the result initially is the same as for a single marker. It then sinks to the level of two uncorrelated markers.

We refer to the time after which the error has dropped significantly as t_{decorr} .

3.2.1.1. Decorrelation Length

In the appendix A we found the decorrelation time t_{decorr} of two markers that were created at the same position. For the Wiener Process we know that $\Delta x \approx \sqrt{2D \Delta t}$. Using this relationship we can estimate the distance the two markers travelled before becoming decorrelated. One conclusion is that measurements that are taken far from the boundary are unaffected by the correlation. This can also be understood in a more direct manner. We already saw that the decorrelation time depends on what we measure. This is then also the case for the decorrelation length. We will continue our investigation with the histogram based density measurement defined in equation (2.5) as it is an important example. The density measurement counts markers that are in the respective bin. It does not distinguish where they are inside the bin. Therefore it does not matter if markers

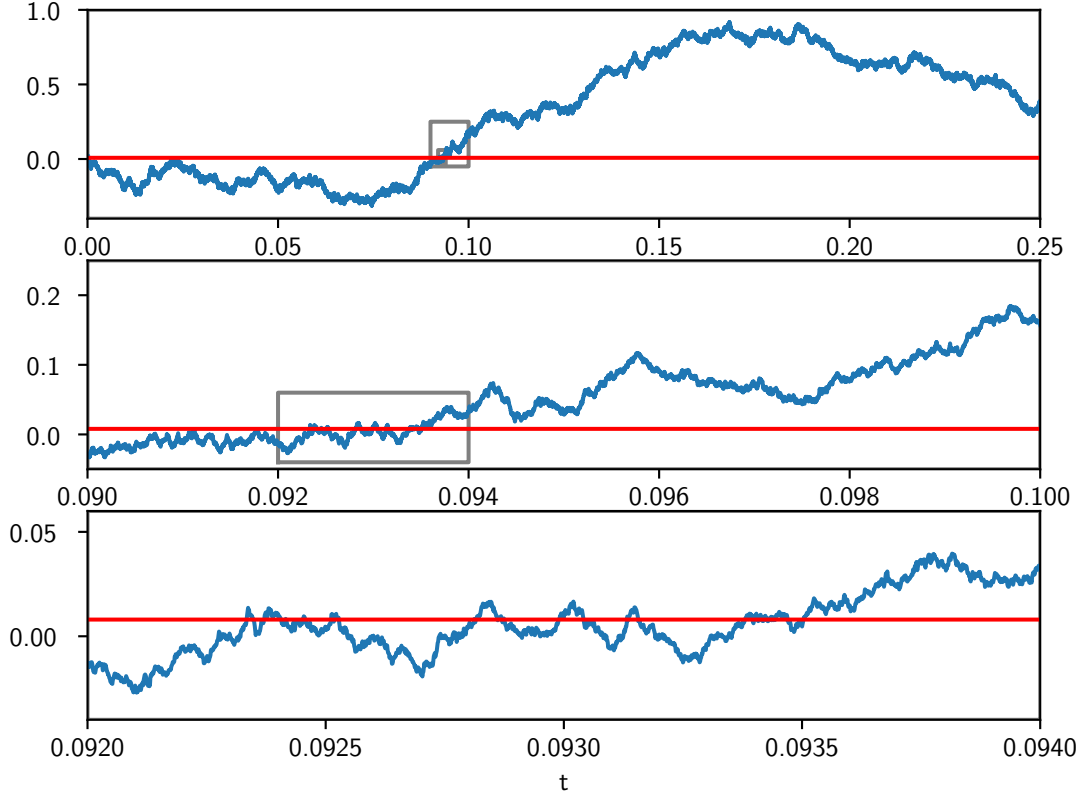


Figure 3.1.: A random walk in x as a function of the time t in three different zoom levels. The grey boxes indicate the plot ranges of the smaller zoom levels. In the first plot it appears that the trajectory crosses the red line only once, but if we zoom in far enough we see that there are in fact many crossings.

that start at the same position are at the same position, or if they are at opposite ends of the bin. The markers have to leave the bin to stop being correlated according to the binned density measurement. For this they have to travel a distance of the order of the bin size l . The decorrelation length l_{decorr} is therefore approximately l .

We can expect that a measurement that was taken several decorrelation lengths away from a boundary is comparably unaffected by the correlation. This distance is longer if the bins used are larger.

3.2.2. Multiple Crossings

The correlated marker positions have one further effect that can be mitigated. We illustrate it with an example:

We start with 100 markers just next to a boundary, on the high-weight side. Let all of them move to the other side, and each marker is split into two. We will refer to two

3. Reweighting

markers that started at the same position as twins. Next assume they all return to the high-weight side. When crossing to the high-weight side approximately half of the markers are deleted. But as we flip a coin for each marker separately it is not unlikely that both twins are deleted, or that both survive.

This means that while we might still have 100 markers, just as in the beginning, some of them are correlated. Therefore we reduced the accuracy by splitting and deleting. If we repeat this we even get correlations between four markers, then eight, and so on. As particles undergoing a Wiener process exhibit a typical zig-zag trajectory such repeated border crossings are likely. As we see in figure 3.1 a typical random walk would cross our boundary often. In fact, the number of crossings diverges if we continue to zoom in because the Wiener process is self-similar [23].

3.2.3. Non-Constant Total Weight

There is an additional problem with multiple crossings, one which cannot be solved by decorrelation. We revisit the example from section 3.2.2. After we split and shoot the initial 100 markers we will not have exactly 100 markers. Instead the number of markers will be a stochastic variable, with a finite variance. When the markers cross the boundary again, even after a long time, there will be some additional uncertainty about the number of markers afterwards.

We see that the number of markers in a simulation will deviate more and more from the initial value the longer we simulate. This also applies to the total weight and mass. This is clearly an undesired behavior.

3.3. Alternative Schemes

As we discussed in the last section the classic reweighting has shortcomings that should be improved. Possible approaches include merging particles instead of deleting them, keeping the first three moments of the distribution constant. This however can lead to artificial thermalization of the distribution [24]. Another approach is the reweighting in the energy dimension in the orbit following code ASCOT 3.5. It was implemented by A. Salmi, and the markers are not required to have the same weight in the same region of phase space. It uses uncorrelated random number for the roulette, and then readjusts the weights to achieve energy or momentum conservation. This approach only allows conservation of a single moment.

Instead of using a probabilistic approach for the roulette one could merge markers that moved into the high weight region. This can however be computationally expensive and lead to a modified mean compared to the case without reweighting [24].

In this section we propose new schemes for reweighting. We describe our new methods and discuss advantages and disadvantages. An overview is shown in table 3.1. The algorithms we are proposing in this section do not alter the mean result of the simulation. As a trade off mass and all other moments of the distribution are not conserved exactly, but the fluctuations around the mean do not increase with time and are kept small.

Shooting schemes		
	advantages	disadvantages
Correlated Roulette 3.3.2.1	very accurate conservation of the particle number fluctuation- <i>decreasing</i> correlation effect in 1D	in multiple dimensions strong local fluctuations
Deterministic Roulette 3.3.2.2	avoids correlations between markers	larger global fluctuations of the particle number requires storing an additional parameter for each marker
Splitting schemes		
	advantages	disadvantages
Classic Splitting 3.2	does not introduce additional fluctuations	only works for integer weight ratios
Correlated Splitting 3.3.3.1	conserves the total weight more accurately than the deterministic splitting	stronger local fluctuations
Deterministic Splitting 3.3.3.2	independent of number of dimensions	larger global fluctuations of the particle number requires storing an additional parameter for each marker for each boundary
Avoiding frequent crossings		
	advantages	disadvantages
Hysteresis 3.3.4.2	independent of diffusion or advection coefficients	
Discrete Time Reweighting 3.3.4.1	waiting between reweightings reduces the overhead compared to simulations with constant weight	requires knowledge of the local diffusion coefficient to adjust the time between reweightings

Table 3.1.: Overview of the schemes proposed in section 3.3. *Classic Splitting* is not new, but included for comparison.

3. Reweighting

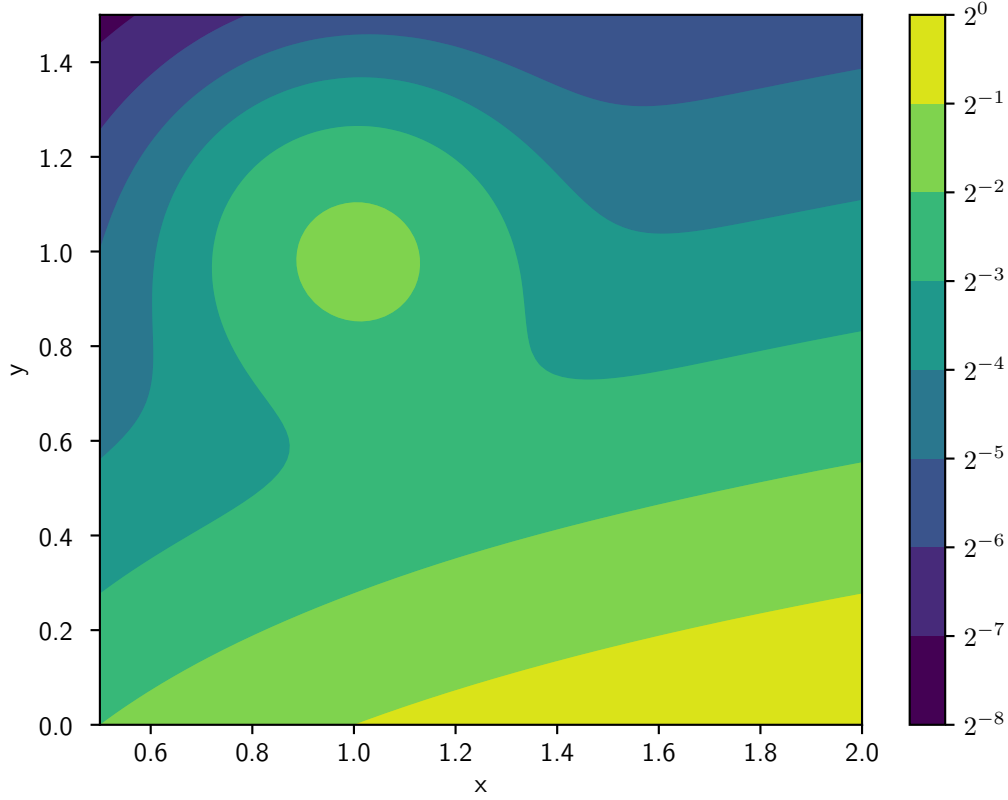


Figure 3.2.: We define a continuous weighting function $w_c = \frac{x}{2} \exp[-2.5y] + \frac{1}{4} \exp\left[-\frac{(x-1)^2 + (y-1)^2}{0.08}\right]$ and define discrete weight regions that differ by a factor of $1/2$. We show the weight regions.

3.3.1. Definition of Weight Regions

Both the classic reweighting and the new schemes we will propose rely on a phase space that is separated into discrete regions. All markers in the same regions have the same weight. It should be easy to compute in which region a marker is as we have to do this often and should not use a lot of computation time. We propose the following procedure:

First we define a continuous weight function $w_c(x)$ which we will use to define our discrete weight regions. We define n weight levels, for example

$$\begin{aligned} W_i &= 2^{-(n-1)} \\ W_{n+1} &= 0 \end{aligned}$$

with $i = 1, \dots, n$. Then, we define the i -th region as set of all points x where

$$W_{i+1} < w_c(x) \leq W_i.$$

The weight of the markers as a function of x is *not* w_c , but the smallest weight W_i that is larger than or equal to $w_c(x)$. The individual regions are not necessarily connected, in general islands may exist. Such islands could for example arise when choosing w_c proportional to the density, what is favorable as is discussed in section 4.4.1, and when the simulation yields a so-called bump-on-tail distribution function [25]. A fictional example for w_c leading to islands and the resulting weight regions are shown in figure 3.2. We compute w_c at the position of the marker to determine what region a marker is in and find the appropriate weight level. Because w_c is continuous, the weight regions are ordered what is important for keeping track where markers were created. If a marker travels from region i to the region j the marker passes through all regions whose index lies in between i and j , because the step-size is distributed according to a Gaussian such large jumps always have a finite possibility.

3.3.2. New Roulette Schemes

We propose three new roulette schemes that mitigate some of the negative effects discussed in section 3.2. Developing them and investigating the behavior of these schemes is the main task of this thesis. As we will see in section 4.3.4 they can indeed reduce the fluctuations of results significantly.

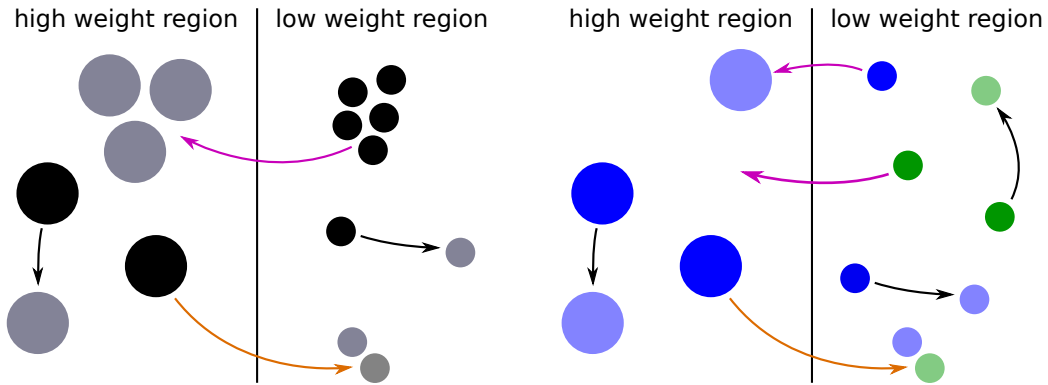
3.3.2.1. Correlated Roulette

The first new scheme is called the **correlated roulette**. It avoids the variance in total weight as discussed in section 3.2.3 by using correlated random numbers.

For example, if we have a weight ratio of 2:1 between the high weight region and the low weight region we have to delete 50 % of the markers coming from the low weight region. The first marker that enters the high weight region is deleted with 50 % probability, just as in the classical roulette. The next marker is deleted when the first one survived, and survives when the first one was deleted. By following this procedure the total weight can only deviate from the mean by the weight of a single low-weight marker. If we have several boundaries this deviation can increase, but will remain low compared to the typical marker number and does not increase with time: For every boundary between regions we can have at most one marker from the lower weight region too much or too little. Because the weight per marker is inversely proportional to the number of markers the fluctuations of the total weight are $\mathcal{O}(1/N)$ compared to the typical scaling of $\mathcal{O}(1/\sqrt{N})$ for Monte Carlo Methods. In figure 3.3a we show a sketch visualizing the correlated roulette.

The correlated roulette does not avoid correlations between many markers as described in section 3.2.2 in more than one dimension. In one dimension there is a beneficial correlation effect as we will discuss in section 4.2.4.2.

3. Reweighting



- (a) An illustration of the *correlated roulette*. When the markers move inside one weight region (black arrows) they are not altered. When a marker moves from the high to the low weight region (golden arrow) it is split into two smaller markers. When markers travel from the low to the high weight region (purple arrow) some are deleted, the weight of the remaining ones is doubled. Because we use correlated random numbers to decide whether to delete a marker, the remaining markers are at most 'half' a marker too much or too little.
- (b) An illustration of the *deterministic roulette*. When the markers move inside one weight region (black arrows) they are not altered. In the high weight region only blue markers exist. When a marker moves from the high to the low weight region (golden arrow) it is split into two smaller markers, one of which is blue and one is green. When a green marker travels from the low to the high weight region (purple arrow) it is deleted. When a blue marker travels from the low to the high weight region (purple arrow) its weight is doubled.

Figure 3.3.: Sketches illustrating the *correlated* and *deterministic* roulettes. We show markers as filled circles. The dark ones circles are the initial markers, the pale markers are the new markers. The arrows show how the markers move. The weight step between the high and low weight regions is 1 : 2.

We can generalize the correlated roulette from the weight ratio 1 : 2 in the example to fractional weight ratios:

$$p := 1 - \frac{w(A)}{w(B)} = \frac{k}{l},$$

with k and l integer and $w(B) > w(A)$. We take an urn and fill it with l balls. Of these k are black, the other $l - k$ are white. Whenever a marker undergoes the roulette we draw a ball from the urn. If it is black we delete the marker, otherwise not. We do not put the ball back into the urn. When it is empty we fill the urn anew.

3.3.2.2. Deterministic Roulette

The deterministic roulette, which we also call the blue-green roulette, on the other hand relies on an additional property that we save for each marker. When splitting a marker we mark one as 'blue', for example the original marker. The newly created marker is marked as 'green'. When a marker travels to the high weight region we let it survive only when it is the 'blue', otherwise we delete it. This means that when we duplicate a marker and delete its twin afterwards, it is as if it was never split in the first place. It therefore cannot be correlated to other survivors. We call this method the **deterministic roulette**, as we do not use a random number to decide whether to delete. In figure 3.3b we show a sketch visualizing the deterministic roulette.

With the deterministic roulette the fluctuations of the total weight do not increase with time. There still are variations in the number of markers and the total weight: Two markers that were created simultaneously will probably not return to the high weight region at the same time. In the time period where only one of the two markers returned we will have either too little or too much total weight. Because we have many markers this is always the case for some of them. But these variations do not accumulate over time because those variations vanish as soon as all markers have returned.

In a simulation one usually has more than one boundaries. Instead of keeping track of the blue-green property for every boundary it is sufficient to store in which region the marker was created.

3.3.2.3. Correlated Blue-Green Roulette

We now have two schemes:

Blue-Green Roulette

We avoid unnecessary correlations between markers and simultaneously limit variations in total weight.

Correlated Random Numbers

We enforce the conservation of total weight by correlating the survival and deletion between markers.

Ideally we would like to have both positive traits. One could try to not immediately decide what marker survives when splitting, but only later when the first of those markers

3. Reweighting

crosses the boundary. But that means that the markers cannot be advanced independent of each other. This is a serious constrain for the implementation and parallelization of the algorithm.

Therefore for now we will not consider such a hybrid scheme.

3.3.3. New Splitting Schemes

Until now we have restricted ourselves to integer weight ratios between regions. If we do not have this cumbersome restrain we would have to create markers of different weight to conserve the total weight. Having markers of different weight in the same region of phase space increases the variance of the result: The variance of our result from the simulation, given by equation (2.15), are

$$\begin{aligned}\text{Var}(I_s) &= \text{Var}\left(\sum_{i=1}^N s(x_i)w_i\right) = \sum_{i=1}^N \text{Var}(s(x_i)w_i) = \sum_{i=1}^N \text{Var}(s(x_i)) w_i^2 \\ &= \text{Var}(s(x)) \sum_{i=1}^N w_i^2\end{aligned}$$

We use Bienaymé's formula [19] and equation (2.6). Minimizing the variance as a function of w_i while maintaining $\sum_{i=1}^N w_i = \text{const}$ with a Lagrange multiplier quickly shows that choosing all w_i identical is optimal. In our derivation we assumed that w_i are constants and do not depend on x_i . Therefore we cannot infer that reweighting itself reduces the accuracy of results even though it depends on nonequal weights.

We will now introduce two different options we found to use fractional instead of integer ratios between the weights of regions that keep the weights in each region equal.

3.3.3.1. Correlated Random Splitting

We can use a probabilistic approach for splitting, similar to what we do when shooting. As uncorrelated random numbers will lead to fluctuations we use correlated random numbers for splitting. In section 3.3.2.1 we described how we can draw random numbers such that we do not introduce growing fluctuations.

Correlated random splitting keeps fluctuations of the first moment of the distribution from growing with time. The difference between the number of markers prescribed by the fractional weight ratio and the number of markers actually in the simulation is independent of the number of markers. The error due to imperfect splitting is only $\mathcal{O}(1/N)$, where N is the number of markers in the simulation, because the weight of the markers is inversely proportional to N .

In more than one dimensions we can however imagine how we still can get fluctuations: Let the domain be a square of side length 1 with the coordinates x and y . The boundary shall be at $x = 0.5$, with small x forming the high weight region. We have uniformly distributed markers in the high weight region. After they move across the boundary we know that the total weight will be almost exactly conserved because of the correlated

random numbers. But it can happen that more new markers were created at high y values than at low y values, or the other way around. This effect is more pronounced if the domain has many dimensions and is large, for example compared to the measurement bin size. This effect is closely related to the correlations between multiple markers described in 3.2.2.

3.3.3.2. Deterministic Splitting

When we consider longer time scales, every marker will cross the boundaries in the system several times. This means that we can use a correlated random number generator for each marker for each boundary individually. Each of those correlated random number generators introduces some variance in total weight, as discussed in 3.3.2.1. Therefore the total weight is not as accurately conserved as when using a single generator. But when using many generators the fluctuations of the distribution function are independent of the number of dimensions or the size of the domain. It might therefore be better than the correlated random splitting in some situations. But the differences between the splitting schemes appear to be only minor when comparing them in section 4.3.4.

When implementing the scheme having that many correlated random number generators might be undesirable. The weight ratio between two regions shall be $w = n/d$, with n and d integer and $n > d$. Instead of many random number generators we use a fixed sequence of Boolean numbers, b , with n entries. $n - d$ entries are 1, d are 0. Every marker stores which element of the sequence b to use next. When a marker crosses the respective boundary from the high to the low weight region, we reduce its weight, and if the current element of the sequence b is 1 we also duplicate it. Afterwards we store the position of the next element of b , if we reached the end of the sequence we start from the beginning.

3.3.4. Avoiding Unnecessary Crossings

As we saw when we discussed the classic reweighting in section 3.2 crossings of markers between regions have negative effects. And because of the zig-zag structure of the Wiener process, as shown in figure 3.1, it is probable to have many consecutive crossings. In those cases it would be better to reweigh only once or not at all instead of after each individual crossing. We propose two adaptations to the schemes that implement this.

3.3.4.1. Discrete Time Reweighting

Until now we considered situations where the markers are instantaneously split or shot when crossing a boundary. As we are discussing numerical algorithms this is not viable: We can only split between time steps of finite length. And checking in which region the marker is currently located and splitting or shooting takes some computational time. This means that we always have to wait a little before reweighting.

By waiting between reweighting we neglect all the crossings that happened in the meantime and were cancelled by other crossings. We could for example only reweight after waiting for one decorrelation time t_{decorr} .

3. Reweighting

In between reweightings the markers of the different regions mix. The typical length over which the markers diffuse in between reweightings depends, other than l_{decorr} , on the diffusion coefficient. Therefore it might be hard to optimize for several boundaries in regions with different diffusion coefficients if we reweight all markers at the same time.

3.3.4.2. Hysteresis

Another way of avoiding unnecessary crossings is to introduce a hysteresis region of width h . This means that a marker has to cross the boundary and advance beyond a distance of $h/2$ to the boundary before being split or shot. As long as we do not reweight the marker it keeps its old weight, its weight is therefore not only a function of position anymore. Inside the hysteresis region we have markers of both high and low weight. When $h \gtrsim l$ this means that the markers are decorrelated when they would be split again because they had to travel far. It is however still uncertain how large the hysteresis has to be to yield good results, what is the exact meaning of ' \gtrsim '? We will determine this in simulations of simple test cases.

3.3.5. Summary of Novel Reweighting Schemes

We define regions in phase space, each with markers of a specific weight. When a marker changes the region it is in we either split it up or delete it, depending on whether the new region has higher or lower weight than the original one.

After splitting the marker positions are correlated and need some time to decorrelate.

We propose two schemes to avoid growing fluctuations that arise when deleting markers at random. We use correlated random numbers or the deterministic roulette to determine whether a marker should be deleted, this is not necessary when the weight ratio between regions is an integer.

3.4. A Simple Model for Analytical Calculations

To compare the schemes and judge their performance we would like to have analytical estimates for the errors. This requires approximations: In the appendix B we show a calculation for the hysteresis region for the deterministic roulette without those simplifications. This calculation is cumbersome and does not lead to a usable result. We therefore build a simple model that we will use for some estimates.

In our schemes the markers do not propagate independent from each other. For example we might create a marker when another marker passes from a high weight to a low weight region. In our simplified model we will nevertheless treat the markers as completely independent and equally distributed:

$$p_{\text{combined}}(x_1, x_2, \dots, x_N) = \prod_{i=1}^N p(x_i),$$

where p_{combined} is the combined probability distribution for all markers and x_i the coordinates of the respective marker. $p(x_i)$ the probability distribution for the i -th marker: the probability of finding the i -th marker in the phase space volume dx at the position x is $p(x_i)dd$. Because we model the markers as independent we know that the relative error scales as $\mathcal{O}(N^{-1/2})$, where N is the mean number of markers. We will therefore calculate the error for only a single marker and scale the error accordingly.

The probability density for a marker without reweighting is proportional to the phase space density. If we reweight the markers we introduce a probability density in one additional dimension: we have the spatial dependency as before, and one additional dimension for the weight. We consider schemes with discrete weights, therefore this additional dimension will be discrete as well.

This probability density for the position and weight of a marker, p , depends on the process. Once we found p we can compute the mean and variance of a measurement. We will mainly be investigating the binned density measurement ρ_{bin} :

$$\rho_{\text{bin}}(x, w) = \frac{\mathbb{1}_{\text{bin}}(x)}{V}w,$$

where x and w are the position and weight of the marker, and V is the volume of the bin. $\mathbb{1}_{\text{bin}}$ is the box function defined in (2.5). It is 1 when x is inside the bin and 0 otherwise. We divide by the (hyper)volume of the bin V to get the density.

We denote the mean with $\langle \cdot \rangle$:

$$\begin{aligned}\langle \rho_{\text{bin}} \rangle &= \sum_w \int_D \frac{\mathbb{1}_{\text{bin}}(x)}{V} w p(x, w) dx \\ \langle \rho_{\text{bin}}^2 \rangle &= \sum_w \int_D \frac{\mathbb{1}_{\text{bin}}^2(x)}{V^2} w^2 p(x, w) dx.\end{aligned}\tag{3.1}$$

D is the domain of the process. Observe that $\mathbb{1}_{\text{bin}}^2(x) = \mathbb{1}_{\text{bin}}(x)$. The variance and the relative error are defined as:

$$\begin{aligned}\text{Var}(\rho_{\text{bin}}) &= \langle \rho_{\text{bin}}^2 \rangle - \langle \rho_{\text{bin}} \rangle^2 \\ E_{\text{rel}} &= \sqrt{\text{Var}(\rho_{\text{bin}}) / \langle \rho_{\text{bin}} \rangle} = \sqrt{\langle \rho_{\text{bin}}^2 \rangle / \langle \rho_{\text{bin}} \rangle^2 - 1}.\end{aligned}$$

As we will test in chapter 4 our schemes keep the expectation value of the distribution function unaffected. This means if we have markers with less weight we have more of them to keep the total weight per volume unaffected.

3.5. Noisy Fluxes

We discussed in 3.2.1 that correlations between markers can reduce the accuracy of the simulation, and that the effect of these correlations decay on the length scale of l_{decorr} . There is yet another effect introduced by having regions with different weights and accuracy. The regions exchange particles, each region has an influx and outflux of particles. If

3. Reweighting

a region has large fluctuations in the number of markers the flux of markers from this region will also fluctuate strongly. The number of markers in the region where this strongly fluctuating flux flows into will also fluctuate more. On the other hand, a region with small fluctuations might decrease the fluctuations of adjacent regions. The relationship between the fluctuations in a region and the fluctuations of the flux depend on the used splitting or shooting scheme because reweighting introduces additional fluctuations.

This effect already exists without reweighting. Consider the Wiener process introduced in 2.4.3.1 on the one dimensional domain $[0, L]$, where L is the length of the domain. The probability distribution $p = 1/L$ for the Monte Carlo markers for the stationary solution is uniform. We consider the binned density measurement

$$s(x) = \frac{L}{l} \mathbb{1}_{\text{bin}}(x),$$

where l is the width of the bin and $\mathbb{1}_{\text{bin}}$ the box function. For the mean and variance of the measurement of s with a single marker we get

$$\begin{aligned} \langle s \rangle &= \int_0^L p(x) s(x) dx = \int_l \frac{1}{L} \frac{L}{l} dx = 1 \\ \langle s^2 \rangle &= \int_0^L p(x) s^2(x) dx = \int_l \frac{1}{L} \frac{L^2}{l^2} dx = \frac{L}{l} \\ \text{Var}_{\text{single}}(f) &= \langle s^2 \rangle - \langle s \rangle^2 = \frac{L}{l} - 1. \end{aligned}$$

When we use N markers instead the variance becomes

$$\text{Var}(s) = \frac{1}{N} \left(\frac{L}{l} - 1 \right) = \frac{L}{N} \left(\frac{1}{l} - \frac{1}{L} \right). \quad (3.2)$$

N/L is the average number of markers per unit length. When $L = l$ all markers lie within the bin, therefore the measurement does not depend on the specific realization and the variance is 0. When $L \gg l$ we get $\text{Var}(s) = L/Nl$, what is the inverse of the expected number of markers in the bin. In this case we can therefore directly use the number of markers in the bin as estimate for the uncertainty of the result.

Consider a domain that is large compared to the bin size, $L \gg l$. Also imagine a region inside this domain with a size of e.g. $5l$. The number of markers inside this region fluctuates because there are noisy fluxes into and outwards of the region. If the whole domain has however a size of $L = 5l$ the number of particles inside the region of size $5l$ is constant because there are no noisy fluxes. Equation (3.2) states that the accuracy of the result is reduced when the domain size is increased while keeping the marker density constant, or in other words, when noisy fluxes are present.

3.5.1. Flux-Bin Model

To test whether the fluxes between regions are important for the accuracy of the simulation we build a model, restricted to the Wiener process, that contains the fluxes between regions and their fluctuations. On the other hand it reduces the effect of two markers that started at the same position being in the same density measurement bin. This separation will not be achieved perfectly, but the so-called Flux-Bin model will still give insights into the origins of fluctuations.

We start by separating the domain into bins, we will call them flux-bins to distinguish them from bins we use for density measurements. For each flux-bin we define a weight for the markers inside. Instead of saving the position for each marker we only store how many markers are in each flux-bin. After the time Δt we redistribute the markers. For this we assume that all particles started in the center of the bin and use the analytical solution (2.10) of the probability density to calculate the transition probabilities to other flux-bins. The width of the flux-bins should be small compared to the time step Δt to avoid introducing an additional diffusive behavior. We now have the transition probability to other flux-bins for each marker in the original flux-bin. To determine how many markers of the original flux-bin travel to each other flux-bin we can draw from the multinomial distribution, a generalization of the binomial distribution for more than two outcomes [26]. To decrease the runtime of the simulations we only distribute the markers between the 20 flux-bins that are closest to the flux-bin the particles originate from.

We then define density-bins that contain several adjacent flux-bins and save the density for each time step. From these density measurements we compute the variance and compare it to a direct implementation of the algorithms. In section 4.2.4.3 we compare the flux-bin model to the direct simulations of the algorithms. We will see that the flux-bin model is able to reproduce the accuracy of the schemes while the simplified analytical model from section 3.4 fails to do so.

We have to account for markers travelling between flux-bins of different weight. We model the two new roulette schemes differently:

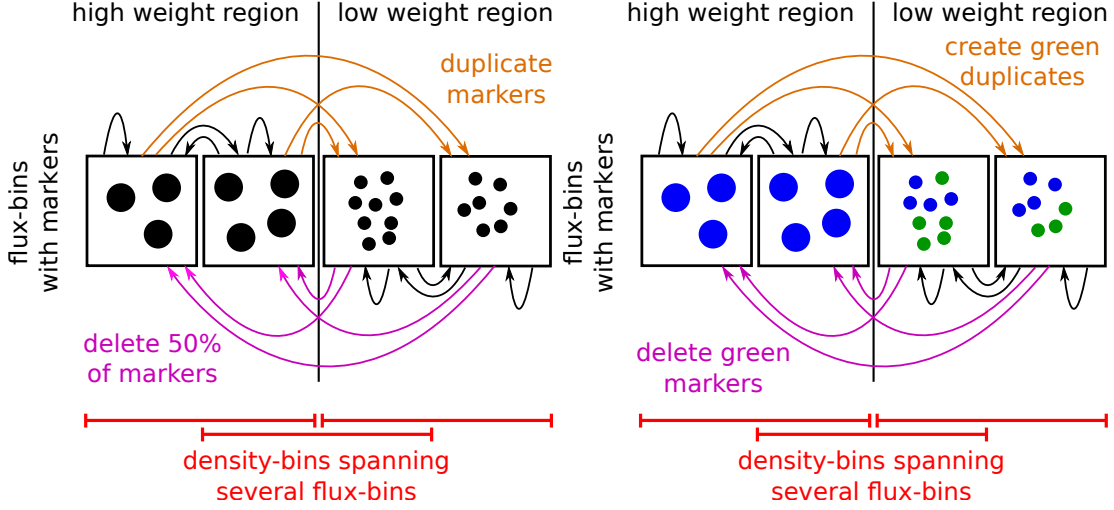
Correlated Roulette

We only consider integer weight ratios between the regions. When n markers travel from the flux-bin A to the flux-bin B, the marker number in flux-bin B is increased by

$$n \left\lfloor \frac{w(A)}{w(B)} \right\rfloor, \quad (3.3)$$

where $\lfloor \cdot \rfloor$ denotes rounding to the next smaller or equal integer and $w(A)$ and $w(B)$ are the weights of the markers in the respective flux-bins. We remove n markers from flux-bin A. To conserve the total weight we remember how much weight we discarded when rounding in equation (3.3) and add them again to flux-bin A.

3. Reweighting



- (a) The flux-bin model for the *correlated roulette*. When markers travel from the high to the low weight region, we add twice as many markers to the target flux-bin as we remove from the origin flux-bin. When markers travel from the low to the high weight region we add half as many markers to the target flux-bin as we remove from the origin flux-bin.
- (b) The flux-bin model for the *deterministic roulette*. When markers travel from the high to the low weight region, we add an additional green marker to the target flux-bin for each marker we remove from the origin flux-bin. When markers travel from the low to the high weight region we only add the blue markers to the target flux-bin.

Figure 3.4.: Sketches for the flux-bin model for the correlated and deterministic roulette. The weight step between the high and low weight regions is 1 : 2. The flux-bins, shown as black boxes, contain markers. The markers of every flux-bin can travel to other flux-bins every time steps, the probability distribution for the transitions is given by equation (2.10) and the positions of the flux-bins. When markers travel between flux-bins inside the same weight region, shown as black arrows, we simply decrease the number in the original bin and increase the number of markers in the target bin. When the markers travel to the low weight region, shown as gold arrows, we additionally have to split the markers. When the markers travel to the high weight region, shown as purple arrows, we additionally have to shoot the markers. The correlated and the deterministic roulette handle markers that travel from one weight region to the other differently. We show the density-bins in red at the bottom of the figures.

Deterministic Roulette

For simplicity we only consider two regions of different weight and only allow integer weight ratios. We keep track of the blue and green markers individually. When a blue marker travels from a low to high weight region we remove it from the original flux-bin and add it to the new flux-bin. If it travels in the other direction we also add a green marker to the new flux-bin. When a green marker travels to another flux-bin we delete it if the new flux-bin is of higher weight, otherwise we move it to the new flux-bin.

3.6. Numerical Estimations of Errors

While we already made some analytical calculations this is only possible for very simple situations. To compare the methods and find good parameter values we have to rely on numerical simulations.

For this I wrote a FORTRAN 2003 program. It is using the different algorithms to simulate various processes. To get the variance we can repeat each simulation several times. For each measure we now have several measurements. The mean μ and variance σ^2 are defined as [21]

$$\mu = \frac{\sum_i s_i}{N} \quad (3.4)$$

$$\sigma^2 = \text{Var}(s) = \frac{\sum_i (s_i - \mu)^2}{N - 1}. \quad (3.5)$$

N is the number of samples which are in the order of 1×10^6 for our purposes. We can therefore neglect the difference between $N - 1$ and N . The standard deviation σ is given by the square root of the variance. We compute the variance and compare the mean of these measurements with simulations without reweighting to judge the performance of the methods. We will use the time average instead of performing the simulation multiple times.

4. Performance of Novel Schemes

4.1. Overview

To judge the performance of our reweighting schemes we have to analyze several splitting and shooting schemes in various configurations. This gives rise to a large number of cases to investigate, especially if one also wants to perform parameter scans. We will first give an overview of the studies we perform and the results thereof before covering the different aspects in detail.

Starting from a single boundary and the Wiener process we will add complexity until we arrive at the full schemes for a more general diffusion-advection equation. Using the conclusions drawn from the analysis of these cases I finally propose a general approach for choosing all parameters of the schemes in section 4.4.

We will use the variance of the results of the schemes as measure for the accuracy. This is only justified if the mean of the results is unchanged by reweighting. We will establish this important property both in the beginning with a single boundary with the Wiener process and when we arrive at a more complex process in section 4.3.4.2.

We start with the Wiener process introduced in 2.4.3.1 in a one dimensional domain with mirror boundary conditions and with constant diffusion coefficient. The domain is split into a low weight and a high weight region, we are only looking at a single boundary. We establish:

- Reweighting indeed increases accuracy in the low weight region. In one dimension the correlated roulette outperforms the deterministic roulette.
- Far from the boundary the accuracy agrees with the flux-bin model.
- Hysteresis and time-discrete reweighting both decrease fluctuations.
- The effects of hysteresis and time-discrete reweighting add predictably. This is important for selecting the hysteresis width and the time between reweighting.
- The decorrelation length is independent of the weight ratio between the regions.
- Using noninteger weight ratios is possible, both proposed methods for splitting perform well.

We then go to multiple boundaries. With an analytical investigation we find that using many boundaries is beneficial compared to using only few. Extending the problem to two dimensions in section 4.2.4.6 we find that the deterministic roulette can perform similarly

4. Performance of Novel Schemes

to the one dimensional case, while the correlated roulette suffers from correlation between many markers.

The goal of using reweighting is to increase accuracy in areas of low density. We introduce a constant flow to get different densities in different regions in section 4.3. At first we focus on a single boundary in section 4.3.3 and find that decorrelation is mostly unaffected by the presence of flow. We then increase the flow velocity such that we get densities that vary over several orders of magnitude in section 4.3.4 and use several boundaries. We find that with reweighting we can achieve the same level of fluctuations in the low density region as without reweighting with a factor of 2000 less markers.

4.2. Wiener Process

4.2.1. Introduction

Diffusion-advection equations contain, as the name suggests, two different terms determining the temporal evolution: diffusion and flow. As discussed in section 3.2.1 the diffusive term decorrelates markers and is a requirement for reweighting. This is not the case for the advection term. We therefore first analyze how our algorithms behave under a pure diffusive process. For simplicity we keep the diffusion coefficient constant, what leaves us with the Wiener process which we introduced in 2.4.3.1. We will restrict ourselves to the constant stationary solution.

For time independent ρ equation (2.11) becomes

$$\Delta\rho = 0. \tag{4.1}$$

For the one dimensional Wiener process solving (4.1) is straight forward. We can obtain it by integrating twice, $\rho(x)$ is a linear function. For mirror or periodic boundary conditions ρ will be constant. A sloped solution would mean that particles appear on one side of the domain and vanish at the other.

In more dimensions equation (4.1) is solved by a constant ρ as well, which is also compatible with mirror and periodic boundary conditions.

4.2.2. Required Number of Markers

To judge the performance of the schemes we have to determine the variance of the results accurately. This requires many repetitions, we typically average over 1 000 000 measurements, and much more cpu time than determining the mean of the results to similar accuracy.

Usually the error of Monte Carlo algorithms is $\mathcal{O}(1/\sqrt{N})$ (equation (2.7)), where N is the number of markers. This however only holds for uncorrelated markers what is not the case for our methods. For many markers we suspect that the error nevertheless is of $\mathcal{O}(1/\sqrt{N})$. We will perform a convergence test for this hypothesis, and to see how many markers are required to be able to reach the $\mathcal{O}(1/\sqrt{N})$ regime. To obtain the convergence behavior we make simulations with different numbers of markers and plot

the variance of the result as a function of the number of markers. As result we looked at the density measurement in the bin with the largest x coordinate, referring to the region with smallest marker weight. In figure 4.1 we see that the convergence order $\mathcal{O}(1/\sqrt{N})$ is indeed recovered for N larger than 100 to 1000, depending on the setting. For our investigations right now 100 markers should be plenty enough. Because we know the scaling is $\mathcal{O}(1/\sqrt{N})$ it would be a waste of computational time to use more markers, the computational time is better invested in more repetitions to improve statistics. If we use more than three boundaries we have to use more markers.

4.2.3. Hysteresis Region in 1D

Before discussing numerical results we derive the error that we expect for the binned density measurement according to the theory introduced in section 3.4. We will compare the theoretical expectations with numerical results in section 4.2.4. The domain shall be $I = [0, 1]$. The hysteresis region starts at a and ends at b with $0 < a < b < 1$. The weights are 1 in the high weight region and w in the low weight region.

The probability density for each species has to be a linear function of x . When there is a source or sink of particles the slope of the probability density may change, this is at a and b respectively. We can immediately write down the densities of the individual marker distributions:

$$p_1(x) = A \begin{cases} 1 & \text{if } x < a \\ \frac{b-x}{b-a} & \text{if } a < x < b \\ 0 & \text{if } x > b \end{cases}$$

$$p_w(x) = A \begin{cases} 0 & \text{if } x < a \\ \frac{1}{w} \frac{x-a}{b-a} & \text{if } a < x < b \\ \frac{1}{w} & \text{if } x > b \end{cases}$$

$$A = \left[\frac{1}{w} + \frac{a+b}{2} \left(1 - \frac{1}{w} \right) \right]^{-1}$$

where p_1 the probability density for markers of weight 1, p_w the probability density of markers of weight w , and A is a normalization constant. A does not depend on the hysteresis width, only on the boundary position $(a+b)/2$.

We can now use equation (3.1) to calculate the variance of the density measurement. To avoid using many case distinctions we require that the bin l lies completely within or outside of the hysteresis region, in figure 4.2 valid and forbidden bin positions are shown.

4. Performance of Novel Schemes

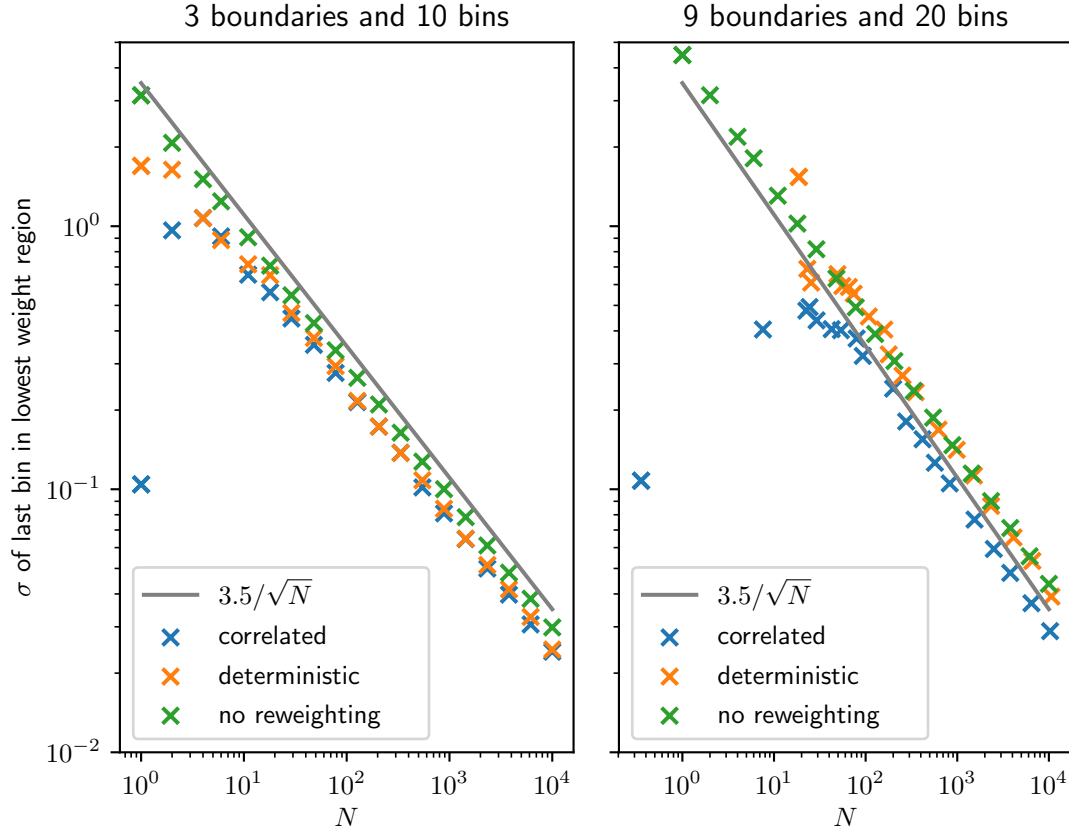


Figure 4.1.: The statistical error σ defined in equation (3.5) of the methods as a function of the number of markers N . On the left we see simulations with 3 boundaries and 10 bins, on the right with 9 boundaries and 20 bins. For comparison there is a line depicting $\mathcal{O}(1/\sqrt{N})$. For large N we have as expected an inverse square root dependency. The required number of markers to reach the $\mathcal{O}(1/\sqrt{N})$ behavior depends on the number of boundaries used.

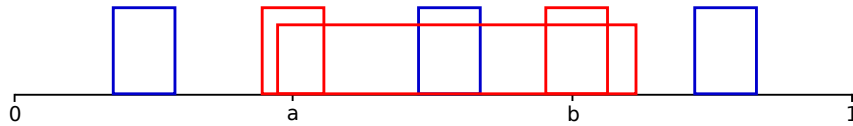


Figure 4.2.: The domain $[0,1]$ with a hysteresis region from a to b . Different bins for measuring the density are shown as rectangles, the three blue ones are covered in equation (4.4). To accurately calculate the mean and variance of the measurements using the three red ones would have to make more case distinctions.

We obtain

$$\begin{aligned} \langle \rho_{\text{bin}} \rangle (x) &= A \\ \langle \rho_{\text{bin}}^2 \rangle (x) &= \frac{A}{l} \begin{cases} 1 & \text{if } x < a \\ \frac{b-x}{b-a} + \frac{1}{w} \frac{x-a}{b-a} & \text{if } a < x < b \\ w & \text{if } x > b \end{cases} \end{aligned} \quad (4.2)$$

$$\text{Var}(\rho_{\text{bin}}) = \langle \rho_{\text{bin}}^2 \rangle - \langle \rho_{\text{bin}} \rangle^2 = -A^2 + \frac{A}{l} \begin{cases} 1 & \text{if } x < a \\ \frac{b-x}{b-a} + \frac{1}{w} \frac{x-a}{b-a} & \text{if } a < x < b \\ w & \text{if } x > b \end{cases} \quad (4.3)$$

As we would expect the variance is lower in regions with lower weight w . In the hysteresis region the variance should be interpolated linearly, neglecting that the markers are not independent of each other.

For the analysis we are interested in the relative error. We therefore divide the variance by the squared mean. For the variance we then get:

$$\begin{aligned} \text{Var}(\rho_{\text{bin}})(x) &= -1 + \frac{1}{Al} \begin{cases} 1 & \text{if } x < a \\ \frac{b-x}{b-a} + \frac{1}{w} \frac{x-a}{b-a} & \text{if } a < x < b \\ w & \text{if } x > b \end{cases} \\ &= -1 + \frac{\frac{1}{w} + \frac{a+b}{2} \left(1 - \frac{1}{w}\right)}{l} \begin{cases} 1 & \text{if } x < a \\ \frac{b-x}{b-a} + \frac{1}{w} \frac{x-a}{b-a} & \text{if } a < x < b \\ w & \text{if } x > b \end{cases} \end{aligned} \quad (4.4)$$

For $x < a$ and $x > b$, whenever the bin is outside of the hysteresis region, the variance is independent of the width of the hysteresis region.

4.2.4. Numerical Investigation

At first we want to understand the behavior of the algorithms when only a single boundary is used. For this we consider the variance multiplied by the mean number of markers, which we denote as 'normalized variance'. We know that the variance is inversely proportional to the mean number of markers, by multiplying with the mean number of markers we can compare simulations with different numbers of markers directly. We also divide by the squared density because we are interested in the relative error, for the constant densities we investigate here this only gives a constant factor. The normalized density can be interpreted as inverse of the relative accuracy we achieve per marker.

We start with an integer weight ratio of 1/2. Noninteger weight ratios will be investigated when we use multiple boundaries in section 4.3.4. As we will split each marker in two we do not need to use a probabilistic splitting mechanism as described in 3.3.3.

As discussed in section 3.3.2 we expect that the particles decorrelate on length scales of the histogram bin width. This means that if we use a histogram to analyze the boundary

4. Performance of Novel Schemes

region, we will only have very few data points as the phenomena we are investigating is of comparable size to the bins. We circumvent this issue by using bins with a width of 0.1, but place one hundred of them in the domain $[0, 1]$. The bins overlap. This gives us more information about what is happening, but it is important to remember that the density would be calculated on a much coarser grid than the lines shown for example in figure 4.3. In the bottom left corner of figure 4.3 we show the overlapping x -positions of the first 20 bins used to compute the density.

In figure 4.3 we can show how the methods proposed in 3.3.2 compare. Both methods increase the accuracy of the simulation in the low weight region as desired, while decreasing the accuracy in the rest of the domain. The difference in accuracy is not as strong as suggested by the theoretical calculation from section 4.2.3. In the following we will analyze various aspects in detail.

4.2.4.1. Unchanged Mean

At first we will ensure that reweighting does not change the mean of the simulation results. For this we run 5000 short simulations (50 time steps where measurements are taken after 10 000 equilibration time steps) and calculate the mean and the standard error of the mean. The standard error of the mean is given by [21]

$$\text{SE} = \frac{\sigma}{\sqrt{m}}$$

where σ is the standard deviation of the samples defined in equation (3.5) and m is the number of samples.

In figure 4.4 we can see that the mean results of the methods lie close to each other, most of the time within one σ of the uncertainty of each other. It is to be expected that some values lie further away than one σ . The standard deviation of a single simulation is by a factor of $\sqrt{5000} \approx 70$ larger than the uncertainty of the mean shown in figure 4.4. This makes it improbable that a deviation of the mean that is significant compared to the noise of the Monte Carlo method is concealed by the noise in figure 4.4.

4.2.4.2. Roulette without further Measures

We will analyze the results of a boundary without any further additions to the reweighting like hysteresis or discrete time reweighting.

Deterministic Roulette (orange in figure 4.3)

The deterministic roulette shows mostly the expected behavior: On the left side of the boundary the variance neither increases nor decreases when approaching the boundary. The accuracy of the density measurement only changes once the bin contains markers from the right side. Pairwise correlated markers should not lead to an increase of variance above the level of the left side.

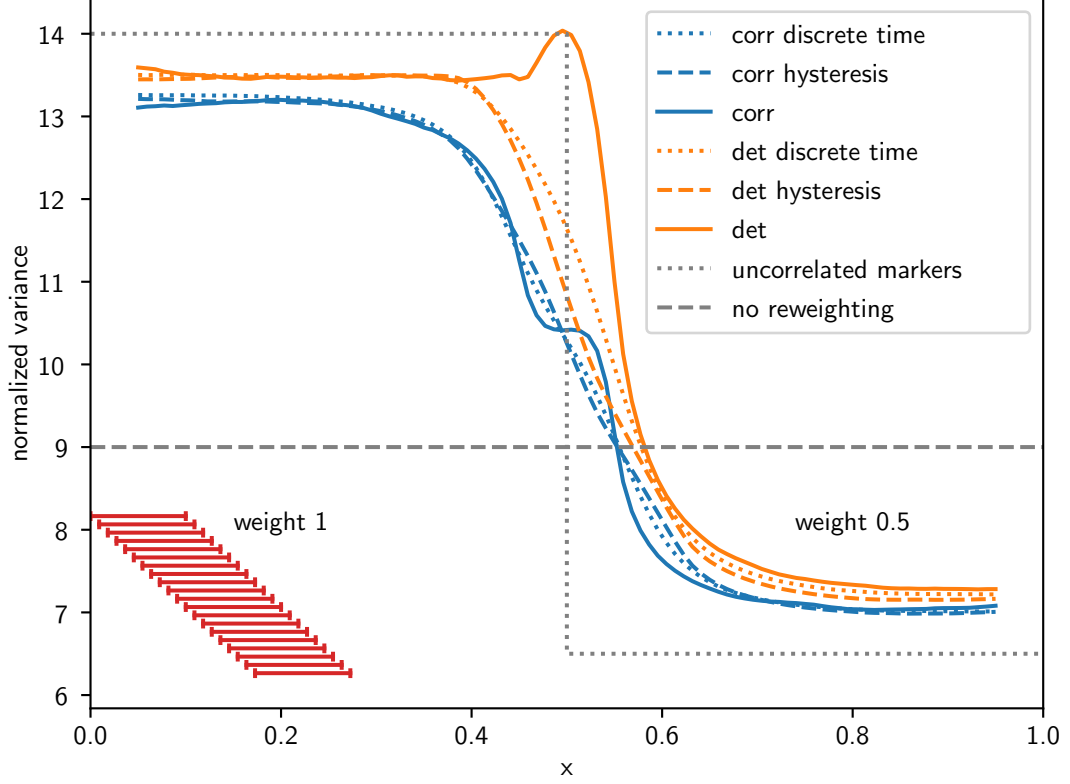


Figure 4.3.: The variance, multiplied with the mean number of markers. The bins have a width of 0.1, but because we use 100 overlapping bins the resolution is finer than one would expect for a histogram with bins of width 0.1. In the bottom left corner we show the overlapping x -positions of the bins. The hysteresis width, if present, is 0.2. The time between reweightings is 2.5×10^{-5} , except for the lines denoted with discrete time reweighting: here we set it to 1×10^{-3} . This corresponds to a typical $\Delta x = 0.045$, approximately half a bin width. For comparison we also show the theoretical values without reweighting, and for no hysteresis and uncorrelated markers. We also include the prediction of the variance by equation (4.4) for uncorrelated markers. The actual results show less variance in the high weight region, and more in the low weight region than predicted. This will be addressed when we discuss the flux-bin model.

4. Performance of Novel Schemes

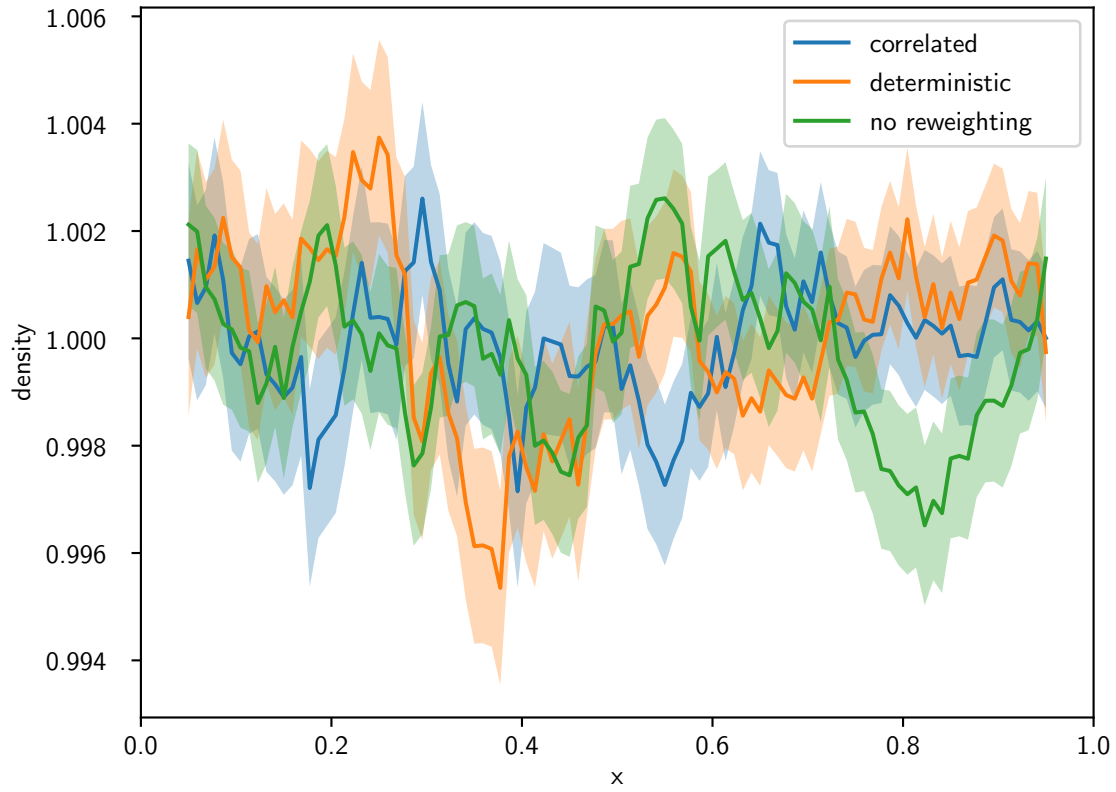


Figure 4.4.: The mean and standard error of the mean of simulations of the Wiener process. The bin size is 0.1, the time between reweighting is 2.5×10^{-5} . All means lie within one σ of each other for most of the data points, suggesting reweighting does not introduce significant changes to the mean of the simulation result.

When a marker is split the resulting twins are not necessarily shot simultaneously. This means that after n crossings there can be 1 to $n + 1$ correlated markers on the right side because we can have more or less 'green' markers than weight conservation would require. The bump is consistent with the expectation that the fluctuations in the number of 'green' markers should reduce the accuracy directly after the boundary.

After the boundary the variance decays, the decay length is as expected of the order of the bin width.

Correlated Roulette (blue in figure 4.3)

The results show in figure 4.3 for the correlated roulette is rather surprising: Already on the left side the error is reduced considerably compared to uncorrelated markers. The markers close to the boundary are more correlated than further away from it. This means that the boundary leads to correlation between the markers such that the error is reduced compared to uncorrelated markers. In the appendix C I present a model explaining the reduced fluctuations due to correlation.

This explains why the variance is so low just left of the boundary. After having crossed the boundary from right to left the markers decorrelate, increasing the error. At the same time markers that crossed from left to right and were therefore split decorrelate as well, decreasing the error. A saddle point is present at the boundary.

Comparison

The correlated roulette is clearly better than the deterministic roulette. With it the variance is always smaller than with the deterministic roulette. Close to the boundary this is probably due to the positive correlation effect. Further away this is due to fluxes between the regions that are less affected of noise. We will also compare the fluctuations in total weight and total number of markers for the deterministic and correlated roulette:

	$\langle N \rangle$	σ_N	$\langle W \rangle$	σ_W
Deterministic	93.1	6.6	63.0	3.8
Correlated	100.5	3.6	66.97	0.35

When we introduced the roulette schemes in section 3.3.2 we explained that weight fluctuations are kept low with the correlated scheme compared to the deterministic scheme. We find the predicted behavior in our simulation results. Fluctuations in the number of markers N arise both because the total weight is not conserved and because the marker positions fluctuate. When more than average markers are in the high weight region the total number of markers is lower than when more than average are in the low weight region.

4. Performance of Novel Schemes

4.2.4.3. Comparison to Flux-Bin Model

In section 3.5.1 we introduced the flux-bin model. It captures the noisy fluxes between regions but reduces correlations between particles originating at the same position by collecting them in bins.

Comparing the flux-bin model to the direct implementation of the schemes serves two purposes: First we want to understand what determines the fluctuation level. Second, if we use small flux-bins we should recover the results from the direct simulation. This gives us an opportunity to test both implementations for mistakes.

Figures 4.5 and 4.6 show both the flux-bin model and the direct simulation of the Wiener process with the correlated and the deterministic roulette. The flux-bin model can predict the accuracy of the schemes further away from the boundaries, and gives results that are very similar to the direct simulation when we use small flux-bins. The results suggest that the variance far from the boundary is indeed determined by the noise level of markers that flow into the regions and the size of the markers, and not only the latter. The analytical result from section 4.2.3 suggests that one can achieve arbitrary high accuracies when selecting a very low weight in the low weight region and a small low weight region. Because the influx of particles will however reduce the accuracy we cannot achieve this. Another insight from the flux-bin model is that both the implementations of the flux-bin model and the direct simulation agree, making errors leading to false results more improbable.

We can predict the variance for the high weight region with the deterministic roulette. If we do not use reweighting, we set $w = 1$ in equation (4.4) and get

$$\text{Var}(\rho_V)(x) = \frac{1}{l} - 1. \quad (4.5)$$

In the deterministic scheme we have two kinds of markers: blue ones survive passing the boundary, green ones are deleted. The blue markers are not influenced in any way by the green markers. The markers in the high weight region are all blue markers, and they behave as if the whole domain was populated by blue markers of high weight. We therefore can use equation (4.5) to calculate the variance, but we have to scale the variance according to Bienaymé's formula [19]. We have a weight ratio of 1 : 2 and the boundary is in the center of the domain. Therefore on average 2/3 of our markers are blue. We multiply the variance given by (4.5) by 3/2 and get $\text{Var}(\rho_V)(x < 0.45) = 13.5$, what is what both the direct numerical simulation and the flux model yield. We require $x < 0.45$ because the bin of width $l = 0.1$ must not be placed across the boundary at $x = 0.5$, otherwise the green markers influence the result.

4.2.4.4. Hysteresis and Time-Discrete Reweighting

Overview

In figure 4.3 we already saw the influence of a hysteresis and discrete-time reweighting. For the correlated roulette both procedures did little more than smoothing out the saddle point. We do not expect an effect due to multiple crossings as there is no direction parallel to the boundary, meaning we cannot correlate a variable other than x itself.

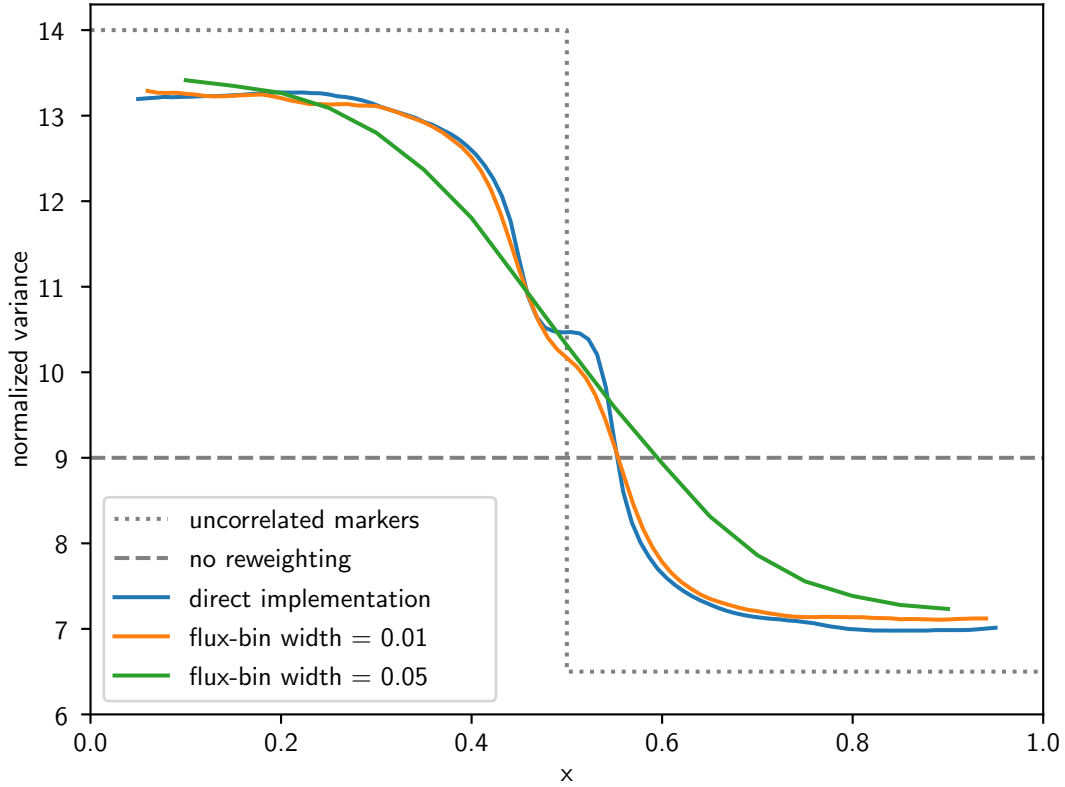


Figure 4.5.: A comparison of the normalized variance for the *correlated roulette* for the direct numerical simulation, and the flux-bin model with high and coarse resolution. The density-bin size is 0.1, the time between reweighting is 2.5×10^{-5} . The flux-bin width is 0.01, for the coarse simulation 0.05. The direct simulation and the flux-bin model with high resolution are very similar, but the flux-bin smooths the saddle point. When using larger flux-bins, denoted as coarse, the transition between the regions differs strongly from the direct simulation. Far from the boundary the results are similar, although the flux-bin model overestimates the fluctuations.

4. Performance of Novel Schemes

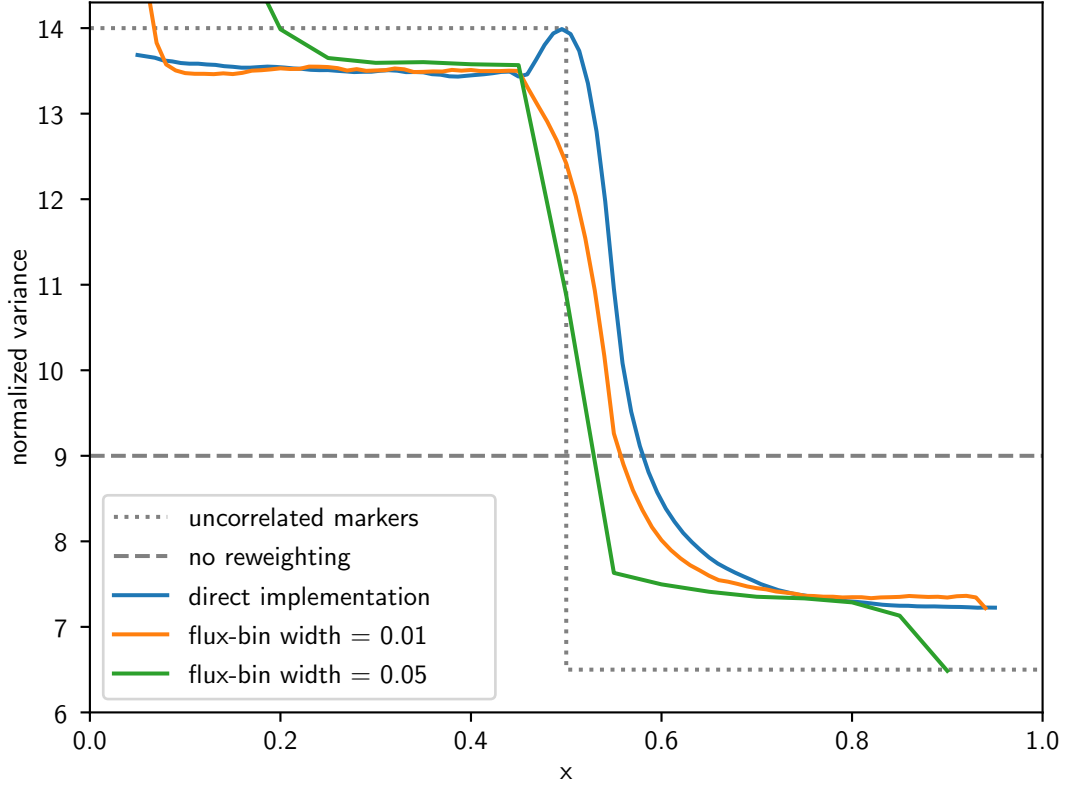


Figure 4.6.: A comparison of the normalized variance for the *deterministic roulette* for the direct numerical simulation, and the flux-bin model with high and coarse resolution. The density-bin size is 0.1, the time between reweighting is 2.5×10^{-5} . The flux-bin width is 0.01, for the coarse simulation 0.05. The flux-bin model with small bins, shown as orange line, is similar to the direct simulation using hysteresis (figure 4.3). When using larger flux-bins, denoted as coarse, the transition between the regions differs strongly from the direct simulation. Far from the boundary the results are similar, close to the edge of the domain we see edge effects for the flux-bin model.

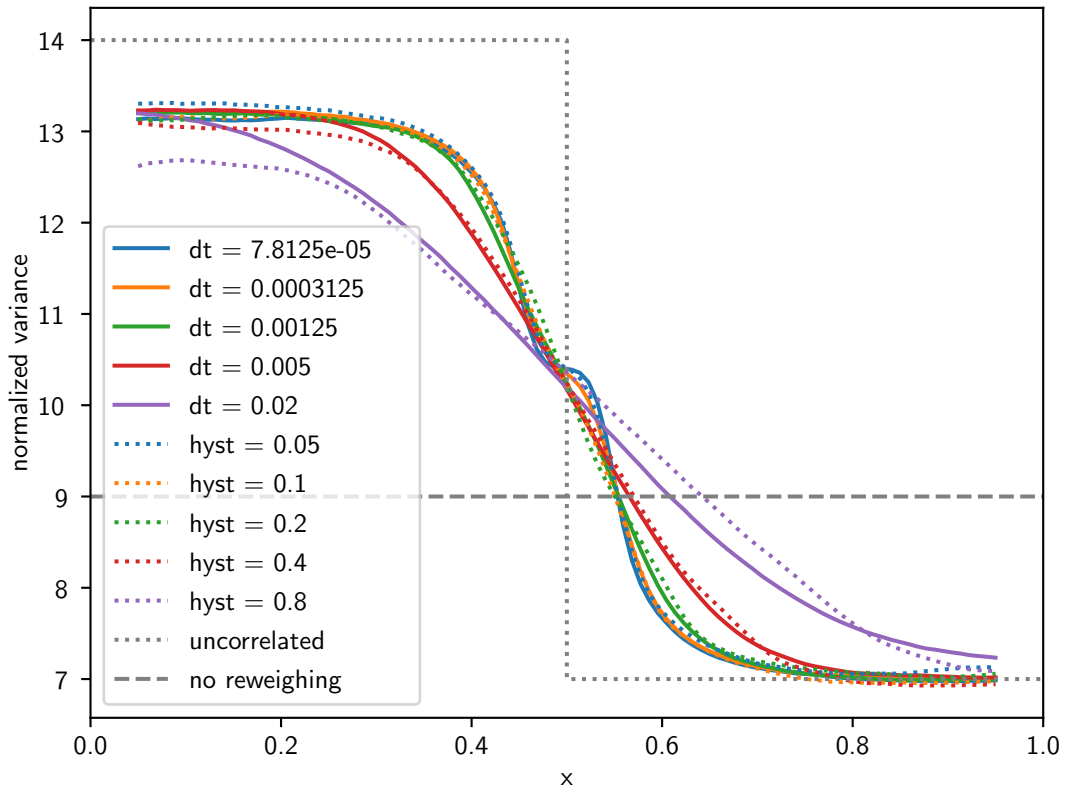


Figure 4.7.: The normalized variance for density measurements using the *correlated roulette*. The bin size is 0.1, the time between reweighting is 2.5×10^{-5} unless otherwise noted. We can see that the effect of discrete time reweighting and of hysteresis are similar. Waiting longer between reweightings reduces the computational cost while the hysteresis does not influence the execution time. Using discrete time reweighting is therefore better, when possible.

For the deterministic roulette this is different. Both methods cut away the bump presumably caused by multiple crossings and extend the region of increased accuracy into the left region.

Correlated Roulette

In figure 4.7 we compare the effects of hysteresis and discrete time reweighting. The characteristic step length between two reweightings is

$$\Delta x_{\text{typ}} = \sqrt{2\Delta t}.$$

We choose the hysteresis width to be $4\Delta x_{\text{typ}}$, and plotted the data for the corresponding discrete time in the same color as the hysteresis data. We can see that the effects of discrete time reweighting are similar to those from the hysteresis.

4. Performance of Novel Schemes

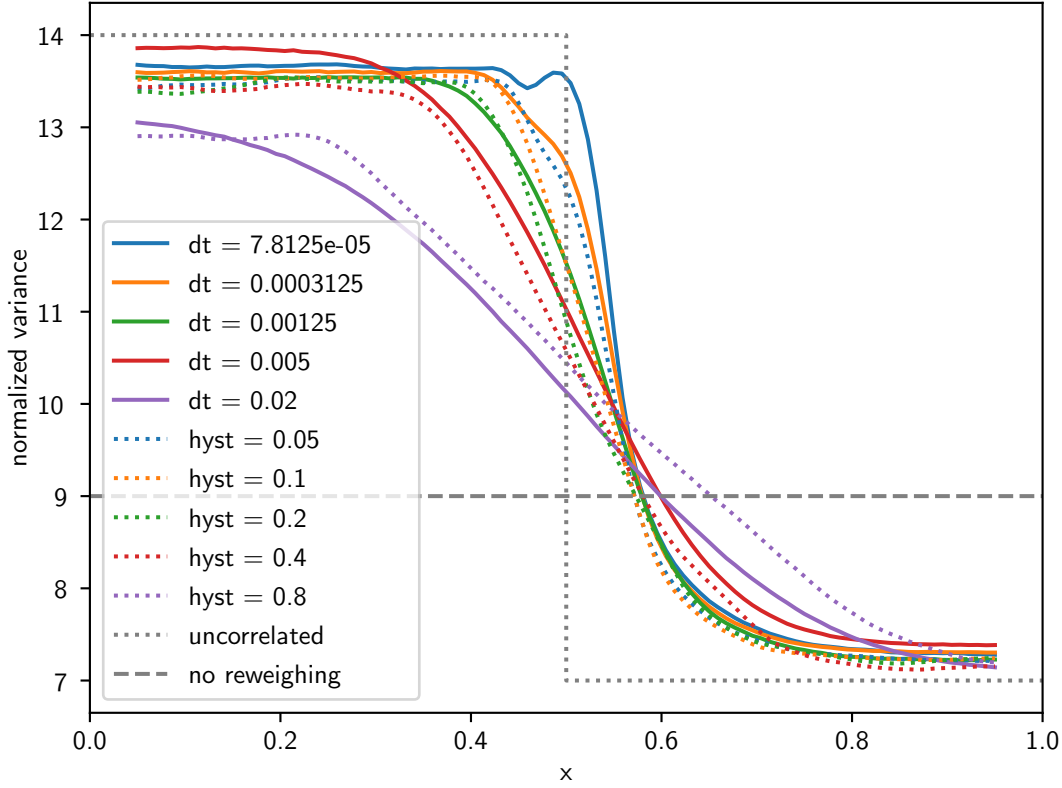


Figure 4.8.: The normalized variance for density measurements using the *deterministic roulette*. The bin size is 0.1, the time between reweighting is 2.5×10^{-5} unless otherwise noted. The results are similar for large hysteresis widths or reweighting after long time periods, when the hysteresis is narrow or reweighting happens fast the hysteresis has a clear advantage.

Deterministic Roulette

We repeated this for the deterministic roulette, we can see the data In figure 4.8 we repeated the comparison between the hysteresis and discrete time reweighting for the *deterministic roulette*. There is not a significant difference between the hysteresis and waiting between reweighting for large parameters. But the hysteresis clearly has advantages for short hysteresis widths compared to short waiting periods between reweightings.

Conclusion

For a weight ratio of 1/2 and the one dimensional Wiener process the correlated roulette does not need further measures to avoid multiple crossings like hysteresis or discrete time reweighting. The deterministic roulette profits from a hysteresis, the effect saturates when the hysteresis width is 2 to 4 times the bin width. Reweighting only after a certain

amount of time has passed, has a similar effect as hysteresis. Wider hysteresis regions and reweighting after longer times has no effect except smoothing the transition between the regions. If the overhead when reweighting often is not an issue using a hysteresis gives better results than reweighting less often. If the overhead is non-negligible reweighting less often can be a good alternative to the hysteresis.

Superposing Effects

In figure 4.7 we see that a hysteresis with width h and reweighting after finite times Δt have similar effects on the variance if

$$h = 4\sqrt{2\Delta t}. \quad (4.6)$$

When we use reweighting in practice a hysteresis and a non-negligible Δt will likely be used together. We will now investigate what the transition width is in this case.

First we define the transition width. The difference between the variance at the position x and the lowest variance at the highest x falls from 100 % at the lowest x and highest variance to 0 % at the highest x and lowest variance. The transition width is then defined as the distance it takes the variance difference to fall from 80 % to 20 %:

We then make a parameter scan where we keep

$$w_{\text{tot}} = w_{\text{hyst}} + w_{\Delta t} = h + 4\sqrt{2\Delta t} = 0.4 \quad (4.7)$$

and vary w_{hyst} .

The results are shown in figure 4.9. We have $w \approx w_{\text{hyst}} + w_{\Delta t}$. The effects of hysteresis and discrete time reweighting add linearly in the sense of equation (4.7). This will be important when selecting the hysteresis width in practice.

4.2.4.5. Dependency of Decorrelation on Weight Ratio

Until now we focused on a weight ratio between the two regions of 1 : 2. When we use a large weight ratio the variance in the high weight region is large, making comparing different weight ratios difficult. We therefore scale the variance of the different cases by a linear function such that they are identical far from the boundary.

In figure 4.10 we can see the scaled variance for weight ratios 1 : 2 and 1 : 8. The saddle point with the correlated roulette is at higher variance with the larger weight ratio, other than that the lines are almost identical. This suggests that we can generalize the results for a weight ratio of 1 : 2. Another conclusion that we can draw is that we cannot achieve arbitrary improvements in the accuracy of the simulation. The noisy influx of particles from the high weight region leads to a noise level in the low weight region that is significantly higher than the analytical expectation, and at least for the deterministic roulette already approaches the uncertainty of the case without reweighting.

4. Performance of Novel Schemes

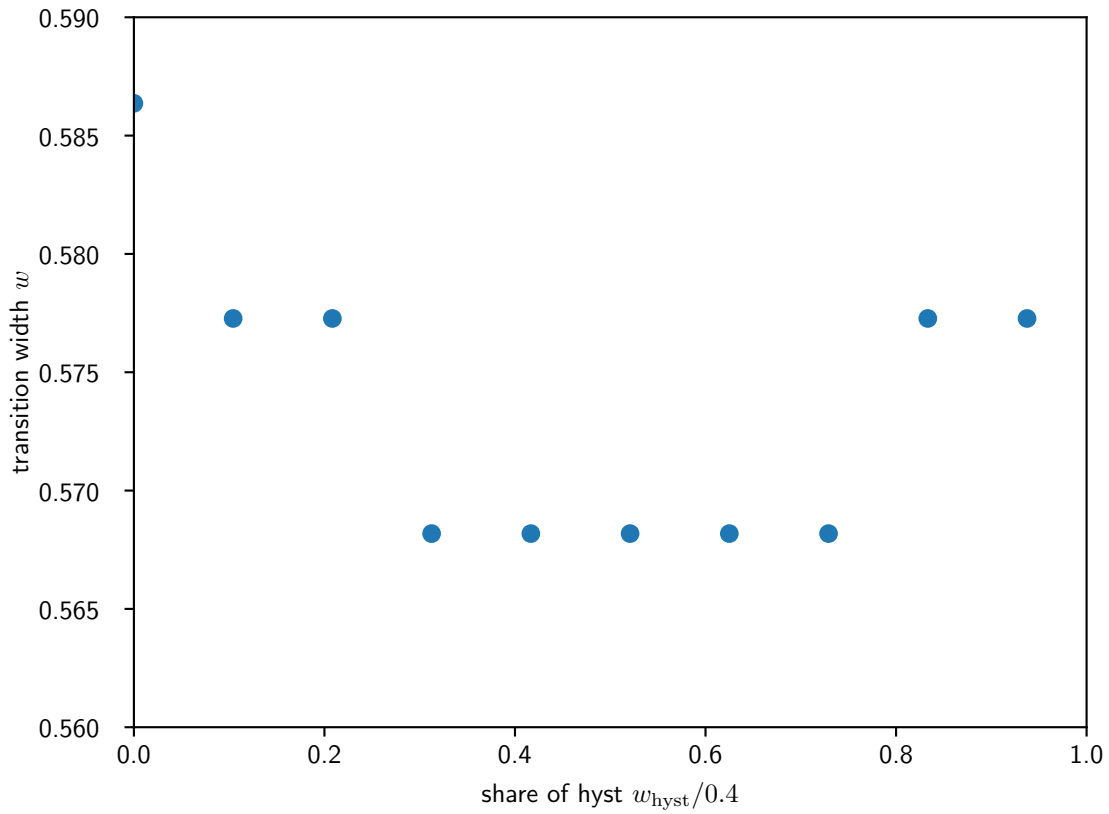


Figure 4.9.: The transition width w as a function of the hysteresis width w_{hyst} . The bin size is 0.1. When w_{hyst} is increased as prescribed by equation (4.7). There are only slight variations of the transition width, the discrete values result from the finite resolution of the measured density function. We can approximate the resulting transition width w as sum of the effects of the hysteresis and discrete time reweighting.

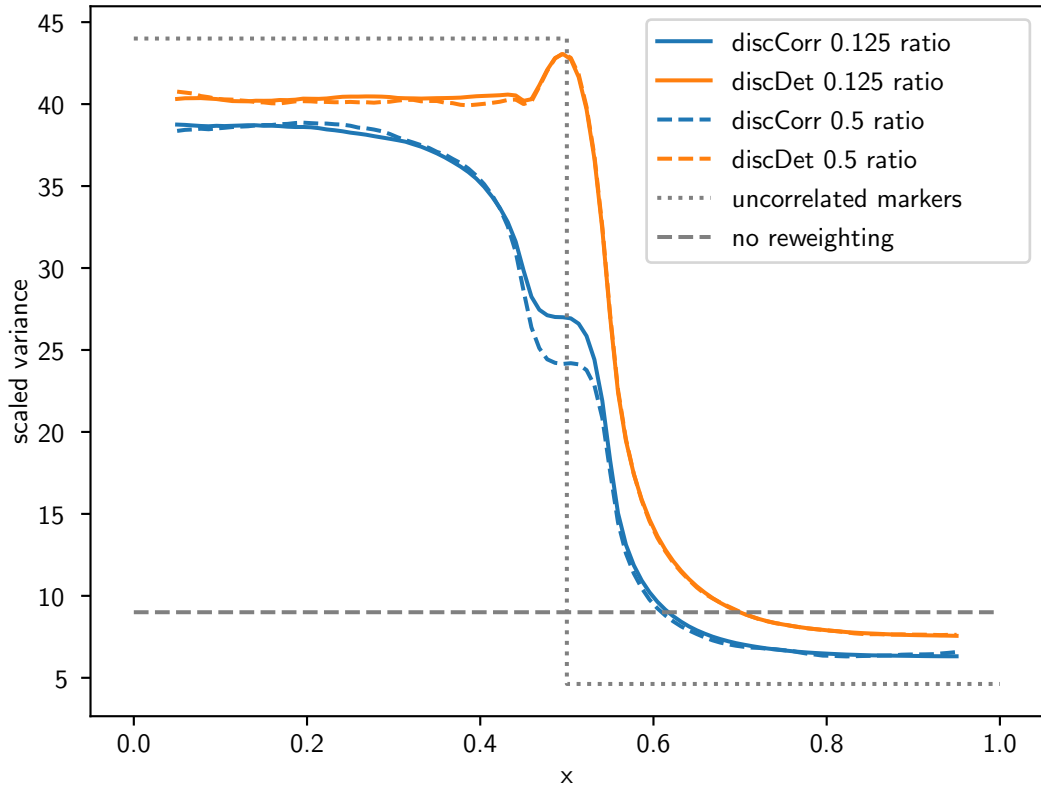


Figure 4.10.: The normalized variance with weight ratio 1 : 2 and 1 : 8. For the weight ratio 1 : 2 it is scaled such that it is identical far from the boundary to the case with 1 : 8. The bin size is 0.1, the time between reweighting is 2.5×10^{-5} . The saddle point sits at different variance, other than this the lines are very similar. We therefore can apply our findings for the weight ratio 1 : 2 also to other weight ratios.

4. Performance of Novel Schemes

4.2.4.6. Two Dimensional

As stated in the overview, we started with the one dimensional Wiener process and add complexity after we analyzed the effects we find. The next step is going from one dimension to two. As we saw in figure 4.3 for the one dimensional problem the correlated roulette is clearly superior to the deterministic roulette: The uncertainty is lower with the correlated roulette everywhere in phase space. This is due to several effects:

- Correlated deletion
- Better conservation of weight close to boundary
- More accurate global conservation of weight

Now imagine a two dimensional domain that is large compared to the bins we use for measuring the density. A boundary separates it in two parts. Across the boundary is a bin in which we measure the average density. Because the bin is so small it is very likely that between two crossings of markers inside of the bin many crossings occur outside of the bin.

A correlated random number generator as described in section 3.3.2.1 depends only on the last few crossings. That means that the random numbers the roulette uses for two consecutive crossings inside of the bin are in most cases uncorrelated because the random numbers are correlated with random numbers used for crossings in *other* bins. Therefore the first two of the advantages of methods utilizing correlated random number generators will likely vanish.

In figure 4.11 we see the uncertainties of the schemes in a two dimensional domain. Let us first look at the schemes without further measures for reducing errors. For the correlated scheme the now uncorrelated shooting leads to large errors close to the boundary. The bump for the deterministic scheme we know from 1D scenarios is flat, otherwise the deterministic roulette is mostly uninfluenced by switching to two dimensions.

When we include a hysteresis region the curves for both schemes resemble their 1D counterparts. For the correlated scheme a possible explanation is: When the markers moved across the hysteresis region they had enough time to mix with markers formerly from y positions outside of the bin.

When we then increase the domain size we can see that the uncertainty is considerably higher with the correlated scheme compared to the deterministic scheme. This suggests that the hysteresis can only effectively counteract locally uncorrelated shooting for small domains.

In conclusion the deterministic roulette appears to be more robust for simulations in multiple dimensions: While the correlated roulette can deliver as good results as the deterministic roulette if the hysteresis region is wide enough to avoid negative effects, it seems to be hard to predict when this condition is met. The deterministic roulette does not show any additional negative effects when going from one to two dimensions. A special case might be when one runs the simulation in multiple dimensions, but uses bins that span all dimensions but one. In the simulation show in figure 4.11 this would correspond to bin widths in y direction of respectively 1 and 10.

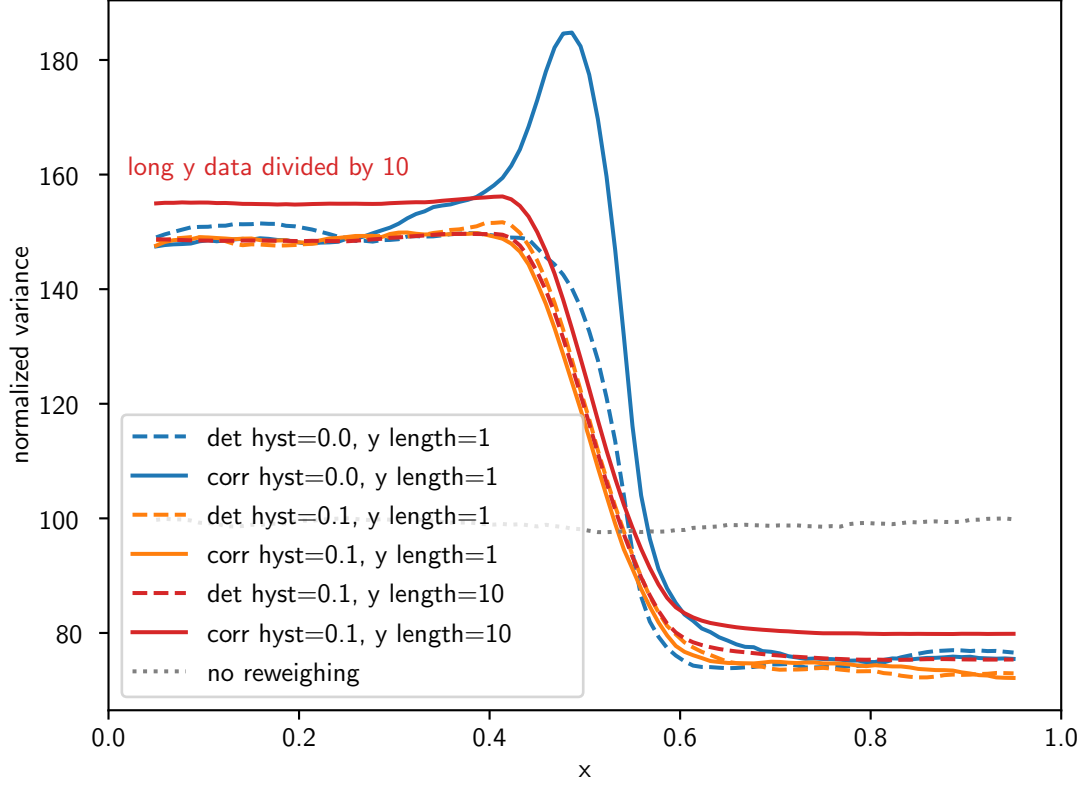


Figure 4.11.: The normalized variance of the measurement of the density with bins of side length 0.1. The time between reweighting is 2.5×10^{-5} . The markers follow the Wiener process, the computational domain is $[0, 1] \times [-0.5, 0.5]$. For the simulations with a long y domain the domain is instead $[0, 1] \times [-5, 5]$. The boundary is at $x = 0.5$, with a weight step of 1 : 2 between the two regions. Because of symmetry, bins with different y coordinates are identical except for edge effect, we only show bins spanning from $y = -0.05$ to $y = 0.05$. The deterministic roulette behaves similarly to the 1D case, while the correlated roulette shows a large increase of variance at the boundary. This is most likely due to locally uncorrelated shooting of markers, and can be mitigated by using a hysteresis. When we increase the domain size however the correlated roulette is less accurate than the deterministic roulette, even when we use a hysteresis. Because the number of markers per phase space volume is lowered by a factor of 10 we expect that the normalized variance is approximately by a factor of 10 larger when we increase the length of the domain in y direction. To compare there results directly we scale these results by a factor of 10.

4.3. Flow

4.3.1. Introduction

The Wiener process we discussed so far leads to a constant distribution function. This is very different from particle distribution functions in nature and experiments. We will now add a second term to our equation representing a constant flow:

$$\Gamma = -D \frac{\partial}{\partial x} \rho + v \rho \quad (4.8)$$

$$\frac{\partial}{\partial t} \rho = -\frac{\partial}{\partial x} \Gamma = D \frac{\partial^2}{\partial x^2} \rho - v \frac{\partial}{\partial x} \rho \quad (4.9)$$

$$dx = \sqrt{2D dt} \xi + v dt. \quad (4.10)$$

We assume D and v to be constant. Again we restrict ourselves to stationary solutions, we set $\partial_t \rho = 0$. We make the ansatz

$$\rho(x) = C_0 \exp \lambda x + C_1 \quad (4.11)$$

and obtain

$$\rho(x) = C_0 \exp \frac{v}{D} x + C_1. \quad (4.12)$$

To avoid any edge effects we will set $v = 0$ close to the domain boundaries. At all other points $v \neq 0$. If we have a net flow of 0, enforced by the boundary conditions, $b = 0$.

4.3.2. Analytical Calculations

For verification and scaling of results we will derive the mean and the variance of the binned density measurement. At first we will show that the density function is continuous even when we change the flow velocity. Then we will derive the density function using continuity and equation (4.12). Finally we calculate the mean and variance of the binned density measurement.

4.3.2.1. Discontinuous Flow

As stated before we will switch off the flow close to the boundaries. We will derive now that the density is continuous, but not differentiable at this position. We start with (4.8) and set it to 0: Integrating (4.9) once leads to a constant flow, but at the boundaries it has to be 0. Therefore it is 0 everywhere.

$$\Gamma = -D \frac{\partial}{\partial x} \rho(x) + v_0 \Theta(x) \rho(x) = 0 \quad (4.13)$$

$$\partial_x \rho = \frac{v_0}{D} \Theta(x) \rho \quad (4.14)$$

$$\rho - \rho_0 = \int_{x_0}^x dx' \frac{v_0}{D} \Theta(x') \rho(x'). \quad (4.15)$$

Θ is the Heaviside step function, which is 0 below 0 and 1 otherwise. Consider the left hand side of equation (4.15). Because the integrand is finite $\rho(x)$ is continuous. Equation (4.14) requires C_1 of (4.12) to be 0.

4.3.2.2. Density Function

In our tests we will set the flow to 0 in the regions closer than b to the domain boundaries. We set the domain length to 1. We know that ρ has to be continuous. Using our results from subsection 4.3.1 we get the following density function:

$$\rho(x) = A \begin{cases} 1 & \text{if } x \leq b \\ \exp \left[\frac{v}{D}(x - b) \right] & \text{if } b < x < 1 - b \\ \exp \left[\frac{v}{D}(1 - 2b) \right] & \text{if } x > 1 - b \end{cases} \quad (4.16)$$

We will interpret ρ as the probability distribution for a single marker. Therefore we have to normalize it:

$$\begin{aligned} \int_0^1 \rho(x) dx &= A \left[b \left(1 + \exp \left[\frac{v}{D}(1 - 2b) \right] \right) + \frac{D}{v} \left(\exp \left[\frac{v}{D}(1 - 2b) \right] - 1 \right) \right] \\ &= A \left[b - \frac{D}{v} + \exp \left[\frac{v}{D}(1 - 2b) \right] \left(b + \frac{D}{v} \right) \right] \\ &\stackrel{!}{=} 1 \\ A &= \frac{1}{b - \frac{D}{v} + \exp \left[\frac{v}{D}(1 - 2b) \right] \left(b + \frac{D}{v} \right)}. \end{aligned}$$

4.3.2.3. Mean and Variance of Binned Density Measurement

With the density function we obtained in the last section we can predict the mean and the variance of the binned density measurement. We will use it in the next section to scale results.

We calculate the moments of

$$f(x) = \frac{\mathbb{1}(x)}{l}$$

4. Performance of Novel Schemes

and the density function (4.16). We only consider $b + l/2 < x < 1 - b - l/2$ and obtain

$$\begin{aligned}\langle f(x) \rangle &= \int_0^1 \frac{1}{l} A \exp \left[\frac{v}{D} (x - b) \right] \\ &= \frac{AD}{lv} \left(\exp \left[\frac{v}{D} (x + l/2 - b) \right] - \exp \left[\frac{v}{D} (x - l/2 - b) \right] \right) \\ \langle f^2(x) \rangle &= \int_0^1 \frac{1}{l^2} A \exp \left[\frac{v}{D} (x - b) \right] \\ &= \frac{AD}{l^2 v} \left(\exp \left[\frac{v}{D} (x + l/2 - b) \right] - \exp \left[\frac{v}{D} (x - l/2 - b) \right] \right).\end{aligned}$$

For the relative error we then get

$$E_{\text{rel}} = \sqrt{\frac{\text{Var}(f)}{\langle f(x) \rangle^2}} = \sqrt{\frac{\langle f^2(x) \rangle}{\langle f(x) \rangle^2} - 1} = \sqrt{\frac{v}{AD} \frac{\exp \left[\frac{v}{D} b \right]}{\exp \left[\frac{v}{D} (x + l/2) \right] - \exp \left[\frac{v}{D} (x - l/2) \right]} - 1}. \quad (4.17)$$

In figure 4.13 we will see that E_{rel} is predicted very accurately.

4.3.3. Single Boundary

We start our investigation of the flow equation again with a single boundary. We place a single boundary in the middle of the domain and see how the results compare to the pure Wiener process. To compare the variance of the density measurement between the Wiener and the flow process we have to scale the variance. Compared to the Wiener process the variance increases for large x because the marker density decreases. We therefore divide the measured variance by the theoretical value without reweighting given in section 4.3.2.3.

For the Wiener process more markers are in the low weight region than for the flow process if the boundary position and weight step are the same. The variance, when divided by the variance without reweighting, is therefore in the whole domain higher for the Wiener process than for the flow process. We additionally scale the variance by a constant factor such that the variance is identical at large x .

The results are shown in figure 4.12. The decorrelation length is very similar, considering numerical noise possibly even identical, for both processes. It is an important quantity when choosing parameters for reweighting, and because it does not depend on the flow we can use our findings about decorrelation for more complicated processes.

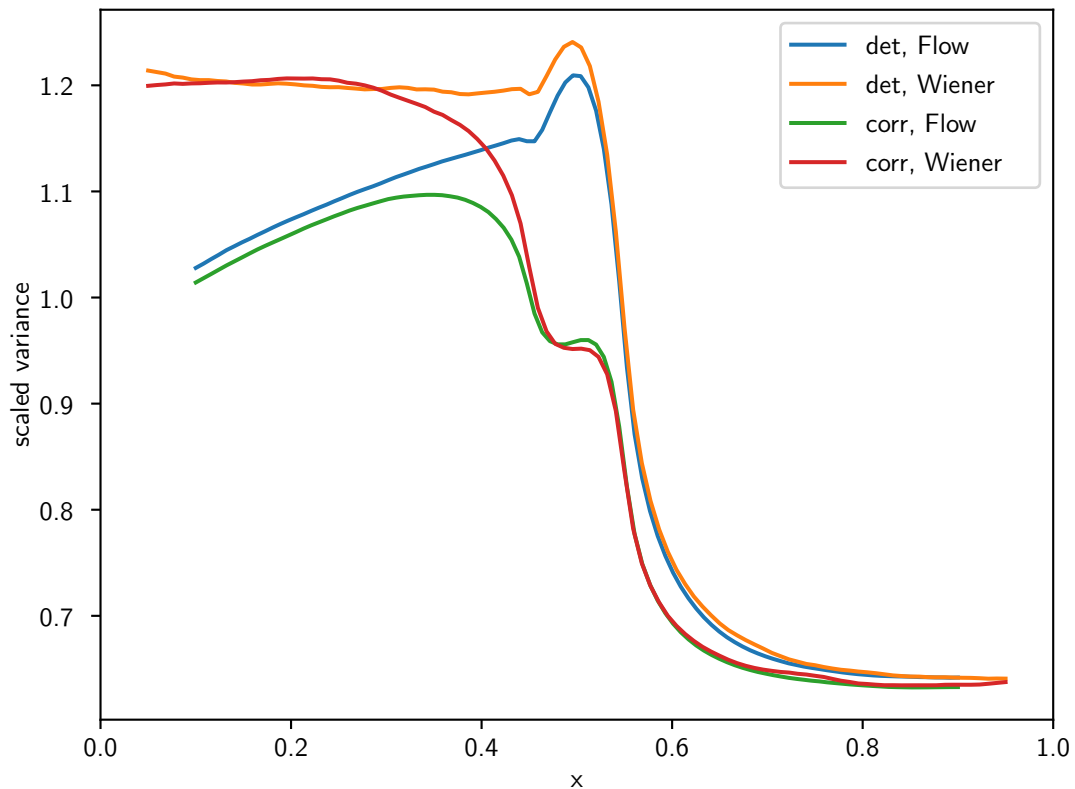


Figure 4.12.: The scaled variance for both the Wiener process as in figure 4.3 and the flow process with $v/D = -2$. The bin size is 0.1, the time between reweighting is 2.5×10^{-5} and the hysteresis width is 0.047. The variance is divided by the variance obtained without reweighting and multiplied by an additional factor such that the variances for the different processes agree for large x . For the correlated roulette the variances show identical behaviour for $x > 0.54$, for the deterministic roulette the variance decays slightly faster for the flow process, but this might be due to noise or imperfect scaling of the results.

4.3.4. Multiple Boundaries

As stated in the introduction 1 we are mainly interested in the low density regions. Therefore we use a strong flow of ≈ -13.3 , resulting in a quickly decreasing density as the diffusion constant is 1. We set the weight regions such that we keep the number of markers per length unit constant. We will place multiple boundaries, more if we use smaller step sizes. According to what we found in section 4.2.2 we increase the number of markers we use from 100 to 1000 from now on.

Using equation (4.16) we see that the density is reduced by a factor of 2.4×10^{-5} over a distance of 0.8. As shown in figure 4.13 this leads to an increase of the variance of the density measurement of about 7×10^4 . We can reduce the variance in the low density region by a factor of about 1800 if we use our reweighing schemes. This means to get the same accuracy without reweighing one has to use 1800 times as many markers. The error is reduced by a factor of ≈ 40 .

4.3.4.1. Weight Ratio

In figure 4.14 we can see how the variance of the measured density depends on the used weight step. Smaller weight steps result in fluctuations that do not change as much in between two boundaries, the variance as function of x does not oscillate as much. The higher accuracy for low densities comes at the cost of lower accuracy for high densities, in section 4.4.1 we show how we can choose how much accuracy at high densities to invest in accuracy at low densities. The size of this effect is similar to what one can achieve by setting the boundaries differently as described in section 4.4.1.

4.3.4.2. Unchanged Mean

As stated in the overview 4.1 it is not sufficient to analyze the variance because the mean of the simulation results might change. To establish that the mean remains unchanged by reweighing also for processes with flow we repeat our analysis that we performed on the Wiener process. We run 350 short simulations (50 time steps where measurements are taken after 10 000 equilibration time steps). Without reweighing we run 2000 instead of 350 simulations because the results without reweighing suffer from more noise than those with reweighing. We compute the mean of the simulation results and the uncertainty of this mean.

In figure 4.15 we show the mean and its uncertainty, but we scale it by $\exp(xv/D)$ to increase the visibility. Other than with the Wiener process the mean without reweighing is only useful for comparison for small x because the simulation results fluctuate much stronger for large x than the simulation results with reweighing. As for the Wiener process the means lie within one σ of each other for most data points, suggesting the mean remains unchanged by reweighing.

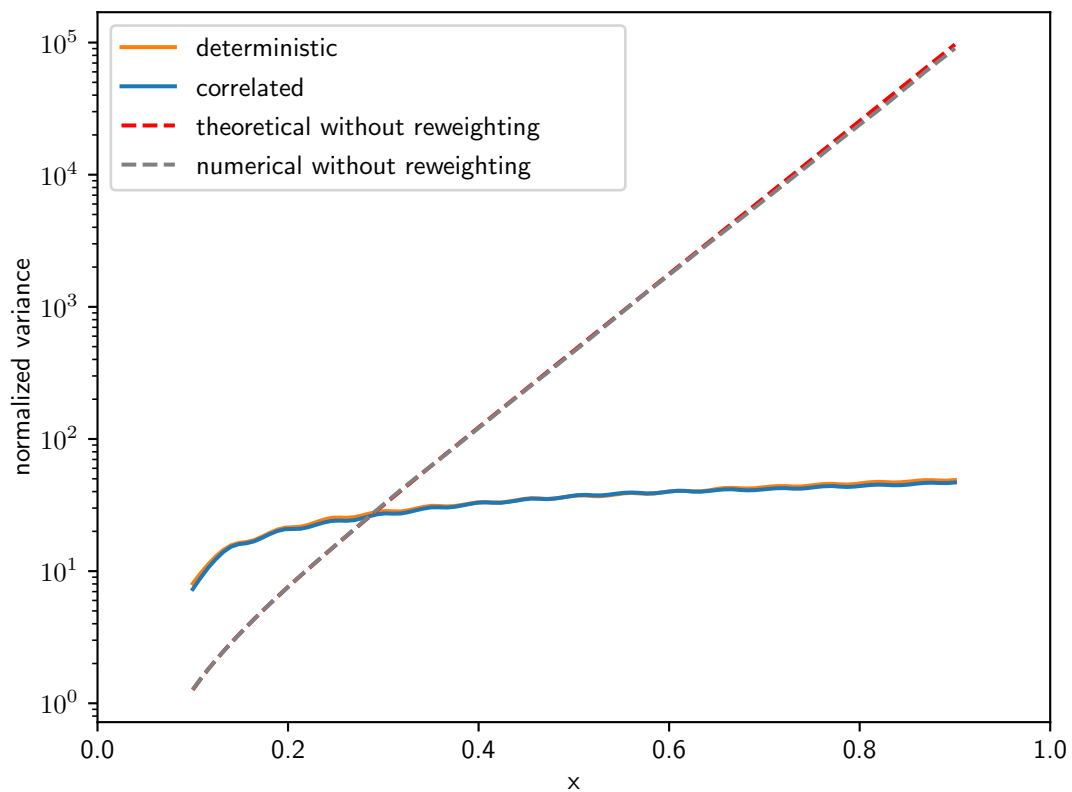


Figure 4.13.: The normalized variance when the density varies considerably in phase space with $v/D \approx -13.314$. The bin size is 0.1, the time between reweighting is 2.5×10^{-5} , the weight step 1 : 2 and the hysteresis width is 90 % of the weight region width. By using reweighting, both with the correlated and the deterministic roulette, we can decrease the error considerably. In figure 4.14 we compare different weight ratios for this case. We also show the prediction for the variance without reweighting from equation (4.17). The agreement is very good, the slight deviation at large x from the numerical result at large x is most likely due to the bad resolution without reweighting at low densities.

4. Performance of Novel Schemes

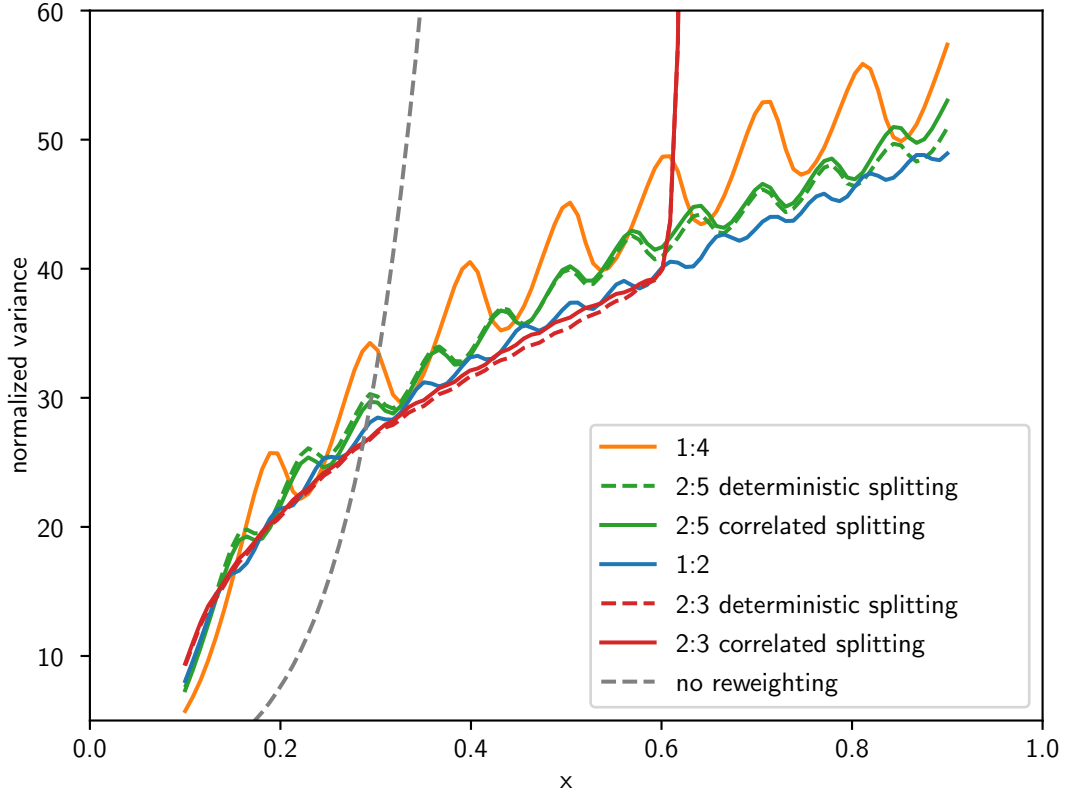


Figure 4.14.: Repetition of figure 4.13, using only the *deterministic roulette* and different weight ratios between regions. The hysteresis width is 90 % of the distance between weight regions. Using the larger weight step 1 : 4 leads to slightly higher accuracy at low x (5.7 instead of 8), starting at $x = 0.15$ the error increases periodically by up to 25 % compared to the 1 : 2 weight step. In section 4.4.1 we will see that increasing the accuracy for high densities leads to a reduction of accuracy for low densities to the extent we see here. Using the noninteger weight step 2 : 5 which lies in between the two integer weight steps leads to a variance in between the two integer weight steps. There is only a small difference between deterministic and correlated splitting. The smaller weight step of 2 : 3 yields only very small oscillations. Our implementation stores the relative weights of the regions as fractions. Therefore our implementation can handle only a limited number of boundaries before problems due to the exponential increasing numerator and denominator occur. This is most likely the reason for the strongly increasing error at $x \approx 0.6$ with the weight step 2 : 3.

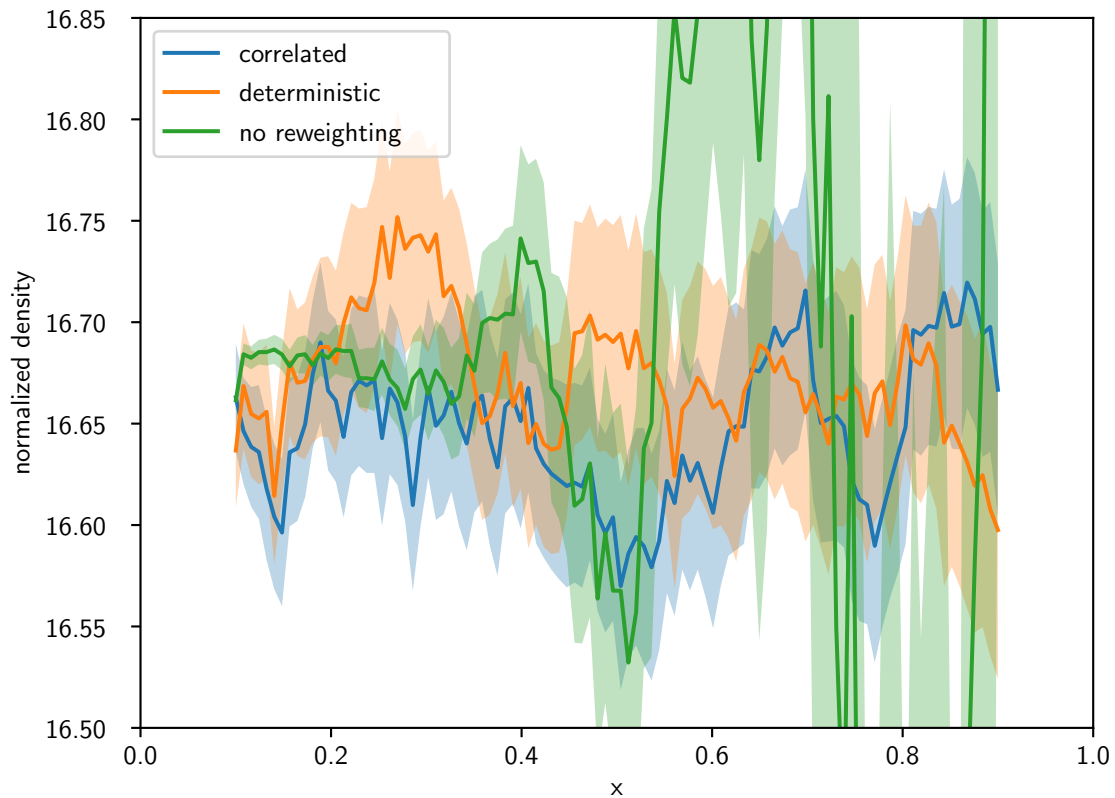


Figure 4.15.: The mean and uncertainty of the mean scaled by $\exp(xv/D)$ as described in 4.3.4.2. All parameters are as in figure 4.13. The solid lines show the mean, all values inside the shaded regions lie within one standard deviation of the mean. We see that the mean without reweighting suffers strongly from noise for large x while it is more reliable for small x . The means for the correlated and deterministic roulette lie within one σ of the result without reweighting and within one another for most data points.

4.4. Parameter Choices

We have many degrees of freedom in our reweighing schemes. Apart from the splitting and shooting we have to decide on the weighing function, the weight steps, the hysteresis widths and the time between reweighing. This is on one hand fortunate because each parameter can be tuned to give the best performance, on the other hand utilizing the schemes is not straight forward. Finally one also has to define a minimum weight one wants to use. This depends on the application.

I therefore propose a standard procedure to choose the parameters based on our findings so far.

4.4.1. Weighting Function

In the section introducing importance sampling [2.4.2](#) we discussed how we would ideally choose the weights if we had uncorrelated markers. Because our markers are not uncorrelated this changes somewhat. Also, even when we only need markers from a small region of phase space for our measure, we have to populate the whole phase space to get the flux densities to and from our region of interest. To get the correct fluxes we also have to resolve the outside region with reasonable accuracy.

One choice for the weighing function that goes very well with this flux requirement is one that is proportional to the phase space density. Then the marker density is constant over all phase space. This is also the weighing function that is suggested by importance sampling when considering the binned phase space density. At the same time it gives a greatly improved accuracy in the low density regions. If the domain is infinite we cannot use a weighting function that results in a constant marker density, we have to introduce a minimal weight after which markers are not split further. In the next section we will see that we can use the markers to determine the present density and the weighting function.

In section [4.3.4](#) we found that for the parameters of the simulation used in section [4.3.4](#) we can gain several orders of magnitude of accuracy in the low density region, but also loose almost an order of magnitude at high densities compared to a simulation without reweighing. When one is interested both in measures that are not peaked in the high tail of the distribution, and measures that are, loosing accuracy in the bulk of the distribution is problematic. An example for a measure that is not peaked is for example the real-space density, an example for a peaked measure is the histogram of the phase space density.

We can however resolve both regions with good accuracy when choosing the continuous weighting function that determines the weight regions accordingly.

To get a worst-case estimate of what we can achieve let us consider the following situation: We run two independent simulations, one with reweighing and one without, each with $N/2$ markers. For the bulk region of the distribution we use the result without reweighing, for the tail the result with reweighing. Compared to the results of a simulation without reweighing and N markers we increased the error in the bulk by $\sqrt{2}$, and compared to a simulation with reweighing and N markers we increased the error in the tail by $\sqrt{2}$. Instead of simply discarding the results of the simulation without reweighing in the tail and the results of the simulation with reweighing in the bulk we can use all

markers, both the reweighted ones and the ones with constant weight, to evaluate the measures. When we do this correctly we can further increase the accuracy of the simulation because we have more sampling positions of the distribution function compared to when we discard markers. In section 3.3.3 we found that using markers of the same weight at the same position is optimal. We will now define a continuous weighting function that yields the same phase space density of markers as running independent simulations with different weighting functions.

In general we can define an ideal weighing function w_i for each measure i one wants to use. This gives us a marker density that would be ideal for each measure:

$$m_i = \frac{\rho}{w_i},$$

where ρ is the phase space density. We combine these marker densities using a weighted average with the weights η_i :

$$m = \frac{1}{\sum_i \eta_i} \sum_i \eta_i m_i = \frac{1}{\sum_i \eta_i} \sum_i \frac{\eta_i}{m_i}. \quad (4.18)$$

This gives us a combined weighing function w

$$w = \frac{\rho}{m} = \frac{\sum_i \eta_i}{\sum_i \eta_i / w_i}. \quad (4.19)$$

An important example is one weighing function optimized for measures without a strong energy dependence such as real space density, that is constant. And another one optimized for measures peaked in the tail of the distribution such as phase space density, which is proportional to the phase space density. If we have a Maxwellian, and therefore an exponentially decaying distribution, this yields

$$w(x) = \frac{\eta_1 + \eta_2}{\eta_1 + \eta_2 \exp[sx]} = \frac{1 + \exp(-sb)}{1 + \exp[s(x - b)]}. \quad (4.20)$$

where s is the decay speed and b the position where both weighing functions contribute the same. We can recover the original weighing functions by letting b go to $\pm\infty$. In figure 4.16 we can see what variances we get for different values of b . As expected we can interpolate between the curve without reweighting and the weighting function that is proportional to the density. It is for example possible to gain more than two orders of magnitude at low densities while losing less than 4 % in the high density region, compared to no reweighting. The accuracy losses we encounter for low densities when increasing the accuracy for high densities is similar to what we saw in figure 4.14 when we used different weight ratios between regions.

4. Performance of Novel Schemes

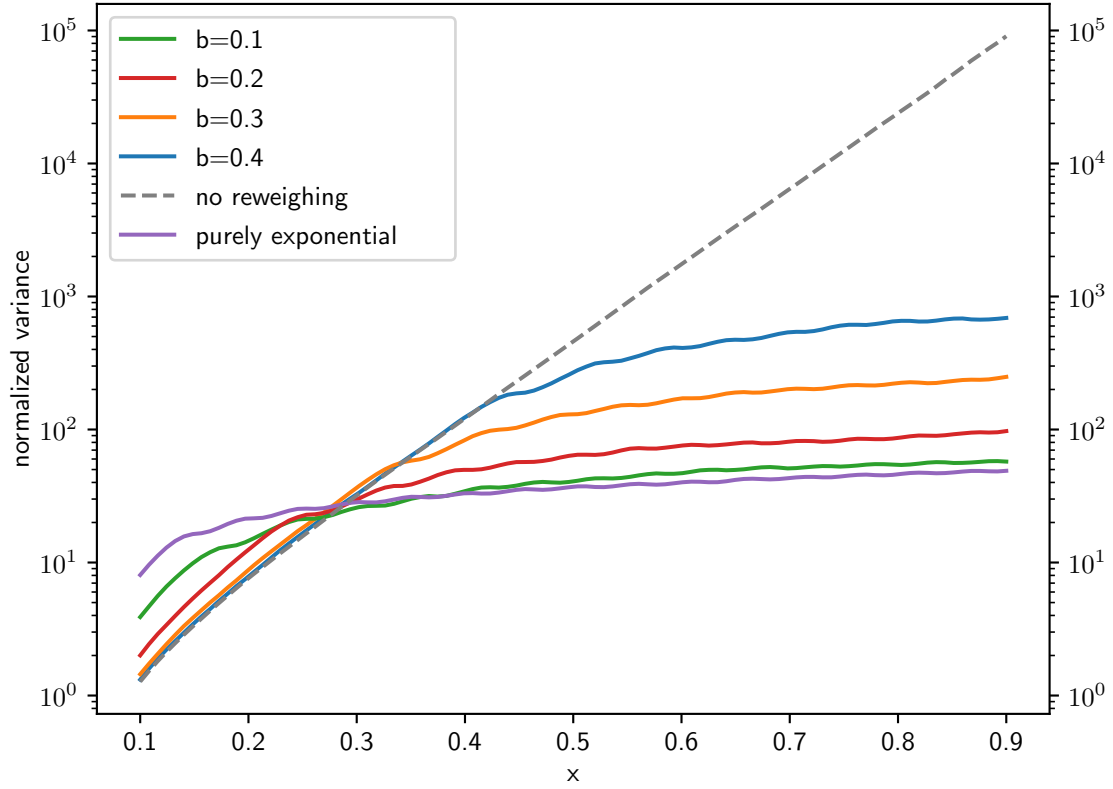


Figure 4.16.: Repetition of 4.13 with the weighing function (4.20), the deterministic scheme and a weight ratio of 1:2. As expected the normalized variance of the new weighing function interpolates between the purely exponential and the constant weighing function case. The parameter b controls how many markers should be put into the tail. As expected we can have a single simulation that is never worse than a factor of 2 with respect to both no reweighing and an exponential weighing function ($b = 0.2$). But we can also increase the accuracy in the high or low density region without compromising it in the other region. It is for example possible to gain more than two orders of magnitude at low densities while losing less than 4% in the high density region, compared to no reweighing.

4.4.1.1. Automatic Weighting Function

Until now we had to guess the distribution function beforehand to choose the weighting function. Instead, we can use the markers themselves to determine the present density and the weighting function. We still can combine different types of weighting function with equation (4.19). For testing purposes we calculate the density with a bin of width 0.3 centered at the marker position for which we want to determine the weighting function. This increases the order of the reweighting procedure from $\mathcal{O}(N)$ to $\mathcal{O}(N^2)$ with N the total number of markers. For efficient simulations one can retain $\mathcal{O}(N)$ by using a histogram to compute the density and then interpolated the value to get a density estimation for each position in phase space. To obtain a reliable value for the density for the weighting function one has to use a minimum number of markers, in our case 1000 were sufficient.

In figure 4.17 we show that using the measured density for the weighting function gives just as good results as using the analytical expression, but there is an effect very similar to using equation (4.20). This is probably due to the large bin we used for calculating the density. This results in a smoothed measurement of the density, which is less steep than the exact density.

4.4.1.2. Mean Number of Markers

When prescribing a weighting function the total, mean number of markers may change with time. For example, let us start with a Maxwellian distribution and a weighting function constant in time that is proportional to the initial Maxwellian distribution. This means the weight of a marker at position x is proportional to the initial distribution function at the position x . When then a highly populated high energy tail develops due to ICRH many markers travel to low weight regions, increasing the total number of markers. This is problematic as the runtime of the simulation would increase considerably and would be hard to predict.

We will now show that this does not occur when we use a weighting function that is a combination of constant weight and weight proportional to the density. We will make the assumption that the weight of the markers follows the weighting function exactly and is not discretized. When we use regions with weight ratios that are not very close to 1 the mean marker number will only be approximately conserved. We also require conservation of mass and that the computational domain does not change during runtime.

The marker density is given by

$$w = \frac{\rho}{m}.$$

For the constant weight we have:

$$\begin{aligned} w &= w_0 \\ m &= \frac{\rho}{w_0}, \end{aligned}$$

4. Performance of Novel Schemes

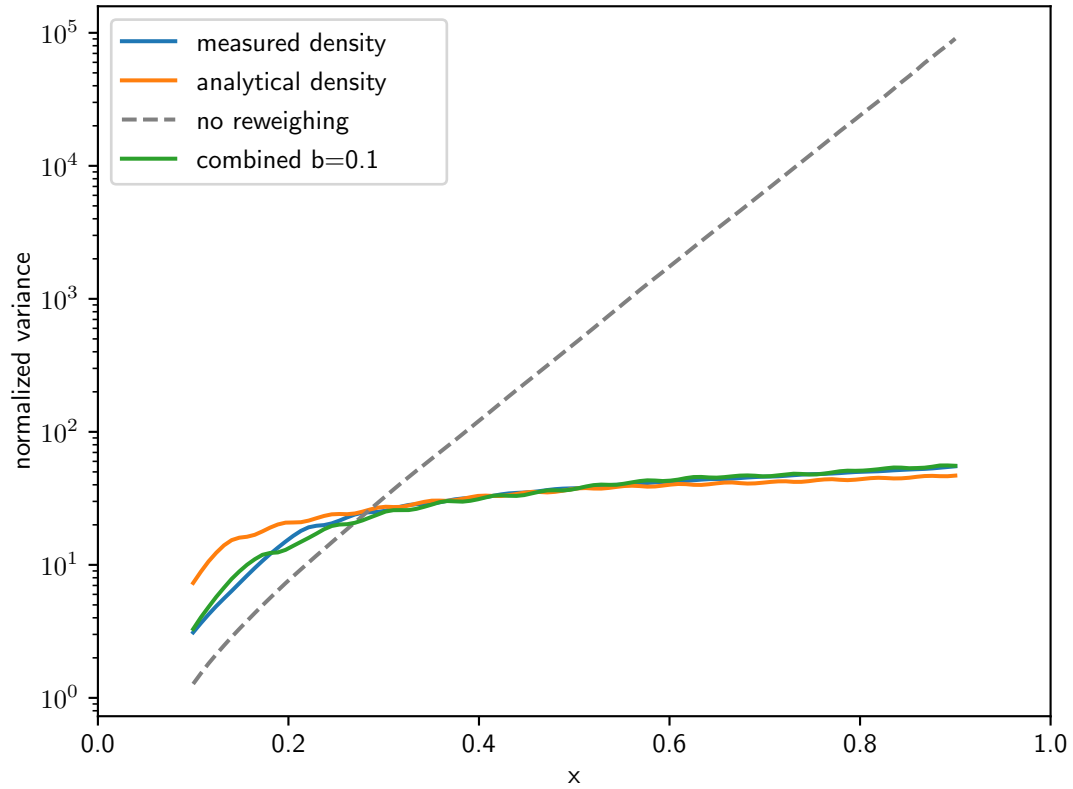


Figure 4.17.: Repetition of 4.13 with the correlated scheme and a weight ratio of 1:2. We now compare a weighting function that is proportional to the analytically known density to one that measures the density with a bin of width 0.3 and the interpolating weighting function (4.20) with $b = 0.1$. The accuracy of the weighting function proportional to the measured density values is very similar to the interpolating weighting function.

and for the proportional weight:

$$w = \rho w_0 \rho_0$$

$$m = \frac{1}{w_0 \rho_0}.$$

Here w_0 and ρ_0 are normalization constants. w_0 is chosen when selecting how many markers one wants to use in the simulation. A convenient choice of ρ_0 is $\rho_0 = \max(\rho(t=0))$. The combined weighting function is then given by equation (4.18)

$$w = w_0 \frac{\eta_1 + \eta_2}{\eta_1 + \frac{\eta_2}{\rho_0 \rho(x)}},$$

where η_1 and η_2 determine the relative importance of the constant and the proportional part of the weighting function.

The mean number of markers N is the zeroth moment of the marker distribution function:

$$\begin{aligned} N &= \int_D m(x) dx \\ &= \int_D \frac{\rho(x)}{w(x)} dx \\ &= \int_D \frac{\rho}{w_0} \frac{\eta_1 + \frac{\eta_2}{\rho_0 \rho(x)}}{\eta_1 + \eta_2} dx \\ &= \frac{1}{w_0} \int_D \rho \frac{\eta_1}{\eta_1 + \eta_2} dx + \frac{1}{w_0} \int_D \rho \frac{\frac{\eta_2}{\rho_0 \rho(x)}}{\eta_1 + \eta_2} dx \\ &= \frac{1}{w_0} \frac{\eta_1}{\eta_1 + \eta_2} \int_D \rho dx + \frac{1}{w_0} \frac{\frac{\eta_2}{\rho_0}}{\eta_1 + \eta_2} \int_D dx. \end{aligned}$$

The first integral is constant if mass is conserved, and the second one if the computational domain does not change during runtime.

4.4.2. Hysteresis, Time between Reweighting, and Weight Steps

In section 4.2.4 we found that using hysteresis reduces the variance and smooths the sharp transition between weight regions, and in section 4.3.4 we found that using many boundaries results also result in an error that does not oscillate when going to lower densities. However, using weight steps such that the distance between weight is of the same order of magnitude as the decorrelation length of the measure one is interested in, and hysteresis widths that are slightly less than the distance between weight regions, already gives good results.

We therefore set the hysteresis widths to the decorrelation length of the measure. The time between reweightings should be as large as possible, but not so large that the markers mix further than with the neighboring regions.

4. Performance of Novel Schemes

In the next step we choose the weight ratios such that the hysteresis regions do not overlap. If the continuous weighting function w_c changes with time one has to make sure that the hysteresis regions do not overlap later on. This is not a problem when the weighing function spreads out over time, for example when using the density for ICRH, which pulls out a tail of the distribution

4.5. Initialization

Depending on the problem one wants to solve, it might be easier to generate markers of equal weight according to the initial condition of the PDE, than to generate markers with varying weight and therefore according to a modified probability distribution.

I present two methods with which one can generate markers with the correct weights for the regions in phase space that they are positioned in using markers that do not have the correct weights. For the deterministic roulette we additionally have to ensure that the region the markers were created in is set such that the correct number of markers is deleted when the markers travel between regions.

The first option, which we will refer to as **simple option**, requires that the original markers have equal weight. We take one original marker and reweight it using the splitting scheme we use for reweighting during runtime, with the additional assumption that the original marker was associated with the highest weight region. Figuratively speaking we can imagine that the original marker originally was in the high weight region. Then the marker travelled in a single time step to the position it occupies now, crossing several boundaries where it was split. This means that if the marker position is in the high weight region it is not changed, if the marker position is in a lower weight region it is split once or several times. We consecutively split and add markers from the original, uniformly weighted markers until we reached the desired number of markers for the simulation. We then adjust the weight of all markers such that the total weight of the reweighted markers is the same as the total weight of the original markers without changing the weight ratios between markers.

By discarding markers the initial distribution function is represented not as accurately by the new markers than the original markers. For example consider the region with the second-highest weights: The markers we have in this region were split, meaning there are correlations between them. At the same time we probably discarded markers that were in this region. We can improve on this, let us at first neglect that we need to set the boundary where the markers will be deleted for the deterministic roulette. We choose the weight of markers in the first region, which also determines the weight in all other regions because of the prescribed weight ratios. We then consider one original marker after the other and determine the region it belongs to. We save what the total weight the original markers that belong to this region that we considered so far is. We also save how much weight our new markers that we created in this region have. We now add as many new markers at the position of the original marker we are currently considering as are required to most accurately make the total weight of the original markers and the new markers equal. If the original marker belongs to a high weight region we will not add a

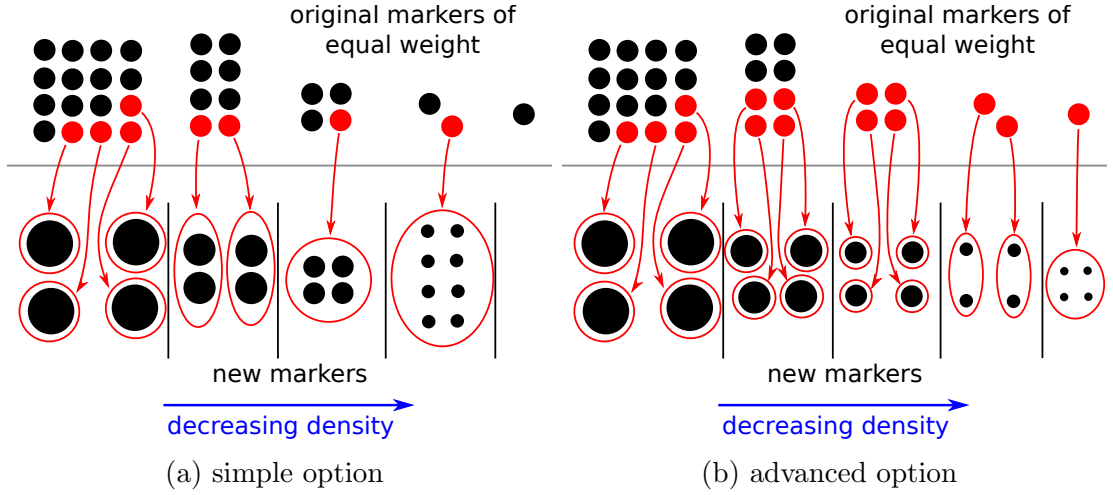


Figure 4.18.: Two sketches illustrating the two different methods for creating markers with weights corresponding to the weight regions, separated by black lines, from markers of equal weight (above the grey line). The original markers that are used for generating new markers are colored red and connected with red arrows to the new markers that are initialized at the position of the original marker. New markers that start at the same position are grouped with a red ellipse. The simple option, shown on the left, is clearly worse than the advanced option shown on the right. In all regions except the one with the highest weight the simple option neglects information about the distribution function because it ignores available original markers. Therefore more markers than necessary start at the same position. Additionally the fourth weight region wrongly contains as much weight as the third weight region, and the fifth weight region is completely void of markers. Compared to the simple option the advanced option uses all available information on the distribution function represented by the original markers and manages to conserve the total weight in each region. The supremacy of the advanced scheme appears even larger when we consider that the case shown for the simple option is a best-case scenario: While the advanced scheme always considers all original markers the simple option picks the first eight markers. Depending on how the markers are stored the first eight markers could for example all be in the first weight region. Then all new markers would also be in the first weight region.

4. Performance of Novel Schemes

marker most of the time, and if the original marker belongs to a low weight region we will create several new markers whenever an original marker belongs to a low weight region. Additionally, this method which we will refer to as **advanced option** can handle initial markers that have nonuniform weight.

We still have to set the boundary where the new markers will be deleted, provided we use the deterministic roulette. For this we first recursively calculate what fraction of markers would be deleted where assuming the markers in the low weight region started in the high weight region and were split when travelling to the lower weight regions: The markers in the first region will never be deleted. A fraction of $1/w - 1$ of the markers in the n -th region should be deleted when travelling to the next higher weight region, where w is the weight step between regions, e.g. 0.5. The rest of the markers in the n -th region have the same composition as the markers in the next higher weight region. One could use the same approach for the parameters needed by the deterministic splitting.

4.6. Implementation in ASCOT-RFOF

So far we have developed and characterized our reweighting schemes using simplified diffusion-advection equations. We will now come back to the physical problem of ICRH. We choose the parameters for reweighting with the help of section 4.4. For solving the evolution of the ion distribution function we utilize the simulation code ASCOT (version 4). ASCOT is a Monte Carlo orbit following code for solving the kinetic equation of motion of plasma particles [8]. Orbit following denotes that the Monte Carlo markers follow the same equation of motion for the deterministic advective part as the physical particles. For the collisions it is assumed that the simulated particles collide only with the background plasma, allowing the use of a Fokker-Planck collision operator like the one we introduced in section 2.3. The RF interaction of the particles is treated by the code library RFOF [9]. RFOF predicts when a marker will be in resonance and calculates the stochastic change in the marker's phase space position when it passes the resonance. The effect of passing the resonance depends on the RF wave amplitude. But we prescribe the absorbed heating power and not the wave amplitude. RFOF handles this by adjusting the wave amplitude such that the prescribed power is absorbed, this adjustment is called normalization. Because of normalization the markers cannot be advanced independent of each other, and a minimum number of markers is required such that the absorbed power can be calculated with sufficient accuracy.

As test case we consider the simulation setup used by Sipilä et. al. to simulate the signal of the fast ion loss detector [4]. It is based on the discharge #33147 of the ASDEX Upgrade tokamak [27] at time $t = 1.0$ s. Temperature and density profiles¹ are shown in figure 4.19, other key parameters of this diverted² H mode³ discharge are:

¹ ρ_{pol} is a measure for the distance to the center of the plasma [28].

²When the plasma is diverted, particles that are further away from the center than $\rho_{\text{pol}} = 1$ are guided away from the main plasma to the divertor [12].

³When the heating power is increased above a certain threshold, turbulent transport of particles and heat from the center of the plasma outwards suddenly drops at $\rho_{\text{pol}} \approx 1$ [29]. The plasma is now in the 'high confinement mode' or H mode. The density and the temperature increase abruptly at

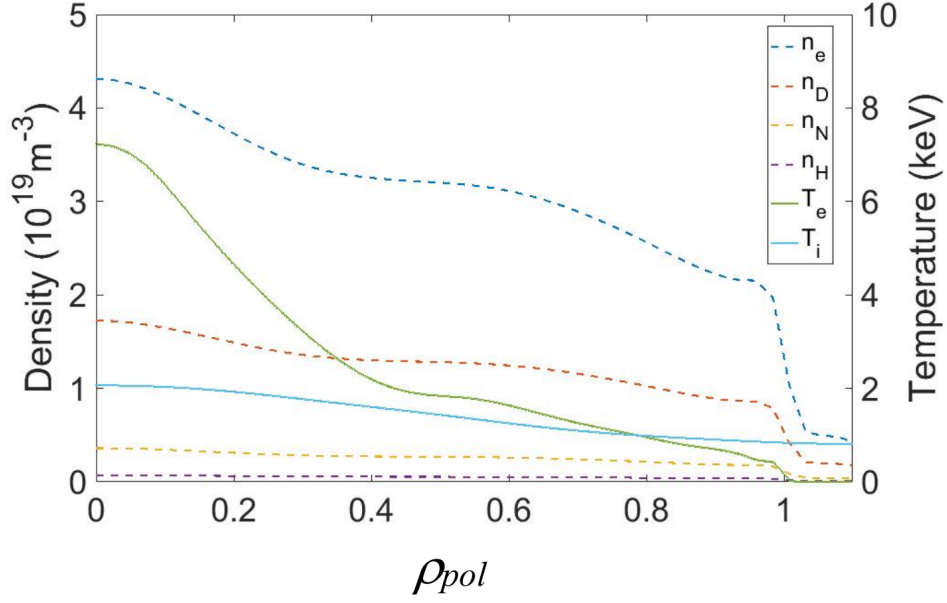


Figure 4.19.: The background density and temperature profiles used in the simulation of ASDEX Upgrade discharge #33147. The figure is taken from Sipilä et. al. [4].

B_t	I_p	NBI	ICRH	ECRH
-2.5 T	700 kA	2.6 MW	3.6 MW	1.3 MW

Here B_t is the toroidal magnetic field, I_p is the current flowing in the plasma, and NBI⁴ and ECRH⁵ are auxilliary heating methods.

4.6.1. Setup for Reweighting

For reweighting we choose a continuous weighting function w_c that is proportional to the number density in phase space. We achieve this by using an automatic weighting function as described in section 4.4.1.1, with a histogram to estimate the phase space density. The first 10 of 20 bins of the histogram have a width of only 10 keV, the next 10 bins a width of 200 keV. The first bins are smaller because we have better statistics due to higher phase space densities, and the thermal bulk of the distribution varies more quickly as a function of energy than in the high energy tail of the distribution. To initialize our markers we use the *advanced option* from section 4.5.

$\rho_{pol} \approx 1$ due to the better insulation, as shown in figure 4.19. The origin of the H mode, which was discovered at ASDEX Upgrade, is still not understood [29].

⁴Neutral Beam Injection (NBI) is a method of heating the plasma. Deuterium atoms with energies in the order of 100 keV are injected into the plasma and deposit their energy by collisions [1]. The atoms have to be neutral, otherwise they would be deflected by the magnetic field from the coils.

⁵Electron Cyclotron Resonance Heating: Similar to ICRH, but uses absorption by electrons instead of by ions [1].

4. Performance of Novel Schemes

We use a histogram with a bin width of 5 keV to analyze the results. As we saw in figure 4.14 the weight ratio between regions does not have a large impact on the performance of the schemes, we choose 1 : 3. With an initial ion temperature T_i of ≈ 1 keV the regions have a width of $-T_i \ln \frac{1}{3} \approx 1.1$ keV. We now need to estimate the diffusion coefficient in energy space to set the hysteresis width and the time between reweightings. Helander [7] gives a Focker-Planck type diffusion operator in equation (3.40). The necessary coefficient, the effective parallel collision frequency, is given in equations (3.47) and (3.48). We change from speed to energy as coordinate and find $D \approx 5 \times 10^9 \text{ eV}^2 \text{ s}^{-1}$ for the diffusion coefficient in energy space for ions with velocities that equal the thermal velocity. With equation (4.6), in which we normalized t with the diffusion coefficient, we can express the typical mixing length due to non-instantaneous reweighting as function of the time between reweightings. We find that with $\Delta t \approx 7 \mu\text{s}$ the mixing length is approximately equal to the region size. If we were to reweight this often, reweighting would use a significant portion of computational time. Because the diffusion coefficient decreases strongly when going to higher energies we can wait longer between reweightings, we set the time between reweightings to 0.15 ms. Except for the first three bins, where the time-discrete reweighting avoids unnecessary crossings due to the large diffusion coefficient, we avoid unnecessary crossings by using a hysteresis with a width of 80 % of the region width. We choose 80 % because that smoothes the transition between weight regions (section 4.2.3) but avoids mixing of markers across several regions. The final decision to take is how many regions we want to use. We are, for example, interested in densities that are $\approx 5 \times 10^{-5}$ times smaller than densities in the thermal bulk. When we choose 10 regions, the last region begins when the phase space density is a factor of $(1/3)^9 \approx 5.1 \times 10^{-5}$ lower than the density at the lowest energy resolved by the histogram for the automatic weighting function. When one is interested in lower densities more regions can be used.

4.6.2. Simulation Results

In figure 4.20 we show the hydrogen distribution function f , calculated using the markers of the simulation after 0.1 s elapsed since the start of the simulation. It was not yet possible to adapt the ASCOT routine that replaces particles lost in wall collisions for cases with non-uniformly weighted markers. Deactivating the replacement of particles leads to a decline of the particle number in the course of the simulation compared to the 500 000 marker simulation from Sipilä et. al. [4]. We therefore scale f such that we can nevertheless compare the simulation results.

The baseline scenario is using no reweighting and 500 000 markers (blue). At low energies f decreases rapidly, it is close to thermal distribution with $T_i \approx 1$ keV. Going to higher energies the distribution flattens and we see the high energy tail we expect based on our discussions in chapter 2. As the density decreases the noise in the result increases because fewer markers are inside the bins of the histogram. When we decrease the number of markers to 7 500 (purple) too few markers fulfill the resonance condition and normalizing the wave field cannot be done accurately enough. The distribution does not develop a strong tail.

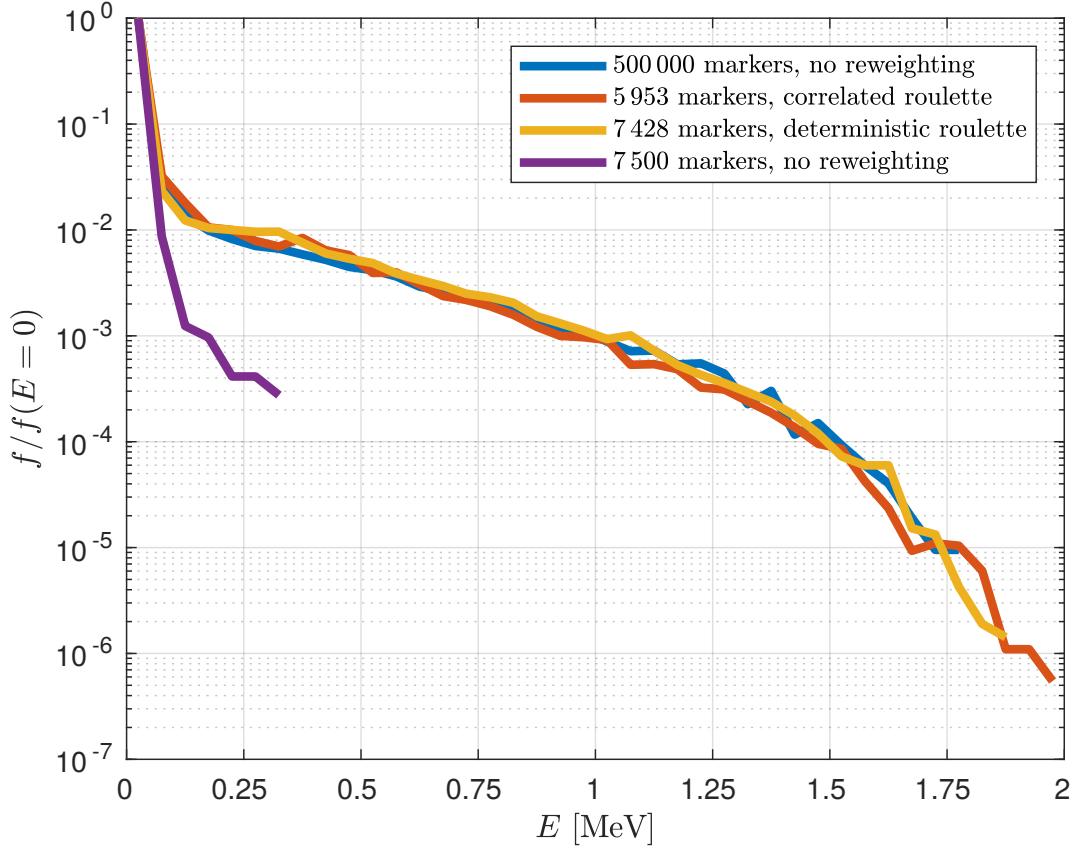


Figure 4.20.: The scaled distribution function of hydrogen ions as a function of energy yielded by the ASCOT-RFOF simulation based on ASDEX Upgrade discharge #33147. f is reconstructed from the markers with a histogram with 40 bins. When we use 500 000 markers without reweighting (blue) a high energy tail develops. When we reduce the number of markers to 7 500 (purple) normalizing the RF field cannot be done reliably anymore because the resonance condition is not met often enough, resulting in too much noise in the absorbed RF power. When we use reweighting (red and yellow) already $\approx 6\,000$ markers are enough to get reasonable statistics for the absorbed RF power and a high energy tail develops. The tail is steeper, and the distribution therefore colder, than for the simulation with 500 000 markers. The distribution is presumably colder because the statistics for the absorbed RF power are still not good enough with only 6 000 markers or because ASCOT has problems with such low marker numbers. With $\approx 7\,500$ markers the discrepancy to the blue curve is already smaller than with $\approx 6\,000$ markers. It is improbable that reweighting itself is the cause of the steeper decline because we did not find any sign that reweighting influences the mean of the simulation in sections 4.3.4.2 and 4.15. The figure shows that with reweighting we can get good results, especially at high energies, with significantly fewer markers than without reweighting.

4. Performance of Novel Schemes

The red and yellow lines show the distribution function obtained with only 5 953 and 7 428 markers with reweighting. Other than for the case without reweighting and 7 500 markers a high energy tail develops. This is presumably because we resolve the trapped particles with a turning point close to the resonance, which we introduced in section 2.3.1.1, better with reweighting (red and yellow curves) than without (purple curve). This would suggest that we have better statistics on the absorbed RF power, enabling more accurate normalization of the RF wave field.

The high energy tail of the curves with reweighting is however steeper and therefore colder than the high energy tail of the simulation with 500 000 markers. The difference is apparent between 250 keV and 750 keV. In sections 4.3.4.2 and 4.15 we did not find any sign that reweighting influences the mean of the simulations. The explanations we deem most probable are that $\approx 7\,500$ markers are still too little for accurate normalization of the wave field or that there exist problems with the handling of small marker numbers by ASCOT. Using $\approx 7\,500$ markers already gives a distribution function closer to the blue line than the $\approx 6\,000$ marker case. Unfortunately time restrictions did not allow for simulations with reweighting and more markers.

Next we will quantitatively compare the fluctuations of the density estimates given by the simulations. We will not include the case with 7 500 markers without reweighting because it fails to develop a high energy tail that we could investigate. In sections 4.2, 4.3 and 4.4 we determined the variance of the results by using many datapoints. We cannot do this for the heating process because each simulation requires more than 10 orders of magnitude more computational time.

We can nevertheless estimate the relative error the simulations yield. We increase the number of bins for the histogram to 2 000, separating each bin from figure 4.20 into 50 subdivisions. We assume that the distribution function changes only slightly across 50 keV and calculate the mean and variance of the density estimates of the 50 subdivisions of each bin. The definitions of the mean and variance are given by equations (3.4) and (3.5). Observe that this does not give an estimate for the fluctuations of the density estimate using the histogram used in figure 4.20 with 40 bins but for a histogram with $40 \cdot 50 = 2\,000$ bins.

As before we will compare the simulations using the normalized variance: the variance divided by the squared mean and multiplied by the marker number. By dividing by the squared mean we obtain a measure for the relative error. By multiplying by the number of markers we finally get a measure for the inverse relative accuracy we obtain per marker we invest in the simulation.

The obtained normalized variance is shown left in figure 4.21. At low energies f changes rapidly, violating the assumption that f changes only little over 50 keV. We therefore only consider $E \geq 0.2$ MeV. With reweighting the accuracy in the high weight region should be increased. The high weight region is however only a few keV wide and not resolved in figure 4.21. Therefore reweighting appears superior everywhere, already at $E = 0.2$ MeV reweighting leads to a considerably lower error of the estimate for f . As shown on the right of figure 4.21 this advantage grows to a factor of ≈ 100 at $E > 1.3$ MeV.

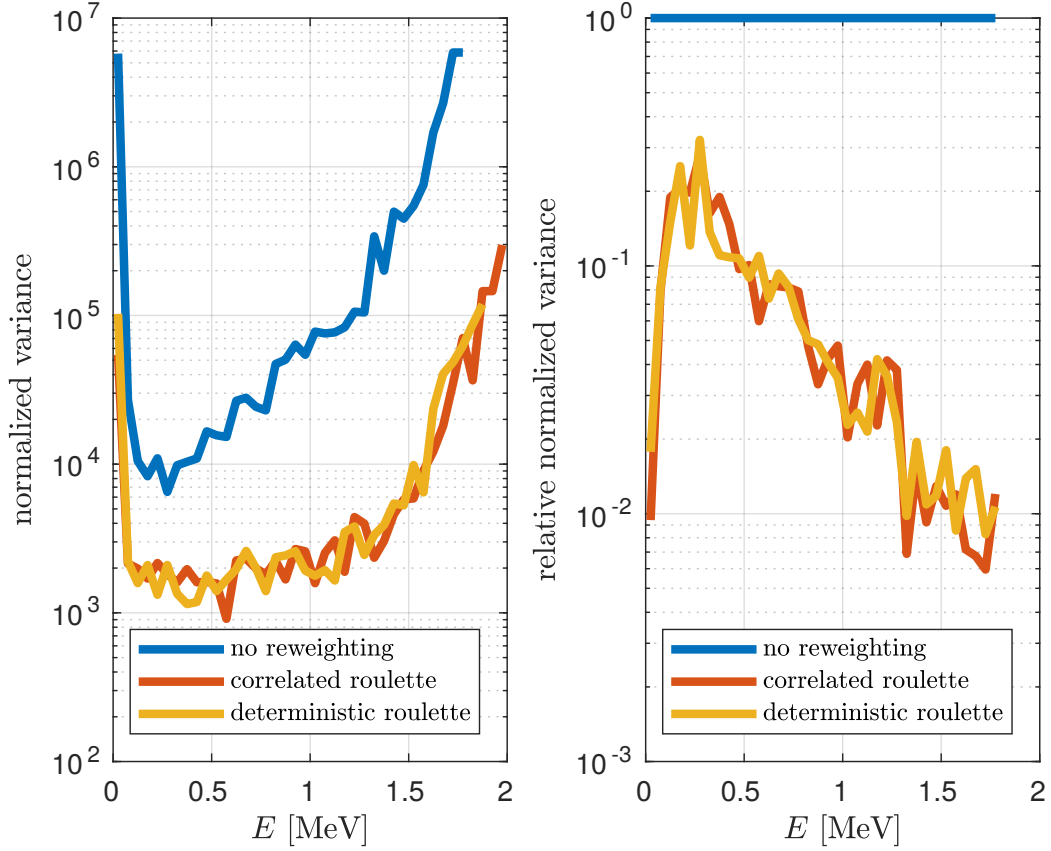


Figure 4.21.: On the left we show the normalized variance for simulations without reweighting and with reweighting using the correlated and the deterministic roulette. For small energies the distribution function, shown in figure 4.20, changes rapidly. Therefore our assumption that the distribution function changes little over 50 keV is invalid, and we get wrong estimates for the amplitude of the fluctuations. Considering only $E \geq 0.2$ MeV we find that, as expected, the fluctuations increase when going to higher energies and lower densities. With reweighting, regardless which roulette we use, the normalized variance at 0.2 MeV is by a factor of ≈ 5 smaller than without reweighting. With reweighting the normalized variance additionally increases much slower than without reweighting. We reweight until the density declines by a factor of $(1/3)^9 \approx 5.1 \times 10^{-5}$. This reduction in density is reached at $E \approx 1.3$ MeV. This is consistent with the faster growth of variance at $E > 1.3$ MeV. On the right we show the same data as on the left, but divide all curves by the normalized variance without reweighting. The variance that we find at $E > 1.3$ MeV is reduced by a factor of ≈ 100 when one uses reweighting. Provided enough markers for normalization are used one can therefore reduce the computational time needed for this simulation by a factor of 100.

4. *Performance of Novel Schemes*

The simulation with 500 000 markers requires $\approx 44\,000$ core hours [4]. Reducing this to ≈ 440 core hours constitutes a substantial improvement. In the current implementation reweighting is not yet parallelized. Because the markers from all 256 cpu cores are reweighted on a single core the execution time is not reduced to 440 core hours but only to $\approx 3\,600$ core hours. The markers can be reweighted independently, except for few communications for the automatic weighting function. Parallelizing the reweighting routines should give access to the full reduction in execution time.

5. Conclusion

The ion distribution function in a fusion plasma evolves according to the Boltzmann equation (2.2), which can be approximated as a diffusion-advection equation. Solutions of diffusion-advection equations can be found using Monte Carlo markers that follow a corresponding stochastic differential equation, the Langevin equation. Without further measures these markers have equal weight and are distributed according to the solution of the diffusion-advection equation. When the solution of the equation is small in a region of phase space, only few markers are inside this region. Therefore this region is not well resolved and we can only obtain inaccurate information about it.

With Ion Cyclotron Resonance Heating (ICRH) a high energy tail of the distribution function, which is only sparsely populated compared to the thermal bulk, develops. The ions in this tail are important. They for example allow the investigation of the confinement of fast ions [4, 5]. The plasma in a fusion reactor with magnetic confinement is envisioned to be heated mostly by energetic, fusion-born α -particles, which need to be confined adequately.

We presented methods that enable us to resolve such regions with a small distribution function accurately. With them we can choose the marker weights freely for different regions of configuration space. When travelling between those regions markers are duplicated or deleted, and their weights are adjusted. Such roulette methods are not new, but we introduced additional measures to avoid detrimental effects. By not using uncorrelated random numbers for the deletion procedure, or roulette, we can avoid growing fluctuations in the total weight and therefore the distribution function. Because of how the methods are constructed this does not alter the expectation value of the distribution function. Compared to importance sampling for integrals two complications arise:

- After duplicating a marker the original marker and the duplicate have to decorrelate to improve statistics. This decorrelation occurs due to the stochastic part in their equation of motion.
- The number of markers in a region is determined by the initial number of markers in the region and the flux of markers into and out of the region. If the adjacent regions are poorly resolved the flux from particles into the region contains a lot of noise. The number of markers in the region then fluctuates strongly. This limits the accuracy one can achieve by using markers of little weight in a region.

The splitting of markers only requires additional measures, which we proposed, when one uses non-integer weight ratios between regions. Determining which markers to delete when they enter another region is always necessary. With a FORTRAN 2003 program we investigated two options for the deletion procedure:

5. Conclusion

Correlated roulette: The number of markers that are deleted is ensured to be close to the number prescribed by the weight ratio. The deviation is of constant order and does not increase with time or with the number of markers used in the simulation. This is achieved with correlated random number generators, hence the name.

Deterministic roulette: The markers that originally entered the region from a region with higher weight will not be deleted when they leave the region again. But the markers that were created when these markers entered the region are deleted when they advance to a region of higher weight. On one hand the deterministic roulette does not conserve the total weight as accurately as the correlated roulette. On the other hand it ensures that markers in the high weight region are not negatively influenced by any fluctuations in the deletion procedure.

For only two regions the correlated roulette leads to more accurate results, at least in one dimensions. When going to density functions that vary over several orders of magnitude and more regions both methods perform similarly. Because the deterministic roulette appears to be more robust in more than one dimension we recommend using it over the correlated roulette. To avoid multiple consecutive crossings we also propose using a hysteresis and to not necessarily reweight after every time step. We also showed how to place the weight regions and choose the parameters required for reweighting.

In a test case with a one dimensional diffusion-advection equation with constant coefficients we demonstrated the advantages of reweighting. We kept the variance of a binned density measurement in the same order of magnitude while the density decreased by a factor of 2.4×10^{-5} . To achieve the same accuracy without reweighting as with reweighting for the lowest density region one would have to use $1800\times$ as many markers. Finally we also implemented our reweighting schemes in the code ASCOT-RFOF. Using a simulation set up by Sipilä et. al. [4] we showed that the runtime needed to obtain accurate solutions for the ion distribution function with ICRH can be reduced by a factor of 100 with reweighting.

In this thesis we developed and characterized reweighting methods that can accelerate Monte Carlo simulations of some diffusive-advective processes by orders of magnitude. The schemes are parallelizable and do not require extensive modifications of the code, provided the code can handle markers of nonuniform weight and a during runtime fluctuating number of markers.

Bibliography

- [1] Jeffrey P Freidberg. *Plasma physics and fusion energy*. Cambridge university press, 2008. ISBN: 978-0-521-73317-5.
- [2] Thomas H Stix. *Waves in plasmas*. Springer Science & Business Media, 1992. ISBN: 978-0-88318-859-0.
- [3] John Killeen et al. *Computational methods for kinetic models of magnetically confined plasmas*. Springer Science & Business Media, 2012. ISBN: 978-3-642-85954-0.
- [4] S Sipilä et al. “Monte Carlo ion cyclotron heating and fast ion loss detector simulations in ASDEX Upgrade”. In: *45th EPS Conference on Plasma Physics*. European Physical Society. 2018. ISBN: 978-1-5108-6844-1.
- [5] Manuel Garcia-Munoz et al. “MHD induced fast-ion losses on ASDEX Upgrade”. In: *Nuclear Fusion* 49.8 (2009), p. 085014. DOI: [10.1088/0029-5515/49/8/085014](https://doi.org/10.1088/0029-5515/49/8/085014).
- [6] WW Heidbrink. “Basic physics of Alfvén instabilities driven by energetic particles in toroidally confined plasmas”. In: *Physics of Plasmas* 15.5 (2008), p. 055501. DOI: [10.1063/1.2838239](https://doi.org/10.1063/1.2838239).
- [7] Per Helander and Dieter J Sigmar. *Collisional transport in magnetized plasmas*. Vol. 4. Cambridge University Press, 2005. ISBN: 978-0-521-80798-2.
- [8] Eero Hirvijoki et al. “ASCOT: Solving the kinetic equation of minority particle species in tokamak plasmas”. In: *Computer Physics Communications* 185.4 (2014), pp. 1310–1321. DOI: [10.1016/j.cpc.2014.01.014](https://doi.org/10.1016/j.cpc.2014.01.014).
- [9] Thomas Johnson et al. “Library for RF interactions in orbit following codes”. In: *AIP Conference Proceedings*. Vol. 1406. 1. AIP. 2011, pp. 373–376. DOI: [10.1063/1.3664996](https://doi.org/10.1063/1.3664996).
- [10] Peter J Mohr, David B Newell, and Barry N Taylor. “CODATA recommended values of the fundamental physical constants: 2014”. In: *Journal of Physical and Chemical Reference Data* 45.4 (2016), p. 043102. DOI: [10.1063/1.4954402](https://doi.org/10.1063/1.4954402).
- [11] PH Rebut et al. “The ITER challenge”. In: *Fusion Engineering and Design* 22.1-2 (1993), pp. 7–18. DOI: [10.1016/S0920-3796\(05\)80004-3](https://doi.org/10.1016/S0920-3796(05)80004-3).
- [12] Ulrich Stroth. *Plasmaphysik: Phänomene, Grundlagen, Anwendungen*. Springer-Verlag, 2011. DOI: [978-3-8348-1615-3](https://doi.org/978-3-8348-1615-3).
- [13] K Ikeda. “Progress in the ITER physics basis”. In: *Nuclear Fusion* 47.6 (2007). DOI: [10.1088/0029-5515/47/6/E01](https://doi.org/10.1088/0029-5515/47/6/E01).
- [14] J. Wesson and D.J. Campbell. *Tokamaks*. Oxford Engineering Science Series. Clarendon Press, 1997. ISBN: 978-0-19-856293-1.

Bibliography

- [15] Mitsuru Kikuchi, Karl Lackner, and Minh Quang Tran. *Fusion physics*. 2012. ISBN: 978-92-0-130410-0.
- [16] Frédéric Cérou and Arnaud Guyader. “Adaptive multilevel splitting for rare event analysis”. In: *Stochastic Analysis and Applications* 25.2 (2007), pp. 417–443. DOI: [10.1080/07362990601139628](https://doi.org/10.1080/07362990601139628).
- [17] Thomas Howard Stix. “Fast-wave heating of a two-component plasma”. In: *Nuclear Fusion* 15.5 (1975), p. 737. DOI: [10.1088/0029-5515/15/5/003](https://doi.org/10.1088/0029-5515/15/5/003).
- [18] G. D. Kerbel and M. G. McCoy. “Kinetic theory and simulation of multispecies plasmas in tokamaks excited with electromagnetic waves in the ion-cyclotron range of frequencies”. In: *The Physics of Fluids* 28.12 (1985), pp. 3629–3653. DOI: [10.1063/1.865319](https://doi.org/10.1063/1.865319).
- [19] Achim Klenke. *Probability theory: a comprehensive course*. Springer Science & Business Media, 2013. ISBN: 978-1-4471-5360-3.
- [20] William Press et al. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 2007. ISBN: 978-0-521-88068-8.
- [21] B. S. Everitt and A. Skrondal. *The Cambridge dictionary of statistics*. Cambridge University Press, 2010. ISBN: 978-0-511-78827-7.
- [22] Setsuo Ichimaru. *Basic principles of plasma physics: a statistical approach*. CRC Press, 2018. ISBN: 978-0-8133-4178-1.
- [23] Paul Embrechts. *Selfsimilar processes*. Vol. 21. Princeton University Press, 2009. ISBN: 978-0-691-09627-8.
- [24] Robert Martin, Justin Koo, and David Bilyeu. *Dynamic Particle Weight Remapping in Hybrid PIC Hall-effect Thruster Simulation*. Tech. rep. Accessed on 2019-04-19. AIR FORCE RESEARCH LAB EDWARDS AFB CA ROCKET PROPULSION DIV, 2015. URL: <https://apps.dtic.mil/docs/citations/ADA627145>.
- [25] TJ Boyd, TJM Boyd, and JJ Sanderson. *The physics of plasmas*. Cambridge University Press, 2003. ISBN: 978-0-521-45912-9.
- [26] Merran Evans, Nicholas Hastings, and Brian Peacock. *Statistical distributions*. Wiley-Interscience, 2000. ISBN: 978-0-471-37124-3.
- [27] D. Aguiam et al. “Overview of ASDEX upgrade results”. In: *Nuclear Fusion* 57.10 (2017). DOI: [10.1088/1741-4326/aa64f6](https://doi.org/10.1088/1741-4326/aa64f6).
- [28] William D D’haeseleer et al. *Flux coordinates and magnetic field structure: a guide to a fundamental tool of plasma theory*. Springer Science & Business Media, 2012. ISBN: 978-3-642-75597-2.
- [29] Fritz Wagner. “A quarter-century of H-mode studies”. In: *Plasma Physics and Controlled Fusion* 49.12B (2007), B1. DOI: [10.1088/0741-3335/49/12B/S01](https://doi.org/10.1088/0741-3335/49/12B/S01).

A. Decorrelation Time

A.1. Definitions and Setup

We consider a simple Brownian motion:

$$\Delta x = k\xi\sqrt{\Delta t},$$

where ξ is a normally distributed random variable with mean 0 and variance 1. $k\xi\sqrt{\Delta t}$ is therefore also a normally distributed random variable with mean 0 and variance $k^2\Delta t$.

$$x(t) = \sum_{i=0}^{n(t)} k\xi(t)\sqrt{\Delta t},$$

with $n = \frac{t}{\Delta t}$. The probability distribution of $x(t)$ is

$$P(x, t) = \frac{1}{\sqrt{2tk^2}} \exp\left(-\frac{(x - x_0)^2}{2tk^2}\right),$$

using the central limit theorem. x_0 is the starting position.

When we restrict x to be in the interval $[0, L[$ and use periodic boundary conditions we split the distribution into segments and add them up:

$$P(x, t) = \frac{1}{\sqrt{2tk^2}} \sum_{i=-\infty}^{\infty} \exp\left(-\frac{(x + iL - x_0)^2}{2tk^2}\right).$$

This sum is absolute convergent, therefore we can rearrange the summands in any way we like.

We will calculate the mean number and variance of two particles in the interval $I = [a, b[$ that were started at the same initial position x_0 . $l = b - a$. We will use the characteristic function of I

$$\mathbb{1}_I(x) = \begin{cases} 1 & \text{if } x \in I \\ 0 & \text{otherwise} \end{cases}$$

Observe that $\mathbb{1}_I^2 = \mathbb{1}_I$.

A.2. Mean Value

$$\langle n_2 \rangle = \frac{1}{L} \int_0^L dx_0 \int dx dy P(x)P(y) [\mathbb{1}_I(x) + \mathbb{1}_I(y)],$$

the integral over x_0 and the division by L correspond to averaging over all possible starting positions. We start by exchanging x and y for the second term

$$\begin{aligned} \langle n_2 \rangle &= \frac{2}{L} \int_0^L dx_0 \int dx dy P(x)P(y) \mathbb{1}_I(x) \\ &= \frac{2}{L} \int_0^L dx_0 \int dx P(x) \mathbb{1}_I(x) \\ &= \frac{2}{L} \int_0^L dx_0 \int dx \frac{1}{\sqrt{2tk^2}} \sum_{i=-\infty}^{\infty} \exp\left(-\frac{(x + iL - x_0)^2}{2tk^2}\right) \mathbb{1}_I(x). \end{aligned}$$

We first perform the x_0 integration, making the x integration trivial:

$$\begin{aligned} \langle n_2 \rangle &= \frac{2}{L} \int dx \mathbb{1}_I(x) \\ &= \frac{2l}{L}, \end{aligned}$$

this result is independent of time.

A.3. Limiting Cases

A.3.1. $t = 0$

At $t = 0$ the probability distributions are two delta distributions:

$$\begin{aligned} \langle n_2^2 \rangle &= \frac{1}{L} \int_0^L dx_0 \int dx dy \delta(x - x_0) \delta(y - x_0) [\mathbb{1}_I(x) + \mathbb{1}_I(y)]^2 \\ &= \frac{1}{L} \int_0^L dx_0 [\mathbb{1}_I(x_0) + \mathbb{1}_I(x_0)]^2 \\ &= \frac{4}{L} \int_0^L dx_0 \mathbb{1}_I(x_0) \\ &= \frac{4l}{L} \end{aligned}$$

The variance is

$$\begin{aligned} \sigma_2(t = 0) &= \langle n_2^2 \rangle - \langle n_2 \rangle^2 \\ &= 4 \left(\frac{l}{L} - \frac{l^2}{L^2} \right). \end{aligned}$$

A.3.2. $t \rightarrow \infty$

At $t \rightarrow \infty$ the probability distributions for both particles are uniform.

$$\begin{aligned}
\langle n_2^2 \rangle &= \frac{1}{L} \int_0^L dx_0 \int dx dy \frac{1}{L^2} [\mathbb{1}_I(x) + \mathbb{1}_I(y)]^2 \\
&= \frac{1}{L^3} \int_0^L dx_0 \int dx dy [\mathbb{1}_I(x) + \mathbb{1}_I(y)]^2 \\
&= \frac{2}{L^3} \int_0^L dx_0 \int dx dy [\mathbb{1}_I(x) + \mathbb{1}_I(x)\mathbb{1}_I(y)] \\
&= \frac{2l}{L} + \frac{2l^2}{L^2}
\end{aligned}$$

The variance is now

$$\begin{aligned}
\sigma_2(t \rightarrow \infty) &= \langle n_2^2 \rangle - \langle n_2 \rangle^2 \\
&= 2\frac{l}{L} + 2\frac{l^2}{L^2} - 4\frac{l^2}{L^2} \\
&= 2\left(\frac{l}{L} - \frac{l^2}{L^2}\right) \\
&= \frac{1}{2}\sigma_2(t=0)
\end{aligned}$$

A.4. Variance for $t > 0$

$$\begin{aligned}
\langle n_2^2 \rangle &= \frac{1}{L} \int_0^L dx_0 \int dx dy P(x)P(y) [\mathbb{1}_I(x) + \mathbb{1}_I(y)]^2 \\
&= \frac{2}{L} \int_0^L dx_0 \int dx dy P(x)P(y) [\mathbb{1}_I(x) + \mathbb{1}_I(x)\mathbb{1}_I(y)] \\
&= \frac{2l}{L} + \frac{1}{Lk^2t} \int dx dy dx_0 \sum_{i,j} e^{-\frac{1}{2k^2t}[(x+iL-x_0)^2+(y+jL-x_0)^2]} \mathbb{1}_I(x)\mathbb{1}_I(y) \\
&= \frac{2l}{L} + \frac{1}{Lk^2t} \int dx dy dx_0 \sum_{i,j} e^{-\frac{1}{k^2t}[\frac{1}{4}(x-y+L(i-j))^2+(x_0-\frac{1}{2}(x+y+L(n+m)))^2]} \mathbb{1}_I(x)\mathbb{1}_I(y) \\
&= \frac{2l}{L} + \frac{1}{Lk^2t} \int dx dy dx_0 \sum_{o,p} e^{-\frac{1}{k^2t}[\frac{1}{4}(x-y+Lo)^2+(x_0-\frac{1}{2}(x+y+Lp))^2]} \mathbb{1}_I(x)\mathbb{1}_I(y).
\end{aligned}$$

In the last two steps we rearranged the exponent and changed our indices. We now can easily factor out the exponential and get two sums, one of them independent of x_0 . The other one is a normal distribution with x_0 as variable, the sum is merely the segmentation for the periodic boundaries.

A. Decorrelation Time

$$\begin{aligned}\langle n_2^2 \rangle &= \frac{2l}{L} + \frac{1}{L\sqrt{k^2t}} \int dx \, dy \sum_o e^{-\frac{1}{4k^2t}(x-y+Lo)^2} \mathbb{1}_I(x) \mathbb{1}_I(y) \\ &= \frac{2l}{L} + \frac{1}{L\sqrt{k^2t}} \int dy \sum_o \sqrt{\pi k^2t} \left[\operatorname{erf}\left(\frac{y-a-Lo}{\sqrt{4k^2t}}\right) - \operatorname{erf}\left(\frac{y-b-Lo}{\sqrt{4k^2t}}\right) \right] \mathbb{1}_I(y)\end{aligned}$$

Now we have to integrate over the two error functions. The antiderivative of the error function can be computed using integration by parts.

$$\begin{aligned}\langle n_2^2 \rangle &= \frac{2l}{L} + \\ &+ \frac{1}{L} \sum_{o=-\infty}^{\infty} \left[-(a-b+Lo) \operatorname{erf}\left(\frac{-a+b-Lo}{\sqrt{4k^2t}}\right) + \sqrt{\frac{4k^2t}{\pi}} \left(e^{-\frac{(a-b+Lo)^2}{4k^2t}} - e^{-\frac{(Lo)^2}{4k^2t}} \right) - Lo \operatorname{erf}\left(\frac{Lo}{\sqrt{4k^2t}}\right) \right. \\ &\quad \left. - (-a+b+Lo) \operatorname{erf}\left(\frac{a-b-Lo}{\sqrt{4k^2t}}\right) - \sqrt{\frac{4k^2t}{\pi}} \left(e^{-\frac{(Lo)^2}{4k^2t}} - e^{-\frac{(-a+b+Lo)^2}{4k^2t}} \right) - Lo \operatorname{erf}\left(\frac{Lo}{\sqrt{4k^2t}}\right) \right].\end{aligned}$$

We see that the two parts of the sum are identical by renaming $o \rightarrow -o$ in the second one. We arrive at

$$\begin{aligned}\langle n_2^2 \rangle &= \frac{2l}{L} + \\ &+ \frac{2}{L} \sum_{o=-\infty}^{\infty} \left[-(-l+Lo) \operatorname{erf}\left(\frac{l-Lo}{\sqrt{4k^2t}}\right) + \sqrt{\frac{4k^2t}{\pi}} \left(e^{-\frac{(-l+Lo)^2}{4k^2t}} - e^{-\frac{(Lo)^2}{4k^2t}} \right) - Lo \operatorname{erf}\left(\frac{Lo}{\sqrt{4k^2t}}\right) \right].\end{aligned}$$

The variance is then

$$\begin{aligned}\operatorname{Var}(t) &= \frac{2l}{L} - \frac{4l^2}{L^2} + \\ &+ \frac{2}{L} \sum_{o=-\infty}^{\infty} \left[-(-l+Lo) \operatorname{erf}\left(\frac{l-Lo}{\sqrt{4k^2t}}\right) + \sqrt{\frac{4k^2t}{\pi}} \left(e^{-\frac{(l-Lo)^2}{4k^2t}} - e^{-\frac{(Lo)^2}{4k^2t}} \right) - Lo \operatorname{erf}\left(\frac{Lo}{\sqrt{4k^2t}}\right) \right].\end{aligned}$$

To recover the limiting cases, the limits for $t \rightarrow 0$ and $t \rightarrow \infty$ have to be taken. Here one must be careful because pulling the limit into the infinite sum is not always permitted. When we nevertheless do this we recover the value for $t \rightarrow 0$. This is not the case for $t \rightarrow \infty$. Here, all terms tend to 0 while the sum approaches $\frac{l^2}{L}$.

This behavior has a physical interpretation. We separated the real axis into intervals of length L . The probability of finding a marker in one interval tends to 0 because it diffuses to $\pm\infty$, but of course the probability of finding the marker anywhere does not.

A.5. Comparison to Simulation

We compare our analytical result to a simulation of the process, the numerical data is averaged over 100 000 runs.

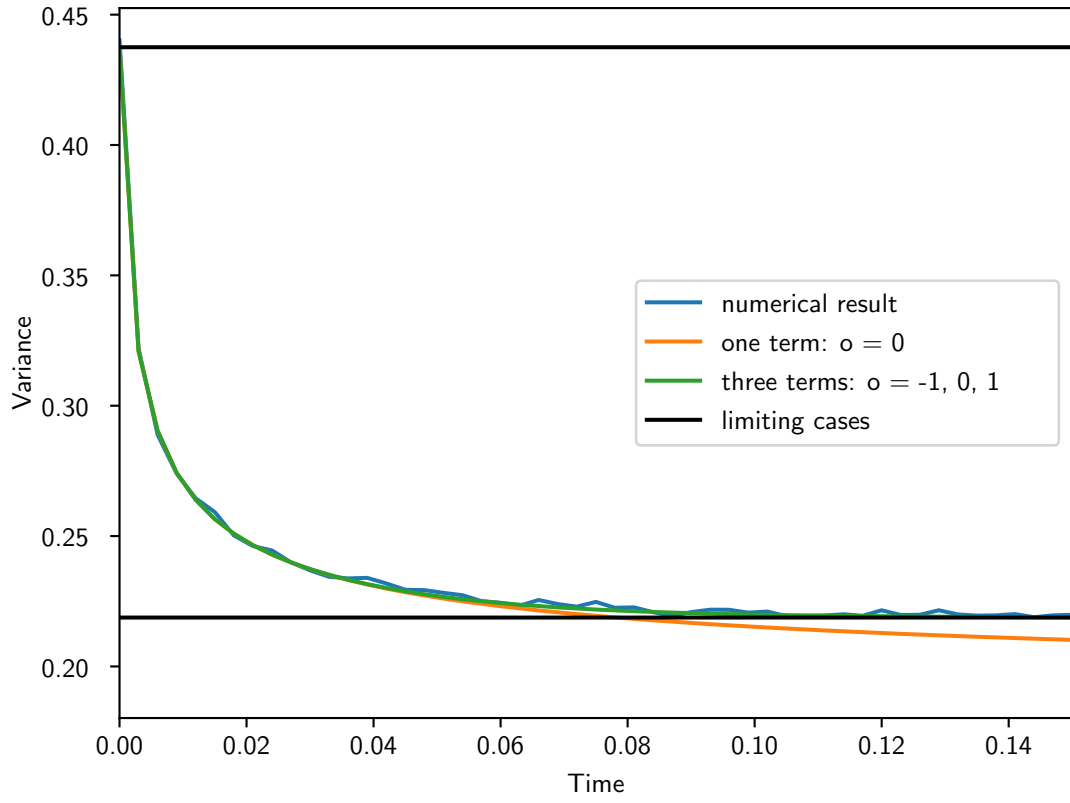


Figure A.1.: We compare our analytical result to the simulation, $L = 1$ and $l = 1/8$. For short times using only one term is sufficient, for longer times three terms are necessary.

A. Decorrelation Time

In figure [A.1](#) we can see both the numerical data and the analytical result. The agreement is very good, already with only one term, or for longer times three. For very long times however any finite number of terms will tend to 0.

B. Probability Density with the Deterministic Roulette

B.1. Definitions and Setup

In most of our calculations we neglect correlations between the markers. Here we start a calculation for the deterministic roulette scheme with hysteresis for the very simple Wiener process. As we will see even this becomes unfeasible.

Markers that survive a transition to the high weight regime will be called blue. Markers that are deleted will be called green. For the probability density itself we will only consider the green markers created by a single blue marker, and not the blue marker itself.

The roulette position is at $x = 0$, the splitting position at $x = a$. We also assume that there are no boundaries. This simplifies the calculations significantly, but makes it impossible to investigate large time scales.

B.2. Probabilities and Probability Distributions for Single Particles

The Green's function for the diffusion equation, and also for the probability density of the Wiener process,

$$\partial_t f(x, t) = D \partial_x^2 f(x, t)$$

is

$$f(x, t) = \frac{1}{\sqrt{4\pi Dt}} e^{-\frac{(x-x_0)^2}{4Dt}},$$

where the particle was started at $t = 0$ and $x = x_0$.

We require $P(x = 0) = 0$ as almost all particles at this position will cross the roulette position in an infinitesimal amount of time. Any nonzero probability would therefore very quickly become zero.

We want to start the marker at $x = a$ and fulfill this requirement. We achieve this by making $P(x)$ antisymmetric:

$$P(x, t) = \frac{1}{\sqrt{4\pi Dt}} \left[e^{-\frac{(x-a)^2}{4Dt}} - e^{-\frac{(x+a)^2}{4Dt}} \right]$$

This is valid for $x > 0$. For negative x $P = 0$.

B. Probability Density with the Deterministic Roulette

The probability Q that the particle survived until a time t is

$$\begin{aligned}
 Q(t) &= \int_0^\infty P(x) dx \\
 &= \int_0^\infty \frac{1}{\sqrt{4\pi Dt}} \left[e^{-\frac{(x-a)^2}{4Dt}} - e^{-\frac{(x+a)^2}{4Dt}} \right] dx \\
 &= \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{a}{\sqrt{4\pi Dt}} \right) \right] - \frac{1}{2} \left[1 - \operatorname{erf} \left(\frac{a}{\sqrt{4\pi Dt}} \right) \right] \\
 &= \operatorname{erf} \left(\frac{a}{\sqrt{4\pi Dt}} \right).
 \end{aligned}$$

This result is plausible: initially $Q = 1$, but for very long times the particle will almost certainly cross the roulette position. There it will be deleted or changed to a different marker type. Therefore $Q \rightarrow 0$.

The rate of deletion of markers created at $t = 0$ and $x = a$ is

$$d(t) = -\partial_t Q(t) = \frac{a}{\sqrt{4\pi D}} \frac{e^{-\frac{a^2}{4Dt}}}{t^{3/2}}. \quad (\text{B.1})$$

B.3. Combined Probability Distribution

Suppose that we start a blue marker at time r_0 at the position $x = 0$. A green marker will be created when it reaches $x = a$. The probability distribution for this is given by equation B.1. The probability distribution for the green marker is therefore

$$P_1(x, t) = \int_{r_0}^t ds_0 d(s_0 - r_0) P(x, t - s_0) \quad (\text{B.2})$$

$$= \frac{a}{4\pi D} \int_{r_0}^t ds_0 \frac{\exp\left(-\frac{a^2}{4D(s_0 - r_0)}\right)}{(s_0 - r_0)^{3/2}} \frac{1}{\sqrt{t - s_0}} \left[\exp\left(-\frac{(x - a)^2}{4D(t - s_0)}\right) - \exp\left(-\frac{(x + a)^2}{4D(t - s_0)}\right) \right]. \quad (\text{B.3})$$

Unfortunately this integral is hard, if not impossible, to solve. A numerical solution can be seen in figure B.1.

This is for a single crossing, but we have to account for several.

We define $d(t < 0) = 0$ and $P(x, t < 0) = 0$. Then the combined probability density of all created particles for a given set of splitting times is

$$P_{tot}(\{x_i\}, t; \{s_i\}) = \sum_i P(x_i, t - s_i).$$

Using equation B.1 again we can calculate the expected probability density of green markers

$$\left\langle P_{tot}(\{x_i\}, t; \{s_i\}) \right\rangle_{sr} = \int_{r_0}^t ds_0 d(s_0 - r_0) \int_{s_0}^t dr_1 d(r_1 - s_0) \int_{r_1}^t ds_1 d(s_1 - r_1) \dots \sum_{i=0}^{\infty} P(x_i, t - s_i).$$

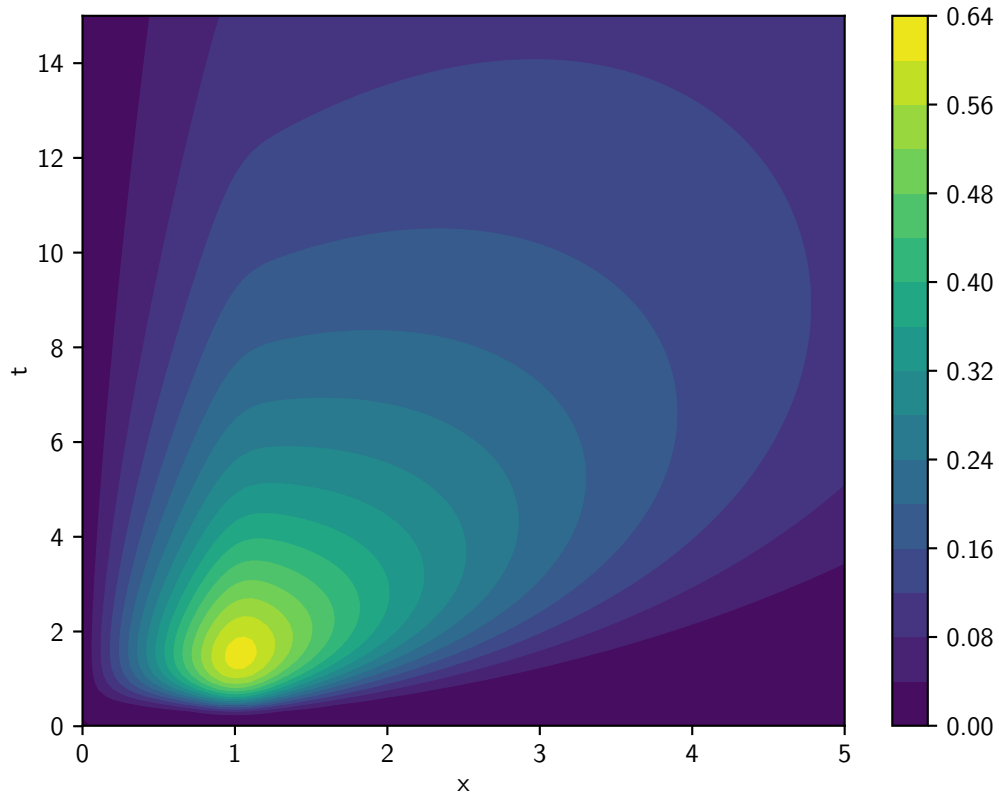


Figure B.1.: A numerical solution to equation B.2 with $a = 4D = 1$ and $r_0 = 0$. We see that some time elapses until creation of the green marker becomes probable. The probability density diffuses in x -space, but is required to be 0 at $x = 0$. It changes rapidly at $x = a = 1$ as there is a source at this position.

B. Probability Density with the Deterministic Roulette

Here we also consider the time it takes for the blue marker to return to $x = 0$. It is possible to pull the summands to the corresponding integrals, but this still seems very hard to solve. Additionally one has to replace the exponentials in the single particle probability distribution by an infinite sum thereof due to the reflecting boundary conditions. As an analytical solution to this problem seems improbable, we instead rely on numerical experiments to characterize the methods.

C. Influence of Correlated Deletion on Variance

When using the correlated roulette in one dimension we observe that in the high weight region we observe lower variance close to the boundary, where the markers are more strongly correlated, that further away where the markers decorrelated already. We can understand this effect by considering a simplified situation: We draw many uniformly distributed random numbers from the interval $[0, 1]$ and count how many are in a specific bin. By dividing this number by the total number of numbers and bin length we get an estimate for the density, which should be 1. Now we delete half of the numbers at random, equivalent to only drawing half as many numbers, and calculate the density again. Finally, instead of deleting half of the numbers at random we delete every second number after sorting them. Then we measure the density again.

The variance of the second density measurement is double the variance of the first measurement, as one would expect because the error of Monte Carlo methods is $\mathcal{O}(N^{-1/2})$. The variance of the third measurement however is almost identical to the variance of the first measurement. After a short calculation, also covering slightly more general situations, we will understand how this is possible.

We draw $N \cdot m$ real random numbers between 0 and 1, with N and m integers, and count the numbers inside a bin of length l . We get a mean and a variance:

$$\begin{aligned}\langle n \rangle_1 &= N m l \\ \sigma_{\text{count},1}^2 &= N m \left(\frac{1}{l} - 1 \right).\end{aligned}$$

In the second case we draw only N numbers, the mean and variance are:

$$\begin{aligned}\langle n \rangle_2 &= N l \\ \sigma_{\text{count},2}^2 &= N \left(\frac{1}{l} - 1 \right).\end{aligned}$$

The third case is more interesting. We start with what we have from the first case, but delete $m-1$ numbers, let one survive, and repeat the process. Before doing this we sort the numbers. If we choose $m = 2$ we have exactly the situation as before. Compared to the first measurement we will have less numbers in our bin, by a factor of $\frac{1}{m}$, but not exactly: Because the random numbers are not necessarily aligned with the bin boundaries we will in general have too many or too few deleted markers. The additional error is however restricted: We cannot be off by more than m counts. If m is small this is only a small contribution to the variance, although we deleted $(m-1)/m$ of all numbers.

C. Influence of Correlated Deletion on Variance

We define the relative error to be

$$\sigma_{\text{rel},i} = \frac{\sigma_{\text{count},i}}{\langle n \rangle_i}.$$

A short numerical experiment verifies that we indeed hardly increase the error. We choose $l = 0.1$, $N = 1000$ and $m = 2$ and average over 1 000 000 realizations. When deleting every second sampling position instead of only using half as many the relative error decreases by a factor of 0.7074. If we would not delete any sampling positions we expect that the standard deviation is $1/\sqrt{2} = 0.7071$.

If we have a boundary with multiplicity m and move $N \cdot m$ markers from the low-weight side to the high-weight side we have a situation similar to deleting every second number in the interval $[0, 1]$.

D. Wiener Process with Multiple Boundaries

When we want to resolve a region of configuration space with higher accuracy there are many degrees of freedom when setting the boundaries. Even after deciding on the difference of weight between the two regions of interest we have to choose the positions and number of boundaries. For example a weight difference by a factor of $1/4$ could be realized by a single boundary or two with a factor of $1/2$ each.

We use the theory introduced in section 3.4 for the analysis.

D.1. Influence of Boundary Position

The region with larger markers will have higher variance than the region with smaller markers. But the variance in each region will not only depend on the weight ratio, but also on the size of the regions. If the low weight region is very small it will hardly have any impact on the accuracy of the high weight region, neglecting eventually arising correlations. If we enlarge the low weight region the number of small markers will be increased and the number of large markers will be increased. Because we gain more small markers than we lose large markers the total number of markers will increase. This means that we increased computational costs without increasing accuracy in the high weight region. The accuracy per marker or computational cost decreased just by changing the boundary position.

D.2. Influence of Number of Boundaries

In section 4.2.3 we calculated that a hysteresis region interpolates linearly between regions. Now let us place many equidistant boundaries with equal weight fractions instead of just one. The variance, without the constant -1 , will be reduced by the same factor at each boundary. This means we do not connect the regions with a linear function but with a convex exponential function. This exponential lies beneath a linear function connecting the same regions. This is a first hint that splitting at many boundaries is better than mixing differently weighted markers in a large hysteresis region.

As we discussed in subsection D.1 we also have to check how large the variance in the low and high weight regions is. An analytical calculation for zero hysteresis and uncorrelated

D. Wiener Process with Multiple Boundaries

markers yields

$$\begin{aligned}\text{Var}(x) &= \frac{w(x)}{l} \int_0^L 1/w(x) \, dx - 1 \\ &= \frac{w(x)}{l} \sum_w L_w/w - 1.\end{aligned}$$

where $w(x)$ is the weight at position x , L is the length of the domain, and l the bin width. We assume that the bin lies inside a weight region.

Now we will place K equidistant boundaries with equal weight ratios W . The index $j = \text{Floor}(\frac{x}{L}K)$ denotes in which region we are. We denote the total weight ratio between the leftmost and rightmost region as m . We therefore have $W = m^{-1/K}$.

$$\begin{aligned}\text{var}(x) &= \frac{W^j}{l} \sum_{i=0}^K \frac{L}{K+1} / W^i - 1 \\ &= \frac{m^{-j/K}}{l} \sum_{i=0}^K \frac{L}{K+1} m^{i/K} - 1 \\ &= \frac{m^{-j/K}}{l} \frac{L}{K+1} \frac{1 - m^{1+1/K}}{1 - m^{1/K}} - 1.\end{aligned}$$

j is approximately proportional to K , therefore the first fraction will only separate $\text{var}(x)$ into discrete steps instead of a continuous function. The rest of $\text{var}(x)$ is independent of x and shrinks monotonously with K .

For equally spaced boundaries it is therefore favorable to use many thereof. The convex shape is preferable over a linear shape due to hysteresis, and the overall factor is decreased.

TRITA-SCI-GRU 2019:049

www.kth.se
www.tum.de
www.ipp.mpg.de