BACHELOR

# Estimation of entropy of random variables to analyze a stochastic model

van Schooten, Bart E.

*Award date:*
2019

# Estimation of Entropy of Random Variables to Analyze a Stochastic Model

Bachelor Final Project Applied Mathematics
Technische Universiteit Eindhoven

Bart van Schooten

Supervisors:
Mark A. Peletier
Markus Hütter
Alessandro Di Bucchianico

August 1, 2019

# Contents

# 1. Introduction

## 1.1. Problem statement

In this thesis we try to find an appropriate numerical approach to solve a specific set of coupled equations, consisting of a stochastic differential equation and an ordinary differential equation. The set of equations is as follows:

$$\mathrm{d}X_t = \frac{1}{\tau_\alpha}(X_t - R)\,\mathrm{d}t + \left(\frac{\partial}{\partial x}\frac{1}{\tau_\alpha}\right)\mathrm{d}t + \sqrt{\frac{2}{\tau_\alpha}}\,\mathrm{d}W_t \tag{1.1}$$

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = -\frac{1}{\tau_R}(\theta - 1) + \frac{k_B}{\theta\eta_{,\theta}}\int_{\mathbb{R}}\frac{p_t}{\tau_\alpha}\left(x - R + \frac{\mathrm{d}p_t}{\mathrm{d}x}\frac{1}{p_t}\right)^2\mathrm{d}x \tag{1.2}$$

Where $\tau_\alpha = \tau_\alpha(X_t, \theta)$, $\tau_R = \tau_R(\theta)$ and $p_t$ is the probability density function of $X_t$. The terms $R$ and $\frac{k_B}{\theta\eta_{,\theta}}$ are constants. Since $\tau_\alpha$ is a function of $\theta$, the stochastic differential equation for $X_t$ depends on $\theta$, while the ordinary differential equation for $\theta$ depends on the density $p_t$ of $X_t$

Because the two equations are dependent, it is impossible to find an exact solution for the probability density function $p_t$. Therefore, a suited numerical approach is needed. Another difficulty of the problem lies in the fact that the ordinary differential equation depends, in a non-linear way, on this density $p_t$.

## 1.2. Context

The equations (1.1) and (1.2) are based on a set of equations formulated by Semkiv et al. (2017). They build a model of elastomers filled with hard nanoparticles, describing the movement of particles in the elastomer. Part of the model describes the configurational temperature of the system, while another part describes the movement of a coupled pair of nano-particles. The main differences between the two sets of equations are that we reduce the dimensions from three to one, fix the position of $R$ and re-scale the model to make it dimensionless. The full equations, as Semkiv et al. (2017) formulated them are given by Equations (1.3) up to (1.11). Here we use their notation. For more information about this notation we refer to their paper.

$$\mathrm{d}Q_\gamma = -\frac{1}{\tau_\alpha}(Q_\gamma - R_\gamma)\,\mathrm{d}t + \frac{k_B T}{k}\left(\nabla^Q_\gamma\frac{1}{\tau_\alpha}\right)\mathrm{d}t + \sqrt{\frac{2k_BT}{k\tau_\alpha}}\mathrm{d}W_{t\gamma} \tag{1.3}$$

$$D_t\theta = -\frac{1}{\tau_R}(\theta - T) + \frac{1}{\theta\eta_{,\theta}}\int\frac{p}{k\tau_\alpha}\left(k(Q_\gamma - R_\gamma) + k_BT\frac{\nabla^Q_\gamma p}{p}\right)^2\mathrm{d}^6\xi \tag{1.4}$$

$$\mathrm{d}R_\gamma = \left[\nabla^r_\mu v_\gamma\right]^{\mathrm{sym}} R_\mu \mathrm{d}t \tag{1.5}$$

with:

$$\tau_\alpha = \tau_g \exp\left[-c_1\frac{c_2(T-\theta) + T(\theta - T_g(z,\Sigma_{loc}))}{T(c_2 + \theta - T_g(z,\Sigma_{loc}))}\right] \tag{1.6}$$

$$\tau_R = \tau^0_R \exp\left[-c_1\frac{c_2(T-\theta) + T(\theta - T_g(z,0))}{T(c_2 + \theta - T_g(z,0))}\right] \tag{1.7}$$

$$T_g(z,\Sigma_{loc}) = T_{g,b}\left(1 + \frac{\beta}{z}\right) - \frac{\Sigma_{loc}}{K} \tag{1.8}$$

$$\Sigma_{loc} = \sqrt{|\frac{1}{2}\left[(\mathrm{tr}\boldsymbol{\Sigma})^2 - \mathrm{tr}(\boldsymbol{\Sigma}\cdot\boldsymbol{\Sigma})\right]|} \tag{1.9}$$

$$\boldsymbol{\Sigma} = \left[\boldsymbol{R}\nabla^R\Phi_t\right] \tag{1.10}$$

$$z = \frac{|\boldsymbol{R}| - d}{2} \tag{1.11}$$

and $k_B$, $T$, $k$, $c_\theta = \theta\eta_{,\theta}$, $\tau_g$, $c_1$, $c_2$, $\tau_R^0$, $T_{g,b}$, $\beta$, $d$ and $K$ constants. Here $p$ is again the probability density function of $X_t$.

The main goal of their model is to describe the behaviour of elastomers filled with hard nanoparticles, while accounting for the physical aging of the glassy bridges of the filler particles. The model offers a step from empirical based models to more appropriate models, and in that way a step towards giving a conclusive description of, for example, the Mullins effect.

The derivation of Equation (1.1) and (1.2) from Equation (1.3) and (1.4) is explained in Appendix A.

## 1.3. Setup of the thesis

We first describe some theory about stochastic differential equations in Section 2 to get a better understanding of the problem. In Section 3, we describe a few statistical concepts. One of these is used in solving the problem, the others to analyze our results. Next we explain in Section 4 the notion of entropy and evaluate four methods to estimate the entropy of a random variable. In Section 5 we analyze a simplified version of the model. This allows us to obtain analytical results, which can be used to test our proposed approach. We then apply our approach to the full model in Section 6. Finally our conclusions are formulated in Section 7.

## 1.4. Main results

Multiple methods exist and can be used to estimate the entropy. With these estimates, a numerical approach can be formulated to solve Equation (1.1) and Equation (1.2). This approach is unbiased for $\theta$ and the variance in $\theta$ appears to be of the order of $\frac{1}{n}$ where $n$ is the amount of samples used. The stepsize $h$ has little influence on the variance of $\theta$, given that it is sufficiently small. This approach can be adapted such that it is suitable for the original problem as formulated by Semkiv et al. (2017).

# 2. Theory stochastic differential equations

In this section we will describe some theory behind stochastic differential equations. Before explaining what stochastic differential equations are, we first introduce the concepts of stochastic processes, Markov processes and diffusion processes, since they play a big role in the theory of stochastic differential equations. All definitions and theorems in this section are taken from Pavliotis (2011) unless mentioned differently.

## 2.1. Stochastic processes

A *stochastic process* is an ordered collection of random variables. To be precise, let $T$ be an ordered set. A stochastic process is a collection of random variables $X = \{X_t \mid t \in T\}$, where $X_t$ is a random variable taking values in a vector space $E$. Typical examples are $T = \mathbb{R}^+$ or $T = \mathbb{N}$ and $E = \mathbb{R}^d$. Instead of $X_t$, we also write $X(t)$.

### 2.1.1. Stationary processes

A stochastic process $X = \{X_t \mid t \in T\}$ is called a *stationary process* if the distributions of the random variables $X_t$ satisfy the following property:

$$(X(t_1), X(t_2), \ldots, X(t_k)) \stackrel{d}{=} (X(t_1 + s), X(t_2 + s), \ldots, X(t_k + s)) \tag{2.1}$$

for all $k \in \mathbb{N}^+$,$t_1, \ldots, t_k \in T$ and $s > 0$, where $\stackrel{d}{=}$ denotes being equal in distribution. Note that two random variables $X$ and $Y$ are equal in distribution if $\mathbb{E}[f(X)] = \mathbb{E}[f(Y)]$ for all $f$.

This implies that for a stationary stochastic process it holds that the distribution of $X(t)$ is equal to that of $X(0)$ for all $t \in T, t > 0$. Indeed, let $k = 1$ and $t_1 = 0$ and apply the definition above.

## 2.2. Markov processes

A *Markov process* is a stochastic process whose behaviour in the future only depends on the current state and not its past. It is therefore sometimes called a memoryless process. To be precise, a stochastic process is called a Markov process if

$$\mathbb{P}(X(t + h) \in A \mid X(s), s \leq t) = \mathbb{P}(X(t + h) \in A \mid X(t)) \tag{2.2}$$

holds for all $h > 0$ and $t \in T$ for which both probabilities are well-defined, where $A$ is an arbitrary set in $E$.

A Markov process $X$ is called a *time-homogeneous Markov process* if the evolution of the process is invariant under time shifts, i.e.:

$$\mathbb{P}(X(t + s) \in A \mid X(t) = x) = \mathbb{P}(X(s) \in A \mid X(0) = x) \tag{2.3}$$

holds for all $x \in E, t \in T, s > 0$ for which both probabilities are well-defined, where $A$ is an arbitrary set in $E$.

We will now analyze some properties of time-homogeneous Markov processes. We first consider the case where the state space is discrete and time is continuous. For example, $E = \mathbb{N}$ and $T = \mathbb{R}^+$. Let $X$ be a time-homogeneous Markov process with discrete state space and continuous time. Since the process is time-homogeneous, we can define:

$$P_{i,j}(t) := \mathbb{P}(X_{s+t} = j \mid X_s = i) = \mathbb{P}(X_t = j \mid X_0 = i) \tag{2.4}$$

with $i, j \in E$, for which the probability is well-defined. We can now introduce the following formula, known as the Chapman-Kolmogorov equation:

$$P_{i,j}(m+n) = \sum_{k \in \mathbb{E}} P_{i,k}(m) P_{k,j}(n) \tag{2.5}$$

It can easily be seen why this holds true. When one moves from state $i$ to state $j$ in $m+n$ timesteps, one has to be in a certain state $k$ after $m$ timesteps. For each state $k$, this happens with probability $P_{i,k}(m)$. To be in state $j$ after $m+n$ timesteps, one has to move from state $k$ to state $j$ in the remaining $n$ timesteps, which happens with probability $P_{k,j}(n)$. Since $X$ is a Markov process, these events are independent, so by summing over all possible states $k$ we get the total probability.

Let $\mu_t$ denote the distribution of $X_t$. The transition probabilities as defined in (2.4), combined with the initial distribution $\mu_0$ fully determine the distribution $\mu_t$ for all $t > 0$.

We first define the generator $G$ of a Markov process with discrete state space and continuous time as follows:

$$G := \lim_{t \downarrow 0} \frac{P(t) - I}{t} \tag{2.6}$$

where $P(t)$ is the matrix containing the probabilities $P_{i,j}(t)$ and $I$ the identity matrix. Note that $P(0) = I$, so $G$ is the derivative of $P(t)$ in 0. We can now derive the following differential equation:

$$\begin{aligned} \frac{\mathrm{d}P(t)}{\mathrm{d}t} &= \lim_{\Delta t \downarrow 0} \frac{P(t+\Delta t) - P(t)}{\Delta t} \\ &\overset{(2.5)}{=} \lim_{\Delta t \downarrow 0} \frac{P(t)P(\Delta t) - P(t)}{\Delta t} \\ &= \lim_{\Delta t \downarrow 0} \frac{P(t)(P(\Delta t) - I)}{\Delta t} \\ &= P(t) \lim_{\Delta t \downarrow 0} \frac{P(\Delta t) - I}{\Delta t} = P(t)G \end{aligned} \tag{2.7}$$

We can now solve this differential equation using an integrating factor:

$$\begin{aligned} \frac{\mathrm{d}P(t)}{\mathrm{d}t} &= P(t)G \\ \frac{\mathrm{d}P(t)}{\mathrm{d}t} - P(t)G &= 0 \\ e^{-Gt}\frac{\mathrm{d}P(t)}{\mathrm{d}t} - e^{-Gt}P(t)G &= 0 \\ \frac{\mathrm{d}}{\mathrm{d}t}(e^{-Gt}P(t)) &= 0 \\ e^{-Gt}P(t) &= c \\ P(t) &= Ce^{Gt} \end{aligned} \tag{2.8}$$

Since $G$ is a matrix, $e^{Gt}$ denotes a power series. Since $P(0) = I$, we have $C = 1$. Combining this with $\mu_t = \mu_0 P(t)$ we get a formula for the distribution $\mu_t$, $t \in T$:

$$\mu_t = \mu_0 \cdot e^{Gt} \tag{2.9}$$

## 2.3. Operator semigroups

When the state space $E$ of a time-homogeneous Markov process $X$ is continuous, for example $\mathbb{R}^d$, we can no longer talk about the probabilities $\mathbb{P}(X_{s+t} = j \mid X_s = i)$ since these probabilities can all be 0. Instead we consider a new probability transition function:

$$P(t, x, A) := \mathbb{P}(X_{s+t} \in A \mid X_s = x) = \mathbb{P}(X_t \in A \mid X_0 = x) \tag{2.10}$$

with $t \in T$, $x \in E$ and $A$ a set in $E$.

For the Chapman-Kolmogorov equation, we also formulate an analogue formula for continuous Markov processes, namely we have the following: Let $X$ be a time-homogeneous Markov process with continuous state space, let $P(t, x, A)$ be the corresponding transition function. The Chapman-Kolmogorov equation is now given by:

$$P(t + s, x, A) = \int_E P(t, x, \mathrm{d}y) P(s, y, A) \tag{2.11}$$

To analyze the behaviour of the transition function and the corresponding densities $\mu_t$, we first introduce some new definitions. Let $X$ be a time-homogeneous Markov process. Let $P(t, x, A)$ be the corresponding transition function. Define now the operator:

$$(P_t f)(x) := \mathbb{E}[f(X_t) \mid X_0 = x] = \int_{\mathbb{R}^d} f(y) P(t, x, \mathrm{d}y) \tag{2.12}$$

with $f \in C_b(\mathbb{R}^d)$.

Now $P_t$ forms a semigroup. A semigroup is a (ordered) set on which an (additive) operation is defined with an identity, but no inverses have to exist. This means that it should hold that $P_0 = I$ and $P_{t+s} = P_t \circ P_s$. Indeed:

$$\begin{aligned}
(P_0 f)(x) &= \mathbb{E}[f(X_0) \mid X_0 = x] = f(x) = (If)(x) \qquad (2.13)\\
(P_{t+s} f)(x) &= \int_{\mathbb{R}^d} f(y) P(t + s, x, \mathrm{d}y) \overset{(2.11)}{=} \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} f(y) P(s, z, \mathrm{d}y) P(t, x, \mathrm{d}z) \\
&= \int_{\mathbb{R}^d} \left( \int_{\mathbb{R}^d} f(y) P(s, z, \mathrm{d}y) \right) P(t, x, \mathrm{d}z) = \int_{\mathbb{R}^d} (P_s f)(z) P(t, x, \mathrm{d}z) \\
&= ((P_t \circ P_s) f)(x)
\end{aligned}$$

Since the transition probabilities as defined in (2.4) also form a semigroup - in fact, if we apply the definition above to a discrete case and choose $f(x) = \mathbb{1}_{\{x=y\}}$, we have $(P_t f)(x) = P_{x,y}(t)$ - it comes to mind to also look at a generator for this semigroup. We define the generator as follows:

$$\mathscr{L} f := \lim_{t \downarrow 0} \frac{P_t f - f}{t} \tag{2.14}$$

For the semigroup defined in (2.4), we found a solution as described in (2.8). We want to apply the same principle here, so we write:

$$P_t = e^{t\mathscr{L}} \tag{2.15}$$

There are other - more rigorous - arguments why this formula should hold, but we will not go in detail about it here, since this is not the main goal of this thesis. Let us now consider $\frac{\mathrm{d}}{\mathrm{d}t} P_t f$, by for example writing $g(x, t) := (P_t f)(x)$ and computing $\frac{\partial g}{\partial t}$:

$$\begin{aligned}
\frac{\partial g}{\partial t} &= \frac{\mathrm{d}}{\mathrm{d}t} P_t f = \frac{\mathrm{d}}{\mathrm{d}t} e^{t\mathscr{L}} \qquad (2.16)\\
&= \mathscr{L} e^{t\mathscr{L}} = \mathscr{L} P_t = \mathscr{L} u
\end{aligned}$$

since $g(x,0) = (P_0 f)(x) = f(x)$, we have the following initial value problem for expectations of functions of $X_t$, also known as the backward Kolmogorov equation:

$$\begin{aligned} \frac{\partial (P_t f)}{\partial t} &= \mathscr{L} P_t f \\ P_0 f(x) &= f(x) \end{aligned} \tag{2.17}$$

Instead of looking at functions of $X_t$, we also want to look at the density of $X_t$. To do this we consider the adjoint operator semigroup $P_t^*$. This operator acts on the dual of the space that the operator semigroup $P_t$ acts on, namely (probability) measures. The adjoint operator semigroup is defined as follows:

$$(P_t^* \mu)(A) := \int_{\mathbb{R}^d} \mathbb{P}(X_t \in A \mid X_0 = x) \, \mathrm{d}\mu(x) = \int_{\mathbb{R}^d} P(t,x,A) \, \mathrm{d}\mu(x) \tag{2.18}$$

The relation between the two operator semigroups is given by:

$$\int_{\mathbb{R}^d} (P_t f)(x) \, \mathrm{d}\mu(x) = \int_{\mathbb{R}^d} f(x) \, \mathrm{d}(P_t^* \mu)(x) \tag{2.19}$$

Since the two operators are adjoint, we can also formulate an adjoint generator $\mathscr{L}^*$ satisfying the following relation:

$$\int \mathscr{L} f(x) h(x) \, \mathrm{d}x = \int f(x) \mathscr{L}^* h(x) \, \mathrm{d}x \tag{2.20}$$

We can again formally formulate an expression similar to Formula (2.15), namely:

$$P_t^* = e^{t\mathscr{L}^*} \tag{2.21}$$

Let $X$ be a time-homogeneous Markov process with adjoint generator $\mathscr{L}^*$, corresponding adjoint semigroup $P_t^*$ and $X_0 \sim \mu_0$. We can now write:

$$\mu_t = P_t^* \mu \tag{2.22}$$

We now consider $\frac{\partial}{\partial t} \mu_t$:

$$\begin{aligned} \frac{\partial}{\partial t} \mu_t = \frac{\partial}{\partial t} P_t^* \mu_0 &= \frac{\partial}{\partial t} e^{t\mathscr{L}^*} \mu_0 \\ &= \mathscr{L}^* e^{t\mathscr{L}^*} \mu_0 = \mathscr{L}^* P_t^* \mu_0 = \mathscr{L}^* \mu_t \end{aligned} \tag{2.23}$$

Assuming that each $\mu_t$ has a density $p(x,t)$ and $\mu_0$ has density $p(x)$, we get the following initial value problem for the density of $X$:

$$\begin{aligned} \frac{\partial}{\partial t} p(x,t) &= \mathscr{L}^* p(x,t) \\ p(x,0) &= p(x) \end{aligned} \tag{2.24}$$

This expression is known as the forward Kolmogorov equation and as the Fokker-Planck equation. If we would know the adjoint operator $\mathscr{L}^*$ and the initial distribution $p(x)$ of a Markov process, we can use the Fokker-Planck equation to determine the distribution of $X_t$.

## 2.4. Diffusion processes

A *diffusion process* is a Markov process $X$ with state space $E = \mathbb{R}$ and $T = \mathbb{R}^+$, with corresponding transition function $P(t, A, s, x) = \mathbb{P}(X_t \in A \mid X_s = x)$ satisfying the following properties:

- Continuity, i.e. for every $x$ and every $\varepsilon > 0$ we have:

$$\int_{|x-y|>\varepsilon} P(t, \mathrm{d}y, s, x) = o(t-s) \tag{2.25}$$

- Drift term: There exists a function $b(x, s)$ such that for every $x$ and every $\varepsilon > 0$ we have:

$$\int_{|y-x|\leq\varepsilon} (y-x) P(t, \mathrm{d}y, s, x) = b(x, s)(t-s) + o(t-s) \tag{2.26}$$

- Diffusion term: There exists a function $\Sigma(x, s)$ such that for every $x$ and every $\varepsilon > 0$ we have:

$$\int_{|y-x|\leq\varepsilon} (y-x)^2 P(t, \mathrm{d}y, s, x) = \Sigma(x, s)(t-s) + o(t-s) \tag{2.27}$$

here $o(t-s)$ is the little-$o$ notation, i.e. $f(x) = o(g(x))$ if $\lim_{x\to 0} \frac{f(x)}{g(x)} = 0$. By assuming that the first and second moment exist, we can replace the integrals in (2.26) and (2.27) by integrals over $\mathbb{R}$, and we can write $b(x, s)$ and $\Sigma(x, s)$ as follows:

$$b(x, s) = \lim_{t\to s} \mathbb{E}\left[\frac{X_t - X_s}{t-s} \mid X_s = x\right] \tag{2.28}$$

$$\Sigma(x, s) = \lim_{t\to s} \mathbb{E}\left[\frac{(X_t - X_s)^2}{t-s} \mid X_s = x\right] \tag{2.29}$$

### 2.4.1. Fokker-Planck equation

The equations (2.26) and (2.27) for $b(x, s)$ and $\Sigma(x, s)$ respectively can be used to find an explicit expression for the generator $\mathscr{L}$ as defined in (2.14). It can be shown that for time-homogeneous processes, so $b(x, s) = b(x)$ and $\Sigma(x, s) = \Sigma(x)$, the following relation holds:

$$\mathscr{L} f(x, t) = b(x)\frac{\partial}{\partial x} f(x, t) + \frac{1}{2}\Sigma(x)\frac{\partial^2}{\partial x^2} f(x, t) \tag{2.30}$$

Similarly, an expression can be found for the adjoint generator $\mathscr{L}^*$. The adjoint generator is now given by:

$$\mathscr{L}^* f(x, t) = -\frac{\partial}{\partial x} b(x) f(x, t) + \frac{1}{2}\frac{\partial^2}{\partial x^2}\Sigma(x) f(x, t) \tag{2.31}$$

The Fokker-Planck equation as formulated in (2.24) now becomes:

$$\frac{\partial}{\partial t} p(x, t) = -\frac{\partial}{\partial x} b(x) p(x, t) + \frac{1}{2}\frac{\partial^2}{\partial x^2}\Sigma(x) p(x, t) \tag{2.32}$$

## 2.5. Stochastic differential equations

With the framework in place, we can now define a stochastic differential equation. Let $X$ be a diffusion process with $T = \mathbb{R}_+$ and $E = \mathbb{R}$. We now write a stochastic differential equation as:

$$\mathrm{d}X_t = b(X_t, t)\mathrm{d}t + \sigma(X_t, t)\mathrm{d}W_t \tag{2.33}$$

where $W_t$ denotes a *Brownian motion* (also known as a *Wiener process*). We will explain more about the Brownian motion in Section 2.5.1. We can also write the equation as a stochastic integral equation, with $x = X_0$ being the initial distribution:

$$X_t = x + \int_0^t b(X_t, t)\mathrm{d}t + \int_0^t \sigma(X_t, t)\mathrm{d}W_t \tag{2.34}$$

This requires us to define the stochastic integral:

$$I(t) = \int_0^t f(s)\mathrm{d}W(s) \tag{2.35}$$

We define the *Itô stochastic integral* as follows, let $t_k = k\Delta t, k \in \{1, \ldots, K\}$ such that $t_K = t$, we now define:

$$I(t) := \lim_{K\to\infty} \sum_{i=0}^{K-1} f(t_k)(W(t_{k+1}) - W(t_k)) \tag{2.36}$$

Note that evaluating $f(t)$ on a different place in the interval $[t_k, t_{k+1}]$ can lead to different results. For example, evaluating $f(t)$ in the midpoint of the interval will result in the *Stratonovich stochastic integral*. We will however not discuss this any further, and from now on use the Itô stochastic integral.

### 2.5.1. Brownian motion

In the definitions above we wrote $W_t$ for a Brownian motion. A Brownian motion $W_t$ is a diffusion process with $b(x, s) = 0$ and $\Sigma(x, s) = 1$, with $W_0 = 0$. However, it can also be characterized by the following properties:

1. $W_0 = 0$ with probability 1.
2. $W_t$ has independent increments. meaning that $W_{t+u} - W_u$, $u > 0$ is independent of $W_s$, $s < t$.
3. the increments are normally distributed, with $W_{t+u} - W_u \sim N(0, t)$ for all $u > 0$.
4. $W_t$ has a continuous path.

Using property 3, we can immediately find the distribution of $W_t$. Choosing $u = 0$, we get $W_t \sim N(0, t)$.

### 2.5.2. Itô's lemma and Itô's isometry

An important theorem when calculating stochastic integrals is Itô's Lemma (Itô (1944)). This theorem can be seen as the stochastic equivalent of the chain rule in calculus. Let $X$ be a stochastic process with state space $E = \mathbb{R}$ satisfying

$$\mathrm{d}X_t = b(X_t)\mathrm{d}t + \sigma(X_t)\mathrm{d}W_t \tag{2.37}$$

and let $f(x,t)$ be a twice differentiable function. Itô's lemma now tells us that $f(X_t,t)$ satisfies the following stochastic differential equation:

$$\mathrm{d}f(X_t,t) = \left(\frac{\partial}{\partial t}f(X_t,t) + b(X_t)\frac{\partial}{\partial x}f(X_t,t) + \frac{1}{2}\sigma(X_t)^2\frac{\partial^2}{\partial x^2}f(X_t,t)\right)\mathrm{d}t + \sigma(X_t)\frac{\partial}{\partial x}f(X_t,t)\mathrm{d}W_t \tag{2.38}$$

Another important formula is Itô's isometry, which states the following:

$$\mathbb{E}\left[\left(\int_0^t f(s)\,\mathrm{d}W_s\right)^2\right] = \mathbb{E}\left[\int_0^t f(s)^2\,\mathrm{d}s\right] \tag{2.39}$$

### 2.5.3. Connection with generators

We can use Itô's lemma to make a connection between the stochastic differential equation describing a stochastic process and its corresponding generator. Taking $f(X_t,t) = f(X_t)$ and rewriting Itô's lemma into an integral form gives us:

$$f(X_t) - f(X_0) = \int_0^t \left(b(X_s)\frac{\mathrm{d}}{\mathrm{d}x}f(x) + \frac{1}{2}\sigma(X_s)^2\frac{\mathrm{d}^2}{\mathrm{d}x^2}f(x)\right)\mathrm{d}s + \int_0^t \sigma(X_s)\frac{\mathrm{d}}{\mathrm{d}x}f(X_s)\,\mathrm{d}W_s$$

Taking the expectation on both sides:

$$\mathbb{E}\left[f(X_t) - f(X_0)\right] = \mathbb{E}\left[\int_0^t \left(b(X_s)\frac{\mathrm{d}}{\mathrm{d}x}f(x) + \frac{1}{2}\sigma(X_s)^2\frac{\mathrm{d}^2}{\mathrm{d}x^2}f(x)\right)\mathrm{d}s\right]$$

Dividing by $t$ and taking the limit $t \downarrow 0$:

$$\lim_{t\downarrow 0}\frac{\mathbb{E}\left[f(X_t) - f(X_0)\right]}{t} = \lim_{t\downarrow 0}\frac{\mathbb{E}\left[\int_0^t \left(b(X_s)\frac{\mathrm{d}}{\mathrm{d}x}f(x) + \frac{1}{2}\sigma(X_s)^2\frac{\mathrm{d}^2}{\mathrm{d}x^2}f(x)\right)\mathrm{d}s\right]}{t}$$

Using $X_0 = x$ and the definition of the generator:

$$\mathscr{L}f = b(x)\frac{\mathrm{d}}{\mathrm{d}x}f(x) + \frac{1}{2}\sigma(x)^2\frac{\mathrm{d}^2}{\mathrm{d}x^2}f(x)$$

So we have:

$$\mathscr{L} = b(x)\frac{\mathrm{d}}{\mathrm{d}x} + \frac{1}{2}\sigma(x)^2\frac{\mathrm{d}^2}{\mathrm{d}x^2} \tag{2.40}$$

### 2.5.4. Euler-Maruyama approximation

When simulating an ordinary differential equation, an easy method to use is the Euler (forward) method, as formulated by Euler (1824). Consider the following initial value problem:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}x &= f(x,t)\\ x(t_0) &= x_0\end{aligned} \tag{2.41}$$

The Euler (forward) method estimates $x(t)$ by taking small steps of size $h$. Let $t_n = t_0 + h\cdot n$. The value of $x(t_n)$ is estimated as:

$$x(t_n) = x(t_{n-1}) + hf(x(t_{n-1}), t_{n-1}) \tag{2.42}$$

We would like to adopt a similar approach for stochastic differential equations. Consider the following problem:

$$\begin{aligned} \mathrm{d}X_t &= b(X_t, t)\mathrm{d}t + \sigma(X_t, t)\mathrm{d}W_t \\ X_0 &= x \end{aligned} \tag{2.43}$$

Let $t_n = t_{n-1} + h$. The value of $X(t_n)$ is now estimated as:

$$X(t_n) = X(t_{n-1}) + b(X(t_{n-1}), t_{n-1})h + \sigma(X(t_{n-1}), t_{n-1})\sqrt{h}W_n \tag{2.44}$$

with

$$W_n \sim N(0, 1) \text{ i.i.d.} \tag{2.45}$$

This method is known as the Euler-Maruyama approximation and makes use of property 3 of a Brownian motion, as described in Section 2.5.1.

## 2.6. Ornstein-Uhlenbeck process

A Ornstein-Uhlenbeck process is a diffusion process with $b(x, s) = -a \cdot (x - \mu)$ and $\Sigma(x, t) = 2D$, with $a, D > 0$. Therefore we can write a Ornstein-Uhlenbeck process $X_t$ as a stochastic differential equation:

$$\mathrm{d}X_t = -a(X_t - \mu)\,\mathrm{d}t + \sqrt{2D}\,\mathrm{d}W_t \tag{2.46}$$

We can also formulate an expression for the generator and adjoint generator of the process:

$$\begin{aligned} \mathscr{L}f &= -a(x-\mu)\frac{\mathrm{d}}{\mathrm{d}x}f + D\frac{\mathrm{d}^2}{\mathrm{d}x^2}f \\ \mathscr{L}^*h &= \frac{\mathrm{d}}{\mathrm{d}x}a(x-\mu)h + D\frac{\mathrm{d}^2}{\mathrm{d}x^2}h \end{aligned} \tag{2.47}$$

We can easily see how the adjoint generator follows from the generator of the process. Namely, let $f$ and $h$ be functions with sufficiently fast decaying boundary terms. We can now write:

$$\int_{\mathbb{R}} \mathscr{L}f g\,\mathrm{d}x = \int_{\mathbb{R}} -a(x-\mu)\frac{\mathrm{d}}{\mathrm{d}x}fh + D\frac{\mathrm{d}^2}{\mathrm{d}x^2}fh\,\mathrm{d}x$$

Using partial integration:

$$= \int_{\mathbb{R}} f\frac{\mathrm{d}}{\mathrm{d}x}a(x-\mu)h - D\frac{\mathrm{d}}{\mathrm{d}x}f\frac{\mathrm{d}}{\mathrm{d}x}g\,\mathrm{d}x$$

Using partial integration for the second term:

$$= \int_{\mathbb{R}} f\frac{\mathrm{d}}{\mathrm{d}x}a(x-\mu)h + fD\frac{\mathrm{d}^2}{\mathrm{d}x^2}g\,\mathrm{d}x = \int_{\mathbb{R}} f\mathscr{L}^*g\,\mathrm{d}x$$

We can use the adjoint generator to find the stationary distribution of an Ornstein-Uhlenbeck process. That is, the distribution of $X_0$ such that the stochastic process becomes a stationary process as defined in Section 2.1.1. If a process is stationary, we have $\frac{\partial}{\partial t}p(x,t) = 0$, so we solve:

$$\mathscr{L}^*p = \frac{\partial}{\partial t}p(x,t) = 0$$

Filling in $\mathscr{L}^*$ and writing $p(x,t) = p(x)$:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}x} a(x-\mu)p(x) + D\frac{\mathrm{d}^2}{\mathrm{d}x^2}p(x) &= 0 \\ a(x-\mu)p(x) + D\frac{\mathrm{d}}{\mathrm{d}x}p(x) &= c \end{aligned}$$

Since $\lim_{x\to\infty} p(x) = \lim_{x\to\infty} \frac{\mathrm{d}}{\mathrm{d}x}p(x) = 0$, we can conclude $c = 0$, so:

$$\begin{aligned} a(x-\mu)p(x) + D\frac{\mathrm{d}}{\mathrm{d}x}p(x) &= 0 \\ \frac{\mathrm{d}}{\mathrm{d}x}p(x) + \frac{a}{D}(x-\mu)p(x) &= 0 \end{aligned}$$

Using partial integration:

$$\begin{aligned} e^{\frac{a}{2D}(x-\mu)^2}\frac{\mathrm{d}}{\mathrm{d}x}p(x) + e^{\frac{a}{2D}(x-\mu)^2}\frac{a}{D}(x-\mu)p(x) &= 0 \\ \frac{\mathrm{d}}{\mathrm{d}x}\left(e^{\frac{a}{2D}(x-\mu)^2}p(x)\right) &= 0 \\ e^{\frac{a}{2D}(x-\mu)^2}p(x) &= C \\ p(x) &= Ce^{-\frac{a}{2D}(x-\mu)^2} \end{aligned}$$

Normalizing such that p(x) is a density gives:

$$p(x) = \sqrt{\frac{a}{2\pi D}}e^{-\frac{a}{2D}(x-\mu)^2} \tag{2.48}$$

or equivalently:

$$X_t \sim \sqrt{\frac{D}{a}}N(\mu, 1) \tag{2.49}$$

# 3. Theory statistics

In this section we will look at some statistical concepts, which we will apply to our problem in later sections. First we describe kernel density estimation. Next we explain hypothesis testing and the $t$-test. Finally we describe two methods of normality testing. Namely the so called Quantile-Quantile plots and the Shapiro-Wilk test.

## 3.1. Kernel density estimation

A *kernel density estimator* is an estimator of a density $f$ of a random variable $X$ using a *kernel function*. Let $X$ be a random variable in $\mathbb{R}$, a kernel function is a function $K$ from $\mathbb{R}$ to $\mathbb{R}$ with the following properties:

- $K(x) \geq 0$
- $\int_{\mathbb{R}} k(x)\,\mathrm{d}x = 1$
- $K(-x) = K(x)$

Let $X_1, X_2 \ldots, X_n$ be independent identically distributed random variables with $X_i \overset{d}{=} X$ for $i \in \{1, \ldots, n\}$ and $K$ a kernel function. We now define a kernel density estimator $f_n$ as:

$$f_n(x) = \frac{1}{n \cdot h} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) \tag{3.1}$$

where $h$ is the so called bandwidth, which has to be chosen. The value of $h$ has great influence on the resulting kernel density estimate, and thus needs to be chosen carefully. The optimal choice of $h$ depends on the density $f$, which is often unknown. Therefore, an estimate needs to be made for $h$. An estimator is the so-called *Silverman's rule of thumb*. The Silverman's rule of thumb, as formulated by Silverman (1986) estimates $h$ as follows:

$$\hat{h} = 0.9 \min\{\hat{\sigma}, \frac{R}{1.34}\} \cdot n^{-\frac{1}{5}} \tag{3.2}$$

with $R$ the inter-quantile range and $\hat{\sigma}^2$ the sample variance:

$$\hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})^2$$

## 3.2. Hypothesis testing

*Hypothesis testing* is a common method in statistics to deduce information of an underlying distribution using data generated from that distribution. The general procedure is as follows:

1. Formulate statistical assumption, for example the type of distribution the data has.
2. Formulate the null hypothesis $H_0$ and the alternative hypothesis $H_1$.
3. Find an appropriate test statistic $T$ and its distribution.
4. Calculate the value of $T$ and compare it to a significance level $\alpha$.

We will illustrate the steps above with an example, namely what is known as the Student's $t$-test. The underlying assumption for this test is as follows. Assume that $X_1, X_2, \ldots, X_n$ are independent and identically distributed with $X_i \sim N(\mu, \sigma^2)$ for some unknown values $\mu \in \mathbb{R}$ and $\sigma^2 > 0$. The null hypothesis $H_0$ is now given by $\mu = \mu_0$ for some $\mu_0 \in \mathbb{R}$. The alternative hypothesis $H_1$ is therefore $\mu \neq \mu_0$. An appropriate test statistic $T$ is now given by:

$$T = \frac{\bar{X} - \mu_0}{\frac{\hat{\sigma}}{\sqrt{n}}} \tag{3.3}$$

Where $\hat{\sigma}$ is the sample standard deviation as defined above. Student (1908) showed that the distribution of $T$ is the so called $t$-distribution with $n-1$ degrees of freedom. We can now calculate a value $t_{1-\alpha/2,n-1}$ such that:

$$\mathbb{P}\left(-t_{1-\alpha/2,n-1} \leq T \leq t_{1-\alpha/2,n-1}\right) = 1 - \alpha \tag{3.4}$$

If now $T < -t_{1-\alpha/2,n-1}$ or $T > t_{1-\alpha/2,n-1}$ we reject the null hypothesis with confidence level $\alpha$. Otherwise, we say we fail to reject the null hypothesis.

Another common approach in hypothesis testing is to not fix a certain significance level $\alpha$ before performing the test, but instead reporting the so called $p$-value of the test. The $p$-value is the significance level for which we borderline our decision whether we accept or reject our null-hypothesis. That is, for all significance levels $\alpha < p$ we reject the null-hypothesis and for all significance levels $\alpha > p$ we fail to reject the null-hypothesis. In this thesis we will report the $p$-value when we perform a test, and use a significance level of $\alpha = 0.05$ to conclude whether or not we reject the null-hypothesis.

## 3.3. Normality testing

Since the normal distribution appears in a lot of applications, and the Student's t-test assumes normality, it is useful to test whether or not some independent data $X_1, X_2, \ldots, X_n$ has an underlying distribution that is normal. We therefore want to test the null hypothesis $H_0$ that $X_i \sim N(\mu, \sigma^2)$ for some $\mu \in \mathbb{R}$ and $\sigma^2 > 0$ against the alternative hypothesis that $X_i \sim Y$ where $Y$ is a random variable that is not normally distributed. Different methods have been developed to test this hypothesis. We will discuss 2 methods, the *Quantile-Quantile plots* and the *Shapiro-Wilk test*.

### 3.3.1. Quantile-Quantile plots

Testing for normality using Quantile-Quantile plots (Q-Q plots) is a qualitative method for testing normality. Let $X_{(1)}, X_{(2)}, \ldots, X_{(n)}$ be the ordered statistics. When the data is normally distributed, we expect that $X_{(i)} \approx Z_{i/n}$, with $Z_{i/n}$ the so called $i/n$-quantiles, i.e. $\mathbb{P}(X \leq Z_{i/n}) = \frac{i}{n}$. When plotting $X_{(i)}$ against $Z_{i/n}$, we thus expect to see a straight line. We can visually inspect whether or not this is the case.

### 3.3.2. Shapiro-Wilk test

A quantitative way to test for normality is the so called Shapiro-Wilk test. Let $X_1, X_2, \ldots, X_n$ be independent data and let $X_{(1)}, X_{(2)}, \ldots, X_{(n)}$ be the ordered statistics. The test statistic $W$ as formulated by Shapiro and Wilk (1965) is now given by:

$$W = \frac{\left(\sum_{i=1}^{n} a_i X_{(i)}\right)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{3.5}$$

with $\bar{x}$ being the sample average and:

$$\begin{aligned}(a_1, a_2, \ldots, a_n) &= \frac{m^T V^{-1}}{C} \\ C &= \|V^{-1}m\| = \left(m^T V^{-1} V^{-1} m\right)^{\frac{1}{2}} \\ m &= (\mathbb{E}[Z_{(1)}], \mathbb{E}[Z_{(2)}], \ldots, \mathbb{E}[Z_{(n)}]) \\ V_{i,j} &= \mathrm{Cov}(Z_i, Z_j)\end{aligned}$$

Finding the exact distribution of $W$ is difficult and also depends on $n$. However, we can still find the critical values $w_{1-\alpha/2,n}$. Using these, we reject the null hypothesis $H_0$ when $W < w_{1-\alpha/2,n}$ and fail to reject $H_0$ when $W \geq w_{1-\alpha/2,n}$.

# 4. Theory entropy

In this section we will define the *differential entropy* of a random variable. We will also look at the relation with the *Fisher information* in the scenario of stochastic processes. Finally we investigate different methods to estimate the differential entropy.

## 4.1. Entropy

Let $X$ be a random variable in $\mathbb{R}^d$ with density $p$. The differential entropy of $X$, as formulated by Thomas and Cover (1991), is defined as follows:

$$H(p) := -\int_{\mathbb{R}^d} p(x) \ln p(x) \, \mathrm{d}x \tag{4.1}$$

When the density $p$ of $X$ is known, we can calculate the entropy exactly. For example, let $X$ be a 1-dimensional random variable which is normally distributed, i.e. it has density:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{4.2}$$

with $\mu \in \mathbb{R}$ and $\sigma > 0$. To determine the entropy, we calculate the following integral:

$$\begin{aligned}
H(f) &= -\int p(x) \ln p(x) \, \mathrm{d}x \\
&= -\int \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \left( \ln\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) - \frac{(x-\mu)^2}{2\sigma^2} \right) \mathrm{d}x \\
&= -\ln\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) + \mathbb{E}\left[\frac{(X-\mu)^2}{2\sigma^2}\right] \\
&= \ln\sqrt{2\pi\sigma^2} + \frac{1}{2\sigma^2}\left(\mathbb{E}[X^2] - 2\mu\mathbb{E}[X] + \mu^2\right) \\
&= \ln\sqrt{2\pi\sigma^2} + \frac{1}{2\sigma^2}\left(\mu^2 + \sigma^2 - 2\mu^2 + \mu^2\right) \\
&= \ln\sqrt{2\pi\sigma^2} + \frac{1}{2} \\
&= \ln\sqrt{2\pi\sigma^2} + \ln\sqrt{e} = \ln(\sqrt{2\pi e}\sigma)
\end{aligned} \tag{4.3}$$

### 4.1.1. Fisher information

When the random variable is a part of a stochastic process $X$, we can relate the entropy $H(p)$ to the so called *Fisher Information*. The Fisher information is defined as follows: Let $X$ be a random variable in $\mathbb{R}$ with probability density $p$, the Fisher information $I(p)$ is now given by:

$$I(p) := \int_{\mathbb{R}} \frac{1}{p(x)} \left( \frac{\mathrm{d}p(x)}{\mathrm{d}x} \right)^2 \mathrm{d}x \tag{4.4}$$

The Fisher information can be related to the entropy if we have a stochastic process $X$ satisfying the following Fokker-Planck equation:

$$\frac{\partial}{\partial t} p = \frac{\partial^2}{\partial x^2} p \tag{4.5}$$

We can now relate the Fisher information and the entropy as follows, writing $p' = \frac{\partial p}{\partial x}$:

$$\begin{aligned} I(p) &= \int_{\mathbb{R}} \frac{1}{p} \left(p'\right)^2 \right) \, \mathrm{d}x \\ &= \int_{\mathbb{R}} p' \frac{p'}{p} \, \mathrm{d}x \\ &= \int_{\mathbb{R}} p' \left(\ln p + 1\right)' \, \mathrm{d}x \\ &= -\int_{\mathbb{R}} p'' \left(\ln p + 1\right) \, \mathrm{d}x \\ &= -\int_{\mathbb{R}} \frac{\partial p}{\partial t} \left(\ln p + 1\right) \, \mathrm{d}x \\ &= -\int_{\mathbb{R}} \frac{\partial}{\partial t} \left(p \ln p\right) \, \mathrm{d}x \\ &= \frac{\partial}{\partial t} H(p) \end{aligned} \tag{4.6}$$

4.1.2. Augmented entropy and Fisher information

The relation above can be extended to more general stochastic processes if we change the definitions of the Fisher information and the entropy slightly, by including an extra term. Consider a stochastic process of the following form:

$$\mathrm{d}X_t = \left(\frac{\mathrm{d}}{\mathrm{d}x} V(x)\right) \mathrm{d}t + \sqrt{2}\mathrm{d}W \tag{4.7}$$

For some differentiable function $V(x)$. The Fokker-Planck equation is now given by:

$$\frac{\partial}{\partial t} p = \frac{\partial}{\partial x} \left(\frac{\mathrm{d}}{\mathrm{d}x} V(x)\right) p + \frac{\partial^2}{\partial x^2} p \tag{4.8}$$

We now define the augmented Fisher information as follows:

$$\widetilde{I}(p) := \int_{\mathbb{R}} p \left(\left(\frac{\mathrm{d}}{\mathrm{d}x} V(x)\right) + \frac{\mathrm{d}p}{\mathrm{d}x} \frac{1}{p}\right)^2 \, \mathrm{d}x \tag{4.9}$$

The augmented entropy is now defined as:

$$\begin{aligned} \widetilde{H}(p) &:= -\int_{\mathbb{R}} p \left(V(x) + \ln p\right) \, \mathrm{d}x \\ &= -\mathbb{E}[V(x)] + H(p) \end{aligned} \tag{4.10}$$

We can now derive the following relation:

$$\begin{aligned} \widetilde{I}(p) &= \int_{\mathbb{R}} p \left(\left(\frac{\mathrm{d}}{\mathrm{d}x} V(x)\right) + \frac{\mathrm{d}p}{\mathrm{d}x} \frac{1}{p}\right)^2 \, \mathrm{d}x \\ &= \int_{\mathbb{R}} p \left(\left(\frac{\mathrm{d}}{\mathrm{d}x} V(x)\right) + \frac{\mathrm{d}p}{\mathrm{d}x} \frac{1}{p}\right) \left(\left(\frac{\mathrm{d}}{\mathrm{d}x} V(x)\right) + \frac{\mathrm{d}p}{\mathrm{d}x} \frac{1}{p}\right) \, \mathrm{d}x \\ &= \int_{\mathbb{R}} \left(\left(\frac{\mathrm{d}}{\mathrm{d}x} V(x)\right) p + \frac{\mathrm{d}p}{\mathrm{d}x}\right) \left(\left(\frac{\mathrm{d}}{\mathrm{d}x} V(x)\right) + \frac{\mathrm{d}p}{\mathrm{d}x} \frac{1}{p}\right) \, \mathrm{d}x \\ &= \int_{\mathbb{R}} \left(\left(\frac{\mathrm{d}}{\mathrm{d}x} V(x)\right) p + \frac{\mathrm{d}p}{\mathrm{d}x}\right) \frac{\mathrm{d}}{\mathrm{d}x} \left(V(x) + \ln(p) + 1\right) \, \mathrm{d}x \end{aligned} \tag{4.11}$$

Using partial integration, and assuming sufficiently fast decaying boundary terms:

$$= -\int_{\mathbb{R}} \frac{\mathrm{d}}{\mathrm{d}x} \left( \left( \frac{\mathrm{d}}{\mathrm{d}x} V(x) \right) p + \frac{\mathrm{d}p}{\mathrm{d}x} \right) (V(x) + \ln(p) + 1) \, \mathrm{d}x$$

Using the Fokker-Planck equation:

$$\begin{aligned} &= -\int_{\mathbb{R}} \frac{\partial p}{\partial t} (V(x) + \ln(p) + 1) \, \mathrm{d}x \\ &= -\int_{\mathbb{R}} \frac{\partial}{\partial t} (V(x)p + p\ln(p)) \, \mathrm{d}x \\ &= \frac{\partial}{\partial t} \widetilde{H}(p) \end{aligned}$$

## 4.2. Estimation performance

When the density of $p$ is not known exact, for example when the random variable belongs to a stochastic process $X$ of which the density can not be calculated exactly, an estimation of the entropy needs to be made. When comparing different estimators, criteria need to be formulated to compare the estimators on. Common criteria are to compare the bias or the mean square error. Let $e$ be the real value we try to estimate, and let $\hat{e_n}$ be an estimator for $e$ using $n$ samples, the bias $b_e(\hat{e_n})$ and the mean square error $MSE_e(\hat{e_n})$ are defined as following:

$$b_e(\hat{e_n}) = \mathbb{E}[\hat{e_n}] - e \tag{4.12}$$
$$MSE_e(\hat{e_n}) = \mathbb{E}[(\hat{e_n} - e)^2] \tag{4.13}$$

Instead of evaluating the behaviour of the estimators for finite $n$, we can also analyze the limiting behaviour of it. We use the following types of consistencies to describe this limiting behaviour:

Weak consistency:

$$\lim_{n\to\infty} \hat{e_n} = e \text{ in probability} \tag{4.14}$$

Mean square consistency:

$$\lim_{n\to\infty} \mathbb{E}\left[(\hat{e_n} - e)^2\right] = 0 \tag{4.15}$$

Strong consistency:

$$\lim_{n\to\infty} \hat{e_n} = e \text{ almost surely} \tag{4.16}$$

Using these definitions, strong consistency implies mean square consistency, which implies weak consistency.

## 4.3. Estimation methods

We consider four different estimators of the entropy $H(p)$. Those being a *resubstitution method*, a *cross-validation method*, a *sample-spacing method* and a *nearest-neighbour method*.

#### 4.3.1. Resubstitution method

Let $X_1, X_2, \ldots, X_n$ be independent and identically distributed random variables with common density $f$. An estimator for the entropy $H(p)$ using a resubstitution method, as formulated by Ahmad and Lin (1976), is now given by:

$$H_n^{\text{Resub}} = -\frac{1}{n}\sum_{i=1}^{n} \ln p_n(X_i) \tag{4.17}$$

where $p_n(x)$ is a kernel density estimator as defined in (3.1) for some kernel $K$ and bandwidth $h$. A heuristic argument on why this method works is that when $n$ gets large, $p_n$ converges to $p$. If $p_n$ would be equal to the density $p$, (4.17) converges in probability to $H(p)$ by the weak law of large numbers. Ahmad and Lin (1976) proved that this method is mean square consistent. Joe (1989) studied this estimator for higher dimensional random variables. He obtained asymptotic terms for the bias and variance of the estimator and showed that these terms can be reduced by using specific kernel functions. He also suggests that the required samplesize $n$ for good estimates rapidly increases with the number of dimensions.

#### 4.3.2. Cross-validation method

Let $X_1, X_2, \ldots, X_n$ be as defined above. An estimator for the entropy $H(p)$ using a cross-validation method, as formulated by Ivanov and Rozhkova (1981) is now given by:

$$H_n^{\text{Crossval}} = -\frac{1}{n}\sum_{i=1}^{n} \ln p_{n,i}(X_i) \tag{4.18}$$

with

$$p_{n,i}(x) = \frac{1}{(n-1)\cdot h}\sum_{\substack{j=1 \\ j\neq i}}^{n} K\left(\frac{x - X_i}{h}\right) \tag{4.19}$$

For some kernel $K$ and bandwidth $h$. A heuristic argument for this estimator is the same as for the resubstitution method, as $p_{n,i}(x)$ also converges to $p(x)$ when $n$ gets large for all $i$. Ivanov and Rozhkova (1981) proved the strong consistency of this method. They also provide upper-bounds for $\mathbb{E}\left[|H_n - H(p)|^r\right]$, $r > 1$ for finite $n$.

#### 4.3.3. Sample-spacing method

Let $X_1, X_2, \ldots, X_n$ be independent and identically distributed random variables taking values in $\mathbb{R}$. An estimator for the entropy $H(p)$ using the sample-spacing method, as formulated by Beirlant et al. (2001) is now given by:

$$H_{n,m}^{\text{SS}} = \frac{1}{n}\sum_{i=1}^{n-m} \ln\left(\frac{n}{m}(X_{(i+m)} - X_{(i)})\right) - \psi(m) + \ln(m) \tag{4.20}$$

here $X_{(i)}$ are the order statistics, $\psi(x) = -(\ln\Gamma(X))'$ the digamma function and $m$ a parameter that has to be chosen. The parameter $m$ smoothens the data in a way, and in that way it is comparable to the parameter $h$ in the kernel density estimator. This method relies on the fact that there exists an inverse relation between the logarithm of $p(x)$ and the logarithm of the distance between samples of the random variable. Under certain tail-conditions on the probability density function $p(x)$, Hall (1984) proved the weak consistency of this method.

4.3.4. Nearest-neighbour method

Let $X_1, X_2, \ldots, X_n$ be independent and identically distributed random variables taking values in $\mathbb{R}^d$. Kozachenko and Leonenko (1987) suggest estimating the entropy $H$ using a nearest-neighbour method in the following way:

$$H_n^{\mathrm{NN}} = \frac{d}{n}\sum_{i=1}^{n}(\ln \rho_i) + \ln C_d + C_e + \ln(n-1) \tag{4.21}$$

with:

$$\rho_i = \min_{j\neq i}\{\|X_i - X_j\|\} \tag{4.22}$$

$$C_d = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)} \tag{4.23}$$

where $d$ is the dimension of the random variable, $C_d$ the volume of a $d$-dimensional unit-sphere (Formula (4.23) only applies when using the euclidean norm), $C_e$ the Euler constant ($= -\int e^{-t}\ln t\,\mathrm{d}t$) and $\Gamma(x)$ the gamma function. Kozachenko and Leonenko (1987) proved the mean square consistency of this method for general $d$. Furthermore, Delattre and Fournier (2017) studied the variance and bias. They provided an upper-bound for the bias and proved the asymptotic normality of the estimator.

Since it is not directly clear how and why the method works, we calculate the expected value of $H_n^{\mathrm{NN}}$ to proof that it is asymptotically unbiased. The proof, as given by Kozachenko and Leonenko (1987), relies on the following theorem:

Let $f : \mathbb{R}^d \to \mathbb{R}$ be a continuous function, $x \in \mathbb{R}^d$ and $(r_k)$ be a sequence in $\mathbb{R}$ with $r_k \to 0$. Then we have:

$$\lim_{k\to\infty}\frac{1}{|B(x,r_k)|}\int_{B(x,r_k)} f(y)\,\mathrm{d}y = f(x) \tag{4.24}$$

where $B(x,r)$ denotes the sphere with radius $r$ around $x$ and $|B(x,r)|$ the volume of the sphere.

Define $\xi_i = \rho_i^d \cdot \gamma \cdot (n-1)\cdot C_d$ with $\rho_i$ and $C_d$ as defined above and $\gamma = e^{C_e}$. We can now rewrite the entropy estimate $H_n^{\mathrm{NN}}$ as:

$$H_n^{\mathrm{NN}} = \frac{1}{n}\sum_{i=1}^{n}\ln \xi_i \tag{4.25}$$

Define $r_n(u) = \left(\frac{u}{C_d\cdot\gamma\cdot(n-1)}\right)^{1/d}$. Now $r_n(u)$ is a sequence in $\mathbb{R}$ with $r_n(u)\to 0$. Next, we fix $i$ and determine the cumulative density distribution of $\xi_i$ given $X_i$.

$$\begin{aligned} F_{x,n}(u) &= \mathrm{P}(\xi_i < u | X_i = x) \\ &= \mathrm{P}(\rho_i < \left(\frac{u}{C_d\cdot\gamma\cdot(n-1)}\right)^{1/d} | X_i = x) \\ &= 1 - \mathrm{P}(\bigcap_{\substack{j=1\\ j\neq i}}^{n} X_j \notin B(x, r_n(u))) \\ &= 1 - \left(1 - \int_{B(x,r_n(u))} p(y)dy\right)^{n-1} \end{aligned} \tag{4.26}$$

Using (4.24), we can let $n$ go to infinity, and get:

$$F_x(u) := \lim_{n\to\infty} F_{x,n}(u) = 1 - \exp\left\{\frac{-p(x)u}{\gamma}\right\} \tag{4.27}$$

We calculate the expected value of the logarithm of $\xi_i$ under this distribution:

$$\mathbb{E}[\ln \xi_i] = \int_0^\infty \ln(u)\frac{p(x)}{\gamma}\exp\left\{\frac{-p(x)u}{\gamma}\right\}\,\mathrm{d}u \tag{4.28}$$

Using the transformation $t = \frac{p(x)u}{\gamma}$:

$$\begin{aligned}
&= \int_0^\infty \ln\left(\frac{t\gamma}{p(x)}\right)\exp\{-t\}\,\mathrm{d}t \\
&= \ln\gamma - \ln p(x) + C_e \\
&= -\ln p(x)
\end{aligned}$$

Overall, this means that for $n$ going to infinity we can write the expected value of $H_n^{\mathrm{NN}}$ as follows:

$$\begin{aligned}
\mathbb{E}[H^{\mathrm{NN}}] := \lim_{n\to\infty}\mathbb{E}[H_n^{\mathrm{NN}}] &= \lim_{n\to\infty}\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \ln\xi_i\right] \\
&= \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^n \mathbb{E}[\ln\xi_i] \\
&= H(p)
\end{aligned} \tag{4.29}$$

thus the estimator is asymptotically unbiased.

## 4.4. Simulation of entropy estimators

To make a comparison between the different estimation methods, we will simulate random data $X_1, \ldots, X_n$ from a distribution of which the exact entropy $H(p)$ is known, and estimate the entropy using the different estimation methods.

### 4.4.1. Setup

In the simulations we will use $n = 1000$, so we will use $X_1, \ldots, X_{1000}$ to estimate the entropy. For each method, we will repeat the simulation 1000 times, to get a good idea of how the estimation method performs. We will generate our data from a normal distribution with mean 0 and variance 1, so $X \sim N(0,1)$. Using (4.3), we know $H(p) = \ln\sqrt{2\pi e} \approx 1.4189$. For each method, we will look at a kernel density estimation of $H_{1000}$. Since $H_{1000}$ is a function of random variables, it is also a random variable, and thus has a probability density function. We estimate this probability density function using the `density` function in `R`. We will also test whether or not the methods are biased, using a $t$-test as described in Section 3.2. Lastly, we report the mean square error of each method.

For the resubstitution method and the cross-validation method, we will use the Gaussian density as kernel function $K$. For the bandwidth $h$ we use Silverman's rule of thumb as described in (3.2). For the sample-spacing method we use $m = 1$. The `R`-code used for this simulation, as well as all the code used for other simulations in this thesis, can be found in Appendix B.

4.4.2. Results

The empirical density for the resubstitution method can be seen in Figure 1. The $p$-value of the $t$-test is smaller than $2.2 \cdot 10^{-16}$. This means that we should reject the null hypothesis stating that the estimated mean is equal to the exact entropy, indicating a bias in the method. The mean square error is $5.455 \cdot 10^{-4}$.

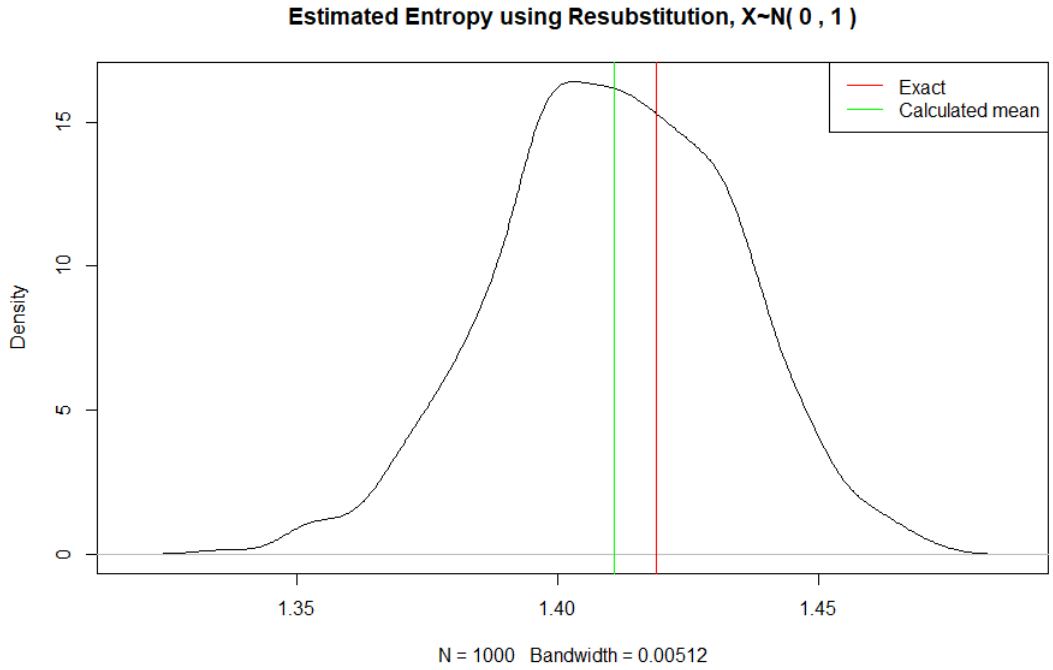


Figure 1: Density of the resubstitution estimator

The empirical density we found for the cross-validation method can be seen in Figure 2. The $t$-test gives a $p$-value of $2.404 \cdot 10^{-12}$. Therefore we should reject the null hypothesis stating that the estimated mean is equal to the exact entropy, indicating a bias in the method. The mean square error in our simulation is $5.477 \cdot 10^{-4}$.

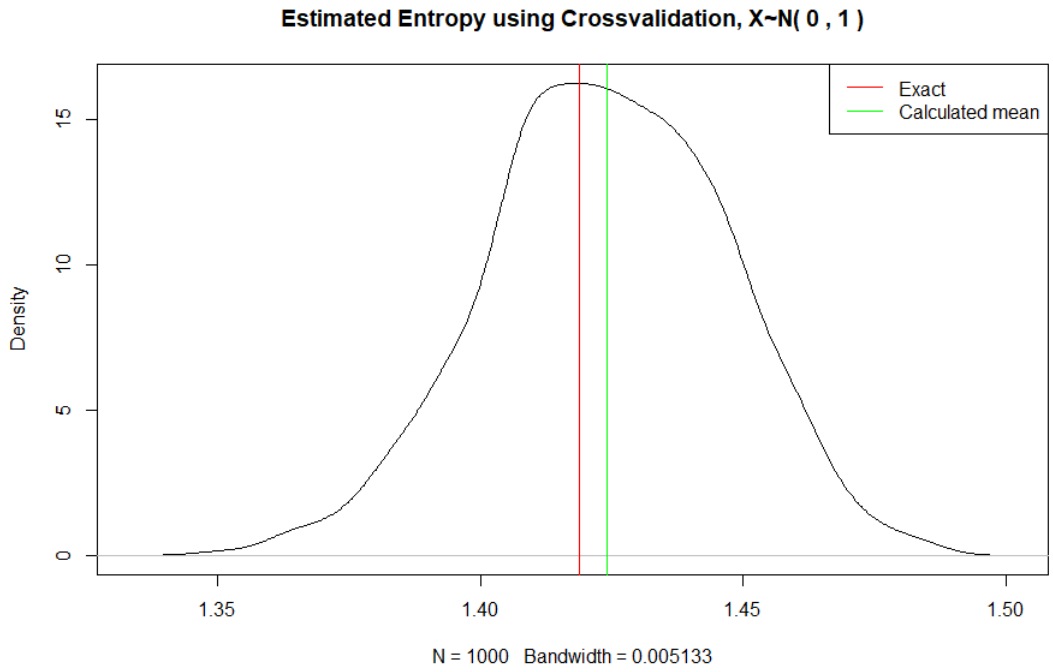


Figure 2: Density of the cross-validation estimator

The empirical density for the sample-space method is shown in Figure 3. The $p$-value of the $t$-test is $4.338 \cdot 10^{-7}$. This means that we should reject the null hypothesis stating that the estimated mean is equal to the exact entropy, indicating a bias in the method. The mean square error is $1.133 \cdot 10^{-3}$.

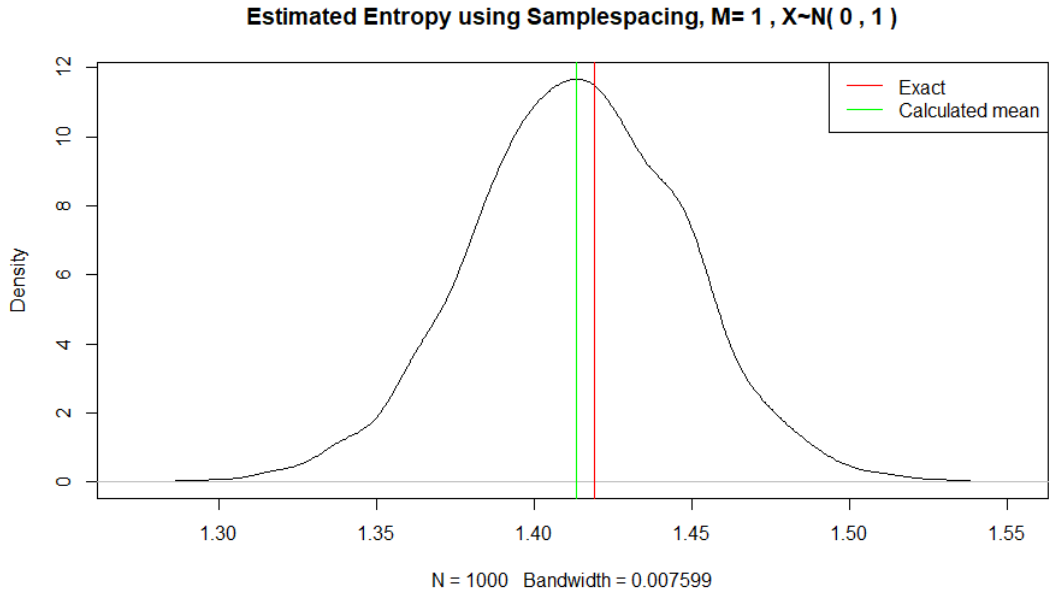


Figure 3: Density of the sample-space estimator

The empirical density for the nearest-neighbour method can be seen in Figure 4. Performing the $t$-test gives a $p$-value of 0.1755. This means that there is no clear evidence to reject the null-hypothesis, indicating that we can assume that the method is unbiased. The mean square error is $2.692 \cdot 10^{-3}$ according to our simulation.

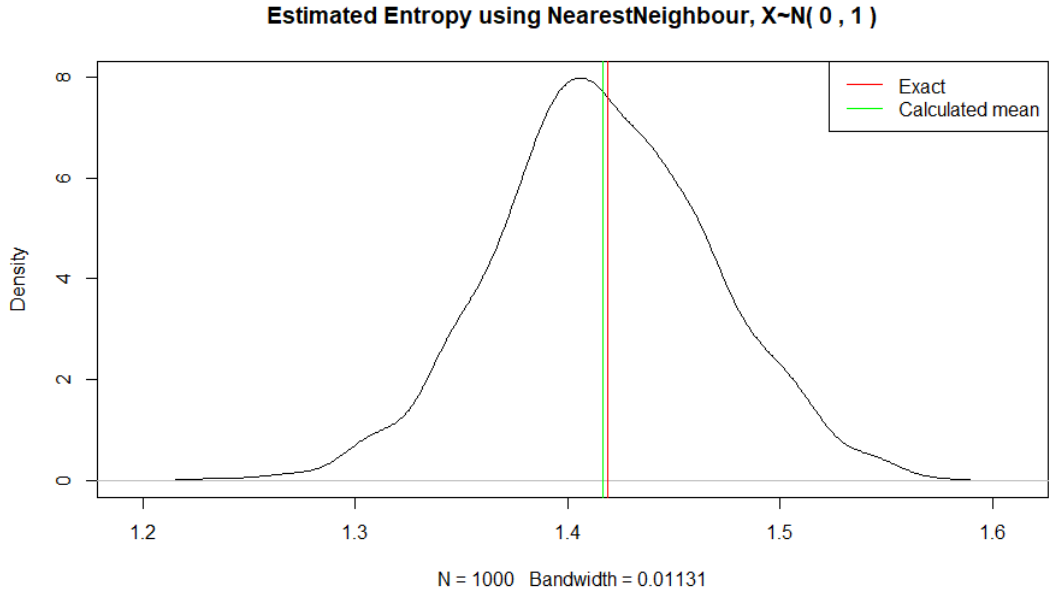


Figure 4: Density of the nearest-neighbour estimator

The results of our simulation are summarized in Table 1

Table 1: Results of the simulation

| Method | Biased | Mean square error |
|---|---|---|
| Resubstitution | Yes | $5.455 \cdot 10^{-4}$ |
| Cross-validation | Yes | $5.477 \cdot 10^{-4}$ |
| Sample-space | Yes | $1.133 \cdot 10^{-3}$ |
| Nearest-neighbour | No | $2.692 \cdot 10^{-3}$ |

## 4.5. Conclusion

According to our simulations all methods except the nearest neighbour method are statistically significantly biased for a samplesize $n = 1000$. The nearest neighbour method, however, has the largest mean square error of the four discussed methods for this samplesize. The resubstitution method and the cross-validation method have the lowest mean square errors.

The sample-spacing method in the form we discussed seems to be the worst option of the four. Firstly because it has a relative high mean square error while still being biased according to our simulation. Secondly because no mean square consistency has been proved for general distributions as far as we know. Lastly because the method only works for 1-dimensional random variables and can not be scaled to higher dimensions, as $\mathbb{R}^d$, $d > 1$ is not ordered, so no order statistics exists.

The mean square consistency has been proved for the three other methods. The cross-validation method is even proved to be strongly consistent. In higher dimensions all methods perform worse than in one dimension, but no clear indication can be made on which method performs better, as this may also depend on the distribution of $X$.

Based on the simulation, when one seeks for an unbiased method, the nearest-neighbour method should be chosen. When one seeks for a method with a low mean square error the resubstitution method or the cross-validation method should be chosen.

# 5. Simplified model

We will first analyze a simplified version of the model described in Section 1.1. In this section, we will assume that $\tau_\alpha$ and $\tau_R$ are both constant in $t$ and $x$. The reason for doing that is by taking $\tau_\alpha$ constant, the stochastic differential equation no longer depends on $\theta$, which allows us to find the exact distribution of $X_t$. We can use this solution to find an explicit expression for the ordinary differential equation, in terms of $\theta$ and $t$. We can then compare a possible numerical approach to existing methods for solving ordinary differential equations.

## 5.1. Solving the stochastic differential equation

When $\tau_\alpha$ is constant, we can re-scale the model with $\tau_\alpha$ instead of $\tau_g$ as we have done in Section 1.1. The model now becomes:

$$\mathrm{d}X_t = -(X_t - R)\mathrm{d}t + \sqrt{2}\mathrm{d}W \tag{5.1}$$

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = -c_\tau(\theta - 1) + c_k \int_{\mathbb{R}} p\left(x - R + \frac{\mathrm{d}p}{\mathrm{d}x}\frac{1}{p}\right)^2 \mathrm{d}x \tag{5.2}$$

By taking $\tau_\alpha$ constant, the stochastic differential equation becomes a Ornstein-Uhlenbeck process, which we can solve using an integrating factor. Let $f(X_t, t) = X_t e^t$ and applying Itô's lemma gives:

$$\begin{aligned}\mathrm{d}f\left(X_t, t\right) &= \left(X_t e^t + (R - X_t)e^t\right)\mathrm{d}t + \sqrt{2}e^t\,\mathrm{d}W \\ \mathrm{d}X_t e^t &= Re^t\mathrm{d}t + \sqrt{2}e^t\,\mathrm{d}W\end{aligned}$$

Rewriting this into an integral form gives:

$$X_t e^t = X_0 + \int_0^t Re^s\mathrm{d}s + \int_0^t \sqrt{2}e^s\,\mathrm{d}W_s$$

Computing the integral gives:

$$\begin{aligned}X_t e^t &= X_0 + R(e^t - 1) + \sqrt{2}\int_0^t e^s\,\mathrm{d}W_s \\ X_t &= X_0 e^{-t} + R(1 - e^{-t}) + \sqrt{2}\int_0^t e^{s-t}\,\mathrm{d}W_s\end{aligned} \tag{5.3}$$

We can use this formula to determine the mean $\mu(t)$ and variance $\sigma^2(t)$ of $X_t$. Assume that $X_0$ is distributed normally, so $X_0 \sim N(\mu_0, \sigma_0^2)$. We now compute:

$$\begin{aligned}\mathbb{E}\left[X_t\right] =& \mathbb{E}\left[X_0 e^{-t} + R(1 - e^{-t}) + \sqrt{2}\int_0^t e^{s-t}\,\mathrm{d}W_s\right] \\ =& \mathbb{E}\left[X_0\right]e^{-t} + R(1 - e^{-t}) + \sqrt{2}\cdot\mathbb{E}\left[\int_0^t e^{s-t}\,\mathrm{d}W_s\right] \\ =& \mu_0 e^{-t} + R(1 - e^{-t})\end{aligned} \tag{5.4}$$

Similarly, for the second moment we compute:

$$\begin{aligned}\mathbb{E}\left[X_t^2\right] =& \mathbb{E}\left[\left(X_0 e^{-t} + R(1 - e^{-t}) + \sqrt{2}\int_0^t e^{s-t}\mathrm{d}W_s\right)^2\right] \\ =& \mathbb{E}\Bigg[\left(X_0 e^{-t}\right)^2 + 2X_0 e^{-t}R(1 - e^{-t}) + 2\sqrt{2}X_0 e^{-t}\int_0^t e^{s-t}\,\mathrm{d}W + (R(1 - e^{-t}))^2 \\ &+ 2\sqrt{2}R(1 - e^t)\int_0^t e^{s-t}\,\mathrm{d}W_s + 2\left(\int_0^t e^{s-t}\,\mathrm{d}W_s\right)^2\Bigg]\end{aligned} \tag{5.5}$$

Computing the expectations and applying Itô's isometry gives:

$$\begin{aligned}
&= \left(\mu_0^2 + \sigma_0^2\right) e^{-2t} + 2\mu_0 e^{-t} R(1 - e^{-t}) + R^2(1 - e^{-t})^2 + 2\int_0^t e^{2s-2t}\mathrm{d}s \\
&= \left(\mu_0 e^{-t} + R(1 - e^{-t})\right)^2 + \sigma_0^2 + 2\left(\frac{1}{2}e^{2t-2t} - \frac{1}{2}e^{0-2t}\right) \\
&= \left(\mu_0 e^{-t} + R(1 - e^{-t})\right)^2 + \sigma_0^2 + (1 - e^{-2t})
\end{aligned}$$

So for the variance we get:

$$\begin{aligned}
\mathrm{Var}(X_t) =& \mathbb{E}\left[X_t^2\right] - \mathbb{E}\left[X_t\right]^2 \\
&= \left(\mu_0 e^{-t} + R(1 - e^{-t})\right)^2 + \sigma_0^2 + (1 - e^{-2t}) - \left(\mu_0 e^{-t} + R(1 - e^{-t})\right)^2 \\
&= \sigma_0^2 e^{-2t} + \left(1 - e^{-2t}\right)
\end{aligned} \tag{5.6}$$

We want to use these formulas to derive a closed form expression for $p_t$, the probability density function of $X_t$. If $X_t$ were to be normally distributed for all $t \geq 0$, we would have such closed form expression. To see whether $X_t$ stays normally distributed if $X_0$ is normally distributed, we solve the Fokker-Planck equation.

### 5.1.1. Solving the Fokker-Planck equation

The generator and adjoint generator of the stochastic process are given by:

$$\mathscr{L}f = -(x - R)\frac{\partial}{\partial x}f + \frac{\partial^2}{\partial x^2}f \tag{5.7}$$

$$\mathscr{L}^* h = \frac{\partial}{\partial x}(x - R)h + \frac{\partial^2}{\partial x^2}h \tag{5.8}$$

We want to show that $X_t$ being normal distributed with time dependent $\mu$ and $\sigma$ solves the Fokker-Planck equation, so:

$$\frac{\partial}{\partial t}p = \mathscr{L}^* p = \frac{\partial}{\partial x}(x - R)p + \frac{\partial^2}{\partial x^2}p \tag{5.9}$$

with

$$p(x,t) = \frac{1}{\sqrt{2\pi}\sigma(t)} e^{-\frac{1}{2}\frac{(x-\mu(t))^2}{\sigma^2(t)}} \tag{5.10}$$

for some functions $\mu(t)$ and $\sigma(t)$, where $p(x,t)$ is the probability density function of $X_t$. We compute the left- and right-hand side of Fokker-Planck equation in terms of $\mu$ and $\sigma$:

$$\begin{aligned}
\frac{\partial}{\partial t}p &= \frac{\partial}{\partial t}\frac{1}{\sqrt{2\pi}\sigma(t)} e^{-\frac{1}{2}\frac{(x-\mu(t))^2}{\sigma^2(t)}} \\
&= -\frac{\sigma'}{\sigma}p + p\left(\frac{(x-\mu)\mu'}{\sigma^2} + \frac{(x-\mu)^2\sigma'}{\sigma^3}\right)
\end{aligned} \tag{5.11}$$

$$\begin{aligned}
\frac{\partial}{\partial x}(x - R)p &= \frac{\partial}{\partial x}(x - R)\frac{1}{\sqrt{2\pi}\sigma(t)} e^{-\frac{1}{2}\frac{(x-\mu(t))^2}{\sigma^2(t)}} \\
&= p - (x - R)p\frac{x - \mu}{\sigma^2}
\end{aligned} \tag{5.12}$$

$$\begin{aligned}
\frac{\partial^2}{\partial x^2}p &= \frac{\partial^2}{\partial x^2}\frac{1}{\sqrt{2\pi}\sigma(t)} e^{-\frac{1}{2}\frac{(x-\mu(t))^2}{\sigma^2(t)}} \\
&= -\frac{1}{\sigma^2}p + \frac{(x-\mu)^2}{\sigma^4}p
\end{aligned} \tag{5.13}$$

Therefore, the Fokker-Planck equation gives us:

$$-\frac{\sigma'}{\sigma}p + p\left(\frac{(x-\mu)\mu'}{\sigma^2} + \frac{(x-\mu)^2\sigma'}{\sigma^3}\right) = p - (x-R)p\frac{x-\mu}{\sigma^2} - \frac{1}{\sigma^2}p + \frac{(x-\mu)^2}{\sigma^4}p \tag{5.14}$$

where $f'$ denotes $\frac{\partial f}{\partial t}$. Dividing by $p$ gives:

$$\frac{(x-\mu)^2\sigma'}{\sigma^3} + \frac{(x-\mu)\mu'}{\sigma^2} - \frac{\sigma'}{\sigma} = \frac{(x-\mu)^2}{\sigma^4} - (x-R)\frac{x-\mu}{\sigma^2} - \frac{1}{\sigma^2} + 1$$

Since this formula has to hold for all $x \in \mathbb{R}$, we get by expending the squares:

$$\frac{\sigma'}{\sigma^3} = \frac{1}{\sigma^4} - \frac{1}{\sigma^2} \tag{5.15}$$

$$-2\mu\frac{\sigma'}{\sigma^3} + \frac{\mu'}{\sigma^2} = -\frac{2\mu}{\sigma^4} + \frac{\mu}{\sigma^2} + \frac{R}{\sigma^2} \tag{5.16}$$

$$\frac{\mu^2\sigma'}{\sigma^3} - \frac{\mu\mu'}{\sigma^2} - \frac{\sigma'}{\sigma} = \frac{\mu^2}{\sigma^4} - \frac{R\mu}{\sigma^2} - \frac{1}{\sigma^2} + 1 \tag{5.17}$$

Solving Equation (5.15) for $\sigma'$ and substitution into Equation (5.16) gives:

$$\sigma' = \frac{1}{\sigma} - \sigma \tag{5.18}$$

$$\mu' = -\mu + R \tag{5.19}$$

Equation (5.17) automatically holds, because substituting (5.18) and (5.19) into (5.17) gives $0 = 0$. Equations (5.18) and (5.19) are indeed solved by formulas for the mean and variance we found earlier, namely:

$$\mu(t) = \mu_0 e^{-t} + R(1 - e^{-t}) \tag{5.20}$$

$$\sigma^2(t) = \sigma_0 e^{-2t} + (1 - e^{-2t}) \tag{5.21}$$

Notice that for $\sigma^2$ the following differential equation holds:

$$\begin{aligned} \sigma' &= \frac{1}{\sigma} - \sigma \\ 2\sigma\sigma' &= 2 - 2\sigma^2 \\ (\sigma^2)' &= -2\sigma^2 + 2 \end{aligned} \tag{5.22}$$

Thus $X_t \sim N(\mu(t), \sigma^2(t))$ with $\mu(t)$ and $\sigma^2(t)$ as described in (5.20) and (5.21), satisfies the Fokker-Planck equation, and is thus the solution to the stochastic differential equation.

## 5.2. Solving the ordinary differential equation

Since we now have a close form expression for the probability density function $p_t$, we can rewrite the ordinary differential equation. Recall that the differential equation was given by:

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = -c_\tau(\theta - 1) + c_k \int_{\mathbb{R}} p\left(x - R + \frac{\mathrm{d}p}{\mathrm{d}x}\frac{1}{p}\right)^2 \mathrm{d}x \tag{5.23}$$

We rewrite this in the following way:

$$\begin{aligned}\frac{\mathrm{d}\theta}{\mathrm{d}t} + c_\tau\theta &= c_\tau + c_k \int_{\mathbb{R}} p\left(x - R + \frac{\mathrm{d}p}{\mathrm{d}x}\frac{1}{p}\right)\left(x - R + \frac{\mathrm{d}p}{\mathrm{d}x}\frac{1}{p}\right)\mathrm{d}x \\ &= c_\tau + c_k \int_{\mathbb{R}} \left((x-R)p + \frac{\mathrm{d}p}{\mathrm{d}x}\right)\left(x - R + \frac{\mathrm{d}p}{\mathrm{d}x}\frac{1}{p}\right)\mathrm{d}x \\ &= c_\tau + c_k \int_{\mathbb{R}} \left((x-R)p + \frac{\mathrm{d}p}{\mathrm{d}x}\right)\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{2}(x-R)^2 + \ln(p) + 1\right)\mathrm{d}x \\ &= c_\tau - c_k \int_{\mathbb{R}} \frac{\mathrm{d}}{\mathrm{d}x}\left((x-R)p + \frac{\mathrm{d}p}{\mathrm{d}x}\right)\left(\frac{1}{2}(x-R)^2 + \ln(p) + 1\right)\mathrm{d}x\end{aligned} \tag{5.24}$$

Using the Fokker-Planck equation:

$$\begin{aligned}\frac{\mathrm{d}\theta}{\mathrm{d}t} + c_\tau\theta &= c_\tau - c_k \int_{\mathbb{R}} \frac{\mathrm{d}p}{\mathrm{d}t}\left(\frac{1}{2}(x-R)^2 + \ln(p) + 1\right)\mathrm{d}x \\ &= c_\tau - c_k \frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathbb{R}} \left(\frac{1}{2}(x-R)^2 p + p\ln(p)\right)\mathrm{d}x \\ &= c_\tau - c_k \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{2}\mathbb{E}\left[(X-R)^2\right] - H(p)\right)\end{aligned} \tag{5.25}$$

Using that $X_t \sim N(\mu(t), \sigma^2(t))$, we can write:

$$\frac{1}{2}\mathbb{E}\left[(X-R)^2\right] = \frac{1}{2}(\mu_0 - R)^2 e^{-2t} + \frac{1}{2}(\sigma_0^2 - 1)e^{-2t} + \frac{1}{2} \tag{5.26}$$

$$H(p) = \frac{1}{2}\ln\left(2\pi e((\sigma_0^2 - 1)e^{-2t} + 1)\right) \tag{5.27}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{1}{2}\mathbb{E}\left[(X-R)^2\right] = -(\mu_0 - R)^2 e^{-2t} - (\sigma_0^2 - 1)e^{-2t} \tag{5.28}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}H(p) = -\frac{(\sigma_0^2 - 1)e^{-2t}}{(\sigma_0^2 - 1)e^{-2t} + 1} \tag{5.29}$$

Therefore, we can write the differential equation as:

$$\begin{aligned}\frac{\mathrm{d}\theta}{\mathrm{d}t} + c_\tau\theta &= c_\tau - c_k\left(\frac{(\sigma_0^2 - 1)e^{-2t}}{(\sigma_0^2 - 1)e^{-2t} + 1} - (\mu_0 - R)^2 e^{-2t} - (\sigma_0^2 - 1)e^{-2t}\right) \\ &= c_\tau - c_k\left(\frac{\sigma^2(t) - 1}{\sigma^2(t)} - (\mu(t) - R)^2 - (\sigma^2(t) - 1)\right)\end{aligned} \tag{5.30}$$

## 5.3. Simulation

### 5.3.1. Setup

In this simulation, we will estimate $\theta(t)$ using two different methods. The two methods are described below. The `R`-code used for this simulation, as well as all the code used for other simulations in this thesis can be found in Appendix B.

#### 5.3.1.1. Method 1

For the first method we use the function `ode` from the `deSolve` package in `R` (Soetaert et al. (2010)), which numerically solves the differential equation (5.30). Since the derivation of Equation (5.30) makes use of the fact that $X_t \sim N(\mu(t), \sigma^2(t))$ when $X_0 \sim N(\mu_0, \sigma_0^2)$, where $\mu(t)$ and $\sigma(t)$ are given by formula (5.20) and formula (5.21) respectively, this method relies on this fact about the distribution of $X_t$.

5.3.1.2. Method 2

For the second method we use a Euler-Maruyama approximation, as described in Section 2.5.4, to simulate the stochastic differential equation (5.1) for multiple independent $X_1, \ldots, X_n$. We then use the nearest-neighbour method as described in Section 4.3.4 to estimate $H(p)$. We choose the nearest-neighbour method out of the four methods we discussed in Section 4.3 as it is the only method that is unbiased. We use the law of large numbers to estimate $\mathbb{E}\left[\frac{1}{2}(X-R)^2\right]$. With these estimates we can numerically solve Formula (5.25) using an Euler-forward method. We use the Euler-forward method instead of more advanced methods as we expect that the biggest errors are made in the estimations of $H(p)$ and $\mathbb{E}\left[\frac{1}{2}(X-R)^2\right]$, and that therefore improving the method used to solve Equation (5.25) will not improve the method by much.

Combining this all together, we get the following procedure for the second simulation. Let $X_{1,m}, \ldots, X_{n,m}, \theta_m$ denote the state-space at $t = t_m$ and let $t_{m+1} = t_m + h$, where $h$ is the stepsize. We compute $X_{1,m+1}, \ldots, X_{n,m+1}, \theta_{m+1}$ as follows:

$$X_{i,m+1} = X_{i,m} - h(X_{i,m} - R) + \sqrt{2h}W_{i,m} \tag{5.31}$$

$$\theta_{m+1} = \theta_m - c_\tau h(\theta_m - 1) - c_k(F_{m+1} - F_m) \tag{5.32}$$

with

$$W_{i,m} \sim N(0,1) \text{ i.i.d.} \tag{5.33}$$

$$F_m = \frac{1}{2}\frac{1}{n}\sum_{j=1}^{n}(X_{j,m} - R)^2 - H_{n,m}^{\mathrm{NN}} \tag{5.34}$$

with $H_{n,m}^{\mathrm{NN}}$ as defined in (4.21) using $X_{1,m}, \ldots, X_{n,m}$. For both methods we use $h = 0.001$. For the second method we use $n = 1000$. We will run two simulations with different starting values for $\mu_0$ and $\sigma_0^2$. In both simulations $X_0$ is normally distributed.

5.3.2. Simulation 1: stationary distribution

For the first simulation we use $\mu_0 = R$, $\sigma_0^2 = 1$ and $\theta_0 = 1$. Under these conditions, the system is stationary, as the formulas for $\mu(t)$ and $\sigma^2(t)$ reduce to $\mu(t) = R$ and $\sigma^2(t) = 1$. The differential equation (5.30) reduces to $\frac{\mathrm{d}\theta}{\mathrm{d}t} = 0$, meaning that $\theta$ is also constant. For the exact solution the choices of $c_\tau$ and $c_k$ do not matter, but in the second method they still play a role. In the simulation we use $c_\tau = c_k = 0.1$. These parameters do not have a big influence on the overall behaviour of the model and only amplify the variance of $\theta$. Note that the value of $R$ also does not matter, as the model is shift-invariant. In the simulation we use $R = 0$. We let $t$ run from 0 to 500. The results of the simulation can be seen in Figure 5.

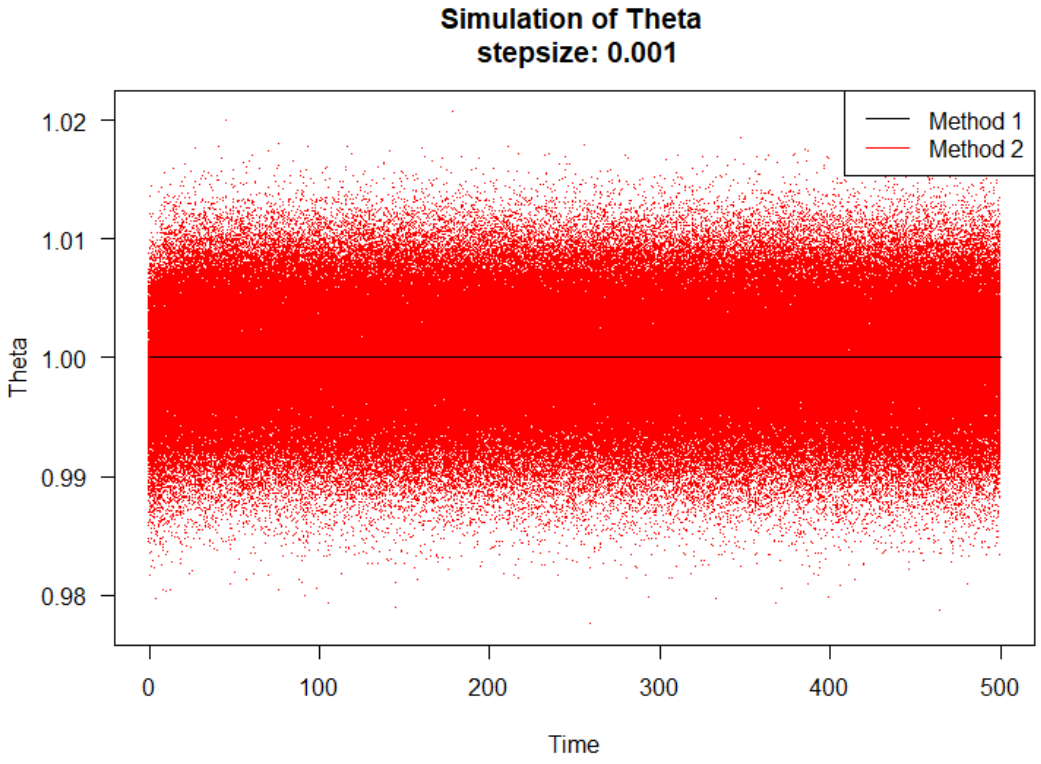


Figure 5: Simulation of $\theta$

To see whether or not the second method has a mean of 1, and thus is unbiased, we want to perform a $t$-test as described in Section 3.2. The assumptions of the $t$-test however requires that our data is independent and normally distributed. The values $\theta(s)$ calculated in the simulation however are clearly not independent, as $\theta(s+h)$ is calculated using $\theta(s)$. The correlations between $\theta(s)$ and $\theta(t)$ however decreases as $|s-t|$ gets larger. We will assume that $\theta(s)$ and $\theta(t)$ are (nearly) independent if $X_{1,s}, \ldots, X_{n,s}$ and $X_{1,t}, \ldots, X_{n,t}$ are (nearly) independent. Note that $X_{i,s}$ and $X_{j,t}$ are independent for all $i \neq j$. Since $X_{i,s}$ and $X_{i,t}$ are never independent for $i \in \{1, \ldots, n\}$, we will assume that they are independent if their correlation is less than 0.01. Since $\sigma^2(t) = 1$, the correlation between $X_{i,s}$ and $X_{i,t}$ for $i \in \{1, \ldots, n\}$ is equal to their covariance. Fix $i \in \{1, \ldots, n\}$, assume without loss of generality that $s < t$ and write $X_s = X_{i,s}$ and $X_t = X_{i,t}$. The covariance is now given by:

$$\operatorname{Cov}(X_s, X_t) = \mathbb{E}\left[(X_s - \mathbb{E}[X_s])(X_t - \mathbb{E}[X_t]\right] \tag{5.35}$$

Using Formula (5.3) and Formula (5.20):

$$\begin{aligned}
&= \mathbb{E}\left[\sqrt{2}\int_0^s e^{u-s}\,\mathrm{d}W_u \sqrt{2}\int_0^t e^{v-t}\,\mathrm{d}W_v\right] \\
&= 2e^{-s-t}\mathbb{E}\left[\int_0^s e^u\,\mathrm{d}W_u \int_0^t e^v\,\mathrm{d}W_v\right] \\
&= 2e^{-s-t}\mathbb{E}\left[\int_0^s e^u\,\mathrm{d}W_u \left(\int_0^s e^v\,\mathrm{d}W_v + \int_s^t e^v\,\mathrm{d}W_v\right)\right]
\end{aligned}$$

Using that the increments of a Brownian motion are independent:

$$= 2e^{-s-t}\mathbb{E}\left[\left(\int_0^s e^u\,\mathrm{d}W_u\right)^2\right]$$

Using Itô's isometry:

$$\begin{aligned} &= 2e^{-s-t}\mathbb{E}\left[\int_0^s e^{2u}\,\mathrm{d}u\right] \\ &= 2e^{-s-t}\left[\frac{1}{2}e^{2u}\right]_0^s \\ &= e^{s-t} - e^{-s-t} \end{aligned}$$

We now solve $\mathrm{Cov}(X_s, X_t) < 0.01$:

$$\begin{aligned} e^{s-t} - e^{-s-t} &< 0.01 \\ e^s - e^{-s} &< 0.01e^t \\ \ln(e^s - e^{-s}) &< \ln(0.01) + t \\ t &> \ln(e^s - e^{-s}) - \ln(0.01) \end{aligned} \tag{5.36}$$

If we use $t = s + 5$, we get the following inequality:

$$t = s + 5 > \ln(e^s) + \ln(100) > \ln(e^s - e^{-s}) - \ln(0.01) \tag{5.37}$$

So we have $\mathrm{Cov}(X_s, X_{s+5}) < 0.01$. If we restrict our results to $\theta(5k)$, $k \in \{0, \ldots, 100\}$, we can safely assume to have independent samples. We still need to validate that these samples are normally distributed. For this we will use a Q-Q plot and the Shapiro-Wilk test as described in Section 3.3. The Q-Q plot can be seen in Figure 6.

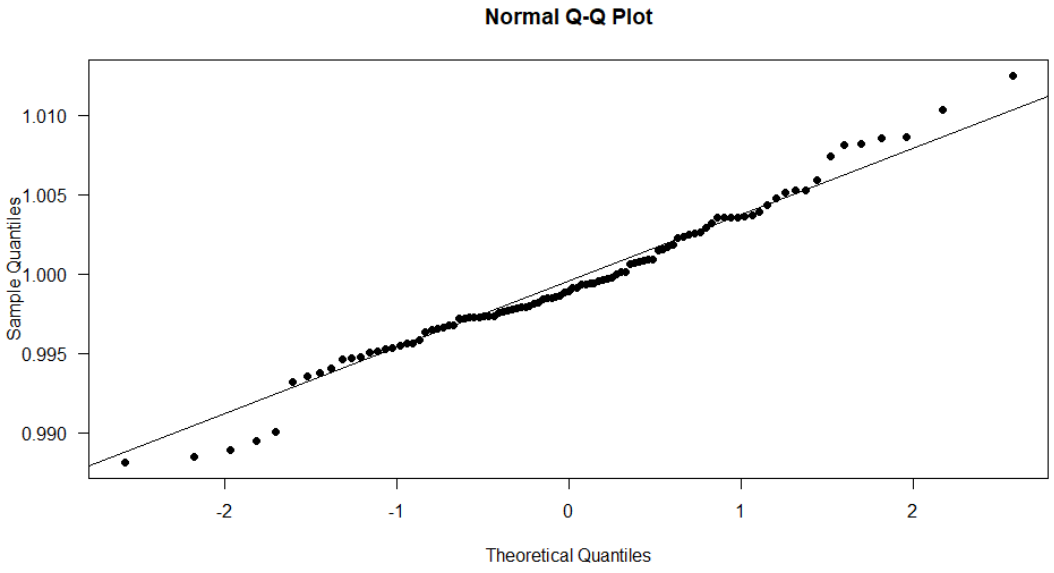


Figure 6: Q-Q plot of the data

The Shapiro-Wilk test gives a $p$-value of 0.1339, meaning that there is no clear evidence to reject normality of the data. The Q-Q plot also supports this conclusion. Therefore we can safely apply the $t$-test. The $t$-test gives a $p$-value of 0.1857, meaning that there is no clear evidence to reject that the mean of our simulation is 1. Therefore we can conclude that our second method used in this simulation is indeed unbiased.

### 5.3.3. Simulation 2: non-stationary distribution

For the second simulation we use $\mu_0 \neq R$ and $\sigma_0^2 \neq 1$. Specifically, we set $\mu_0$ four standard deviations away from $R$, where we use the standard deviations of the stationary distribution. We set $\sigma_0^2$ to two times the variance of the stationary distribution. We do this to see how the model behaves when it is far away from the stationary distribution. The other parameters are the same as in simulation 1. A full list of the parameters is shown in Table 2.

Table 2: Choices of the parameters

| parameter | value |
|---|---|
| $\mu_0$ | 4 |
| $\sigma_0^2$ | 2 |
| $R$ | 0 |
| $c_\tau$ | 0.1 |
| $c_k$ | 0.1 |
| $\theta_0$ | 1 |

We let $t$ run from 0 to 50. The result of both methods can be seen in Figure 7.

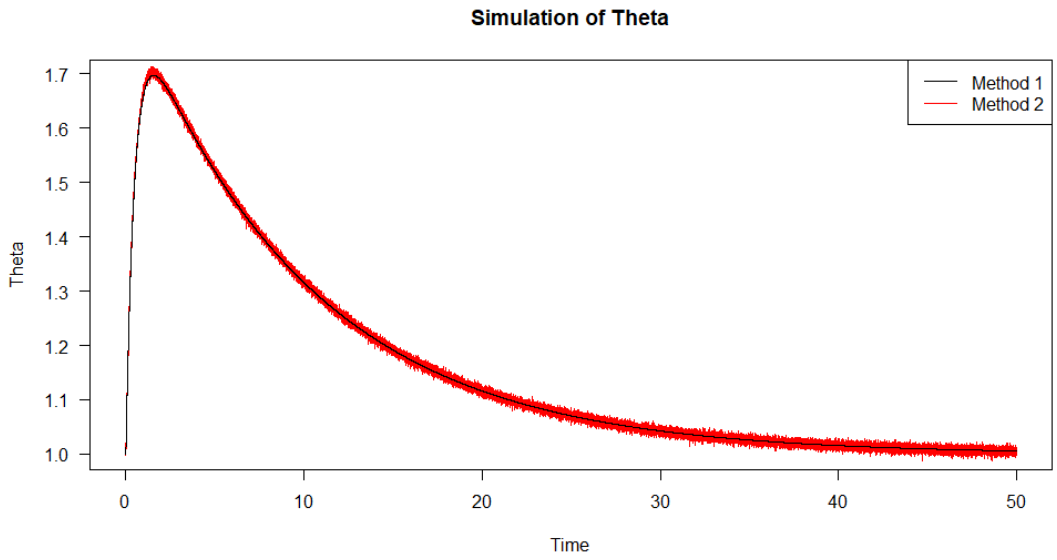


Figure 7: Simulation of $\theta$

We first analyze the distance between the two functions generated by the two methods. For this, we use a relative distance. We define the distance of $f$ relative to $g$ on the interval $[a, b]$ as:

$$d_{[a,b]}(f \mid g) := \sqrt{\frac{\int_a^b |f(t) - g(t)|^2 \, \mathrm{d}t}{\int_a^b g(t)^2 \, \mathrm{d}t}} \tag{5.38}$$

Since the simulation gives us discrete data, we compute the relative distance in the following

way:

$$d_{[0,50]}(f \mid g) = \sqrt{\frac{\sum_{i=0}^{50/h} |f(t_i) - g(t_i)|^2 h}{\sum_{j=0}^{50/h} g(t_i)^2 h}} \tag{5.39}$$

In simulation 2, the distance of method 2 relative to method 1 is $4.09 \cdot 10^{-4}$. Next, we want to analyze whether or not method 2 is unbiased. We will test its bias against method 1. To perform a $t$-test, we again need independent and normally distributed data. Since the covariance between $X_s$ and $X_t$ did not depend on the initial distribution, we can still use inequality (5.37). The variance of $X_s$ and $X_t$ is at least 1 for all $s, t > 0$, so the correlation between $X_s$ and $X_t$ is smaller than their covariance for all $s, t > 0$. Therefore, the correlation between $X_s$ and $X_{s+5}$ is smaller than 0.01 in the second simulation. Write $\Delta_\theta(t)$ for the difference between the values of $\theta$ of method 1 and method 2 at time $t$. To check whether or not $\Delta_\theta(5k)$, $k \in \{1, \ldots, 10\}$ are normally distributed, we use a Q-Q plot and the Shapiro-Wilk test. The Q-Q plot can be seen in Figure 8. The Shapiro-Wilk test gives a $p$-value of 0.2293, meaning that there is no clear evidence to reject normality of the data. The Q-Q plot supports this conclusion. We can now apply a $t$-test to our data. The $t$-test gives a $p$-value of 0.4567, meaning that method 2 is unbiased with respect to method 1.

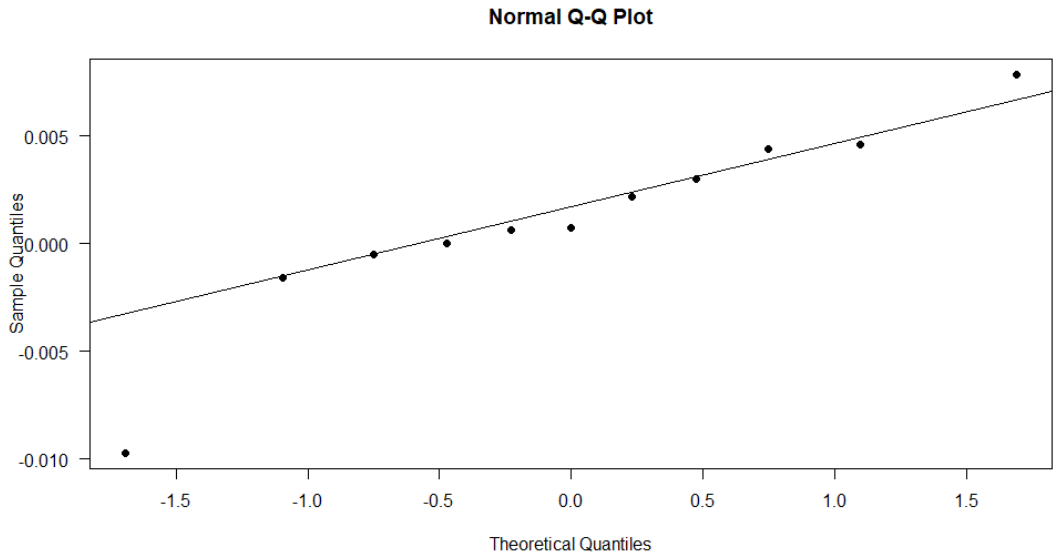


Figure 8: Q-Q plot of $\Delta_\theta$

## 5.4. Analysis of the samplesize and stepsize

To analyze the effect that the samplesize $n$ and the stepsize $h$ have on the simulation, we repeat simulation 2 as described in Section 5.3.3 for different samplesizes or stepsizes. For each pair of samplesize and stepsize, we repeat the simulation 100 times. With every simulation we look at the value of $\theta$ at $t = 5$. We choose $t = 5$ as our point of comparison because at this point the simulation has been running for a while, but the system has not reached its stationary distribution yet. Therefore we expect to see the biggest differences around this time for the different samplesizes or stepsizes. For each combination of stepsize and samplesize, we compute the average value of $\theta$ at $t = 5$ and the so called 95%-confidence intervals. This is an interval

$[a, b]$ for which it holds that $\mathbb{P}(\theta(5) \in [a, b]) = 0.95$. Since $\theta$ is normally distributed for fixed $t$, as shown above, the 95%-confidence intervals are given by:

$$\left[\mathbb{E}[\theta(t)] - 1.96\sqrt{\mathrm{Var}(\theta(t))}\,,\, \mathbb{E}[\theta(t)] + 1.96\sqrt{\mathrm{Var}(\theta(t))}\right] \tag{5.40}$$

We first analyze the effect of the samplesize $n$. The stepsize will be fixed by $h = 0.01$. The results of the simulation can be seen in Figure 9. Note that the x-axis has a logarithmic scale. As expected, the variance of $\theta$ decreases as the samplesize increases. However, the mean is constant for all the samplesizes tested.

To analyze the effect of the stepsize $h$, we fix the samplesize at $n = 1000$ and run the simulation for different stepsizes. The results can be seen in Figure 10. Note that again the x-axis has a logarithmic scale. It appears that the stepsize has little to no effect on simulation at $t = 5$ for a samplesize $n = 1000$. To see what effect the stepsize has on the overall simulation, we repeat simulation 2 as described in Section 5.3.3 for $h = 1$. The results are shown in Figure 11. The method used still performs well when the rate of change of the slope is not too big. At the peak value of $\theta$, this is however the case, resulting in a too high peak.

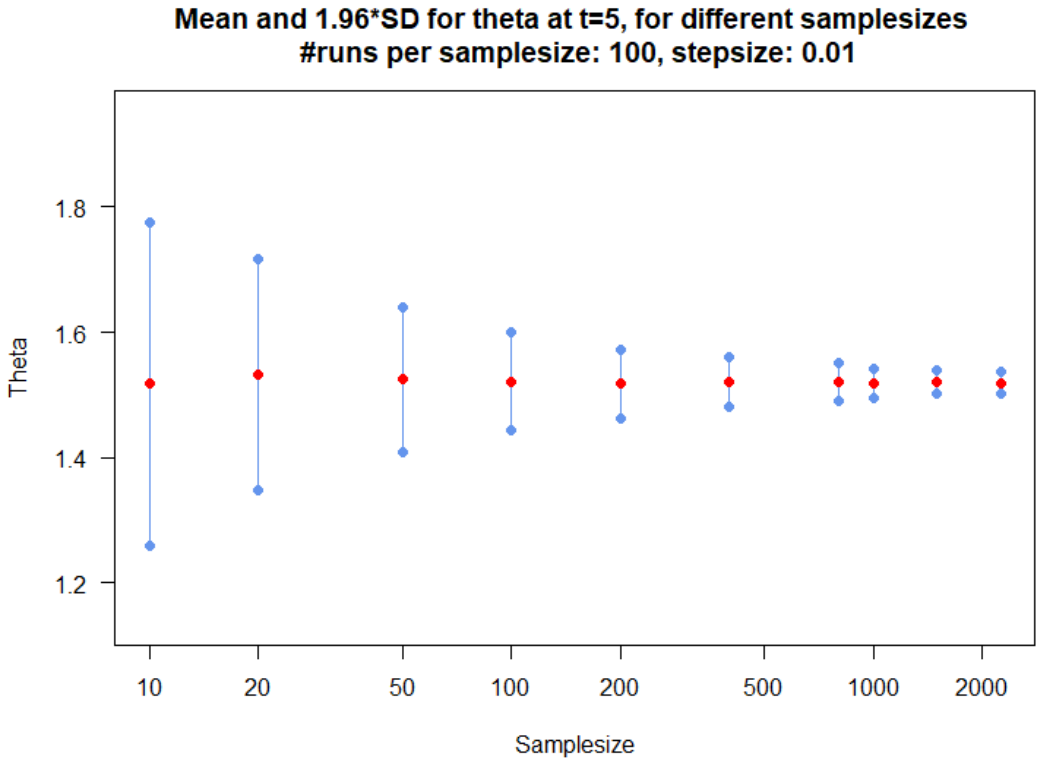


Figure 9: Mean and 95%-confidence intervals for different samplesizes

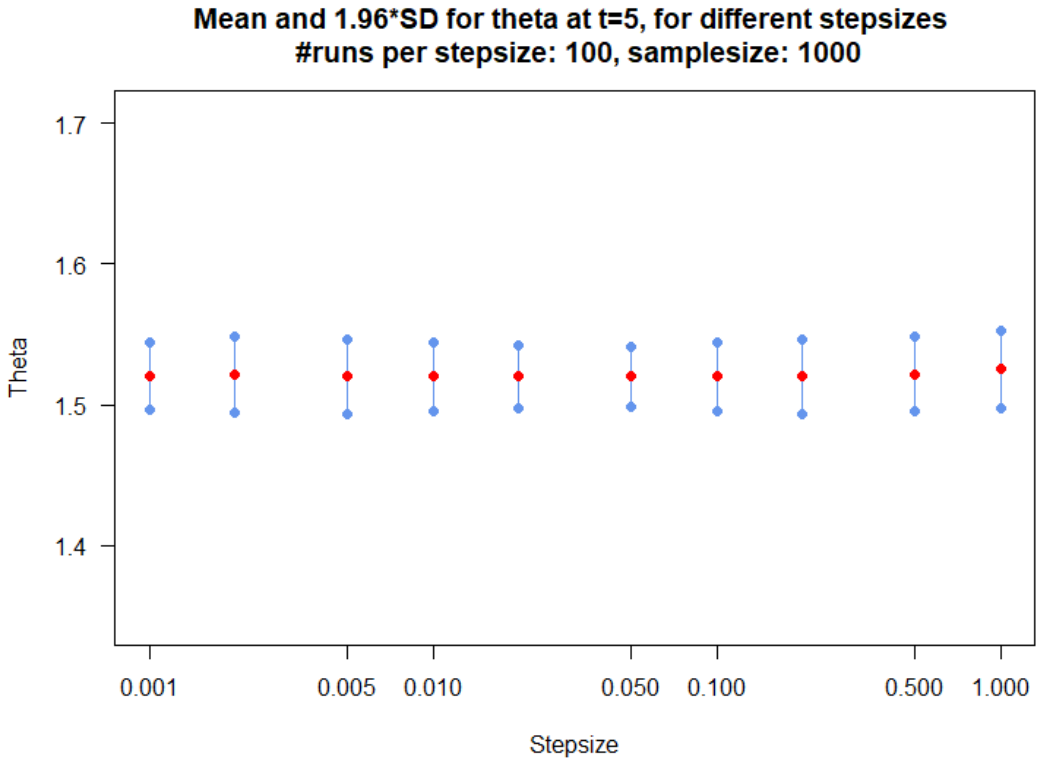


Figure 10: Mean and 95%-confidence intervals for different stepsizes

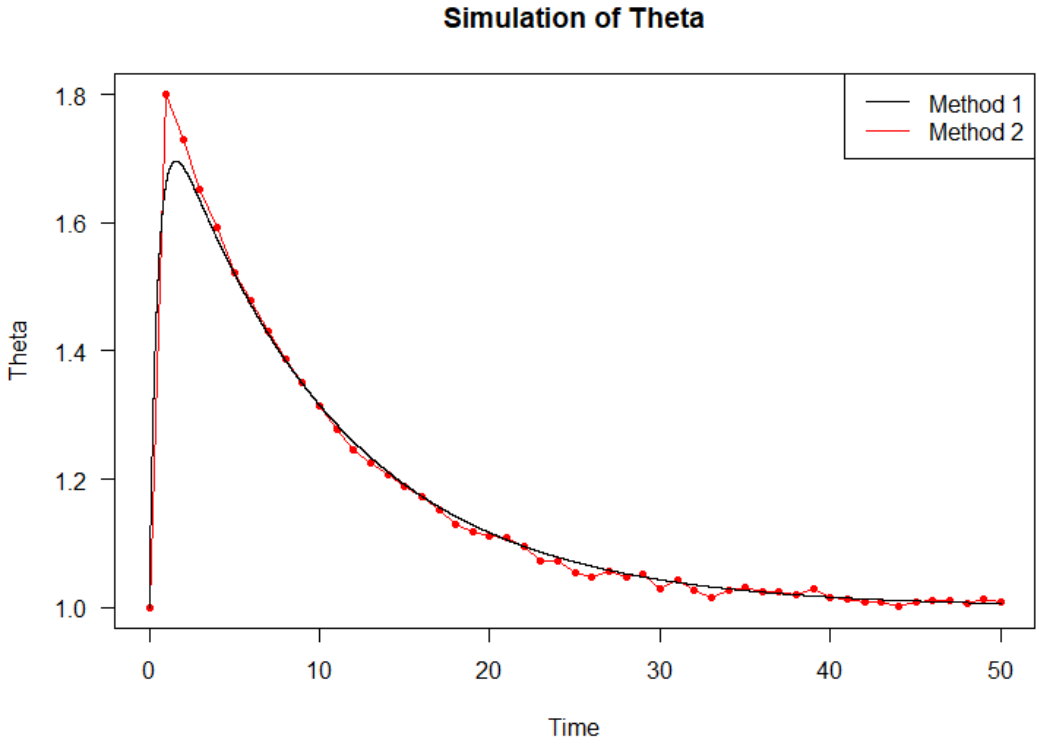


Figure 11: Simulation of $\theta$ with stepsize $h = 1$

To further analyze the variance of $\theta$ at $t = 5$ as a function of the samplesize, we plot the variance against the different samplesizes tested in our simulation. The results can be seen in

Figure 12. Here, both the x-axis and the y-axis have a logarithmic scale. The fitted line is given by:

$$\text{Var}(\theta) = 0.172\frac{1}{n} \tag{5.41}$$

with $n$ the sample size. This suggests that the variance decays at a rate of 1 over the samplesize. These results are however purely numerical based on our simulations. To obtain analytical results further research must be conducted on the time-dependent behaviour of $H^{\text{NN}}$. For example, the covariance between $H^{\text{NN}}_{n,m}$ and $H^{\text{NN}}_{n,m+1}$ plays a crucial role in the analysis of the variance of $\theta$. Research has already been done on the variance of $H^{\text{NN}}_n$ by Delattre and Fournier (2017). They proved, under some conditions on the distribution $X$, that $H^{\text{NN}}_n$ satisfies a central limit theorem. No research has however, as far as we know, been conducted on the time depended behaviour of $H^{\text{NN}}_n$.

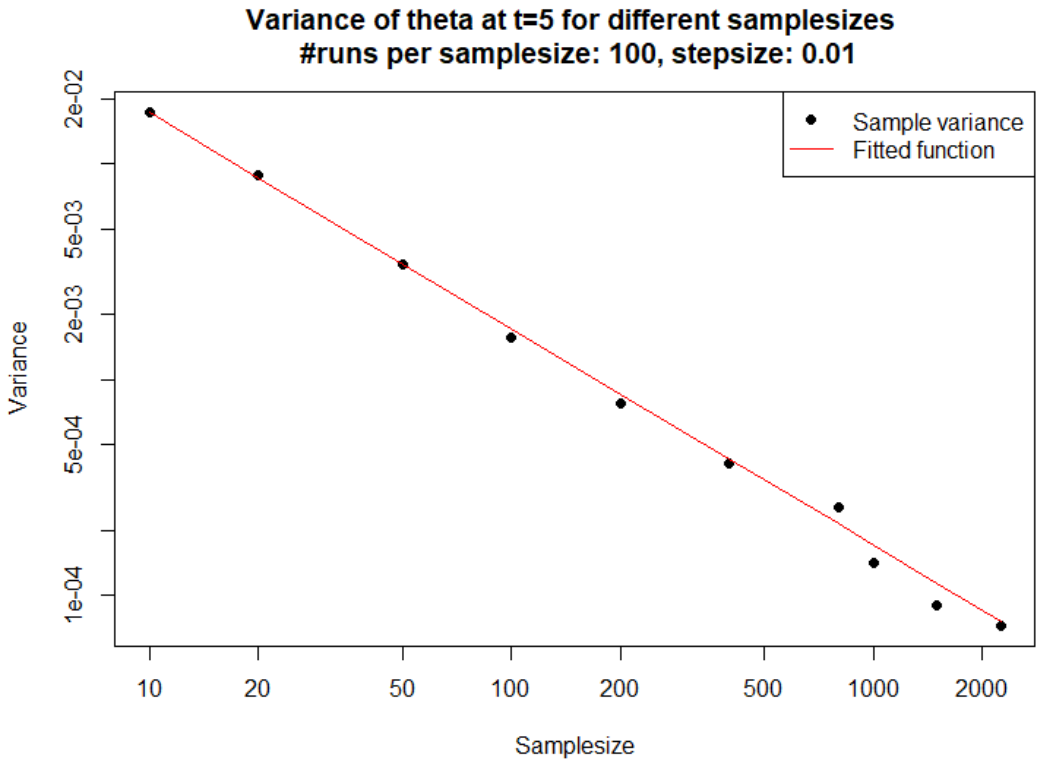


Figure 12: Variance of $\theta$ for different samplesizes

## 5.5. Conclusion

Under the assumption that $\tau_\alpha$ and $\tau_R$ are constant in $x$ and $\theta$, the systems become decoupled, which allows us to find the exact distribution of $X_t$. When furthermore $X_0$ is normally distributed, $X_t$ is normally distributed with mean $\mu(t)$ and variance $\sigma^2(t)$ for all $t > 0$. With this knowledge, we can find an explicit expression for the ordinary differential equation in terms of $\theta$ and $t$. This differential equation can be numerically solved by existing methods.

We formulated a discrete model to approximate the set of equations. The model is unbiased, both when approximating a stationary distribution and a non-stationary distribution. The stepsize used in this discrete model has no big influence on the long term behaviour of $\theta$, but a too big stepsize will result in less accurate results when large oscillations happen. Simulations show that increasing the samplesize lowers the variance, at a rate of 1 over the samplesize. To proof this analytically, further research needs to be done.

# 6. Full model

After analyzing the simplified model, we now turn our eye towards the full model as described in Section 1.1. We apply a similar analysis as we did to the simple model. First we analyze the stationary distribution of the stochastic differential equation and rewrite the ordinary differential equation. Next we discretize the model and run multiple simulations to analyze the behaviour of the model. Finally, we see what impact the samplesize and stepsize have on our model.

Recall that the full model, with $\tau_\alpha$ and $\tau_R$ non-constant is given by (The derivation of the model from the model of Semkiv et al. (2017) is shown in Appendix A):

$$\mathrm{d}X_t = \frac{1}{\tau_\alpha}(X_t - R)\,\mathrm{d}t + \left(\frac{\partial}{\partial x}\frac{1}{\tau_\alpha}\right)\mathrm{d}t + \sqrt{\frac{2}{\tau_\alpha}}\,\mathrm{d}W_t \tag{6.1}$$

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = -\frac{1}{\tau_R}(\theta - 1) + \frac{k_B}{\theta\eta_{,\theta}}\int_{\mathbb{R}} \frac{p_t}{\tau_\alpha}\left(x - R + \frac{\mathrm{d}p_t}{\mathrm{d}x}\frac{1}{p_t}\right)^2 \mathrm{d}x \tag{6.2}$$

with:

$$\tau_\alpha = \exp\left\{-c_1 + c_1\frac{c_2\theta}{c_2 + \theta - T_g(z, \Sigma_{loc})}\right\} \tag{6.3}$$

$$\tau_R = \exp\left\{-c_1 + c_1\frac{c_2\theta}{c_2 + \theta - T_g(z, 0)}\right\} \tag{6.4}$$

$$T_g(z, \Sigma_{loc}) = \frac{T_{g,b}}{T}\left(1 + \frac{\beta}{z}\right) - \frac{\Sigma_{loc}}{TK} = C - \frac{k_B}{K}|R(R - x)| \tag{6.5}$$

with $R$, $k_B$, $\theta\eta_{,\theta}$, $c_1$, $c_2$, $C$ and $K$ constant, and $p_t$ being the probability density function of $X_t$.

## 6.1. Fokker-Planck equation

We first analyze the Fokker-Planck equation of the stochastic differential equation (6.1). The Fokker-Planck equation is given by:

$$\begin{aligned}\frac{\partial}{\partial t}p &= \frac{\partial}{\partial x}\left(\frac{1}{\tau_\alpha}(x - R)p\right) - \frac{\partial}{\partial x}\left(\left(\frac{\partial}{\partial x}\frac{1}{\tau_\alpha}\right)p\right) + \frac{\partial^2}{\partial x^2}\left(\frac{1}{\tau_\alpha}p\right) \\ &= \frac{\partial}{\partial x}\left(\frac{1}{\tau_\alpha}(x - R)p\right) - \frac{\partial}{\partial x}\left(\left(\frac{\partial}{\partial x}\frac{1}{\tau_\alpha}\right)p\right) + \frac{\partial}{\partial x}\left(\left(\frac{\partial}{\partial x}\frac{1}{\tau_\alpha}\right)p + \frac{1}{\tau_\alpha}\left(\frac{\partial}{\partial x}p\right)\right) \\ &= \frac{\partial}{\partial x}\left(\frac{1}{\tau_\alpha}(x - R)p\right) + \frac{\partial}{\partial x}\left(\frac{1}{\tau_\alpha}\frac{\partial}{\partial x}p\right)\end{aligned} \tag{6.6}$$

We can not solve this equation, since $\tau_\alpha = \tau_\alpha(x, \theta)$. However, we can find the stationary solution, by solving $\frac{\partial}{\partial t}p = 0$ and writing $p(x, t) = p(x)$:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{\tau_\alpha}(x - R)p\right) + \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{\tau_\alpha}\frac{\mathrm{d}}{\mathrm{d}x}p\right) &= 0 \\ \frac{1}{\tau_\alpha}(x - R)p + \frac{1}{\tau_\alpha}\frac{\mathrm{d}}{\mathrm{d}x}p &= c\end{aligned} \tag{6.7}$$

since $\lim_{x\to\infty} p(x) = \lim_{x\to\infty} \frac{\mathrm{d}}{\mathrm{d}x}p(x) = 0$, we get $c = 0$, so:

$$\begin{aligned}\frac{1}{\tau_\alpha}(x - R)p + \frac{1}{\tau_\alpha}\frac{\mathrm{d}}{\mathrm{d}x}p &= 0 \\ \frac{\mathrm{d}}{\mathrm{d}x}p + (x - R)p &= 0\end{aligned}$$

Using the method of integrating factor:

$$\begin{aligned} e^{\frac{1}{2}(x-R)^2}\frac{\mathrm{d}}{\mathrm{d}x}p + e^{\frac{1}{2}(x-R)^2}(x-R)p &= 0 \\ \frac{\partial}{\partial x}\left(e^{\frac{1}{2}(x-R)^2}p\right) &= 0 \\ e^{\frac{(x-R)^2}{2}}p &= \hat{c} \\ p &= \hat{c}e^{-\frac{1}{2}(x-R)^2} \end{aligned}$$

Normalizing such that $p$ is a density gives:

$$p = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}(x-R)^2} \tag{6.8}$$

or equivalently:

$$X_t \sim N(R,1) \tag{6.9}$$

## 6.2. Ordinary differential equation

Using the Fokker-Planck equation, we can rewrite the ordinary differential equation (6.2) in a similar way as we did in Section 5.2. Namely, we rewrite the equation as follows:

$$\begin{aligned} \frac{\mathrm{d}\theta}{\mathrm{d}t} &= -\frac{1}{\tau_R}(\theta-1) + \frac{k_B}{\theta\eta_{,\theta}}\int_{\mathbb{R}}\frac{p_t}{\tau_\alpha}\left(x-R+\frac{\mathrm{d}p_t}{\mathrm{d}x}\frac{1}{p_t}\right)^2\mathrm{d}x \\ &= \frac{1}{\tau_R}(\theta-1) + \frac{k_B}{\theta\eta_{,\theta}}\int_{\mathbb{R}}\frac{p_t}{\tau_\alpha}\left(x-R+\frac{\mathrm{d}p_t}{\mathrm{d}x}\frac{1}{p_t}\right)\left(x-R+\frac{\mathrm{d}p_t}{\mathrm{d}x}\frac{1}{p_t}\right)\mathrm{d}x \\ &= \frac{1}{\tau_R}(\theta-1) + \frac{k_B}{\theta\eta_{,\theta}}\int_{\mathbb{R}}\left((x-R)\frac{p_t}{\tau_\alpha}+\frac{\mathrm{d}p_t}{\mathrm{d}x}\frac{1}{\tau_\alpha}\right)\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{2}(x-R)^2+\ln p_t+1\right)\mathrm{d}x \end{aligned} \tag{6.10}$$

Using partial integration and assuming that $p_t$ decays sufficiently fast:

$$= \frac{1}{\tau_R}(\theta-1) - \frac{k_B}{\theta\eta_{,\theta}}\int_{\mathbb{R}}\frac{\mathrm{d}}{\mathrm{d}x}\left((x-R)\frac{p_t}{\tau_\alpha}+\frac{\mathrm{d}p_t}{\mathrm{d}x}\frac{1}{\tau_\alpha}\right)\left(\frac{1}{2}(x-R)^2+\ln p_t+1\right)\mathrm{d}x$$

Using the Fokker-Planck equation (6.6):

$$\begin{aligned} &= \frac{1}{\tau_R}(\theta-1) - \frac{k_B}{\theta\eta_{,\theta}}\int_{\mathbb{R}}\left(\frac{\partial}{\partial t}p_t\right)\left(\frac{1}{2}(x-R)^2+\ln p_t+1\right)\mathrm{d}x \\ &= \frac{1}{\tau_R}(\theta-1) - \frac{k_B}{\theta\eta_{,\theta}}\int_{\mathbb{R}}\frac{\partial}{\partial t}\left(\frac{1}{2}(x-R)^2p_t+\ln(p_t)p_t\right)\mathrm{d}x \\ &= \frac{1}{\tau_R}(\theta-1) - \frac{\partial}{\partial t}\frac{k_B}{\theta\eta_{,\theta}}\left(\mathbb{E}\left[\frac{1}{2}(x-R)^2\right]-H(p_t)\right) \end{aligned} \tag{6.11}$$

## 6.3. Simulation

To analyze the behaviour of the model, we formulate a discrete version of the model, which we then simulate under different conditions.

### 6.3.1. Setup

Using Equation (6.1) and Equation (6.11), we can formulate a discretization of the model similar to how we did in Section 5.3. Let $X_{1,m}, \ldots, X_{n,m}, \theta_m$ denote the state-space at $t_m = t$ and let $t_{m+1} = t + h$, with $h$ being the stepsize. We compute $X_{1,m+1}, \ldots, X_{n,m+1}, \theta_{m+1}$ as follows:

$$X_{i,m+1} = X_{i,m} - \frac{h}{\tau_\alpha(X_{i,m}, \theta_m)}(X_{i,m} - R) + h\left(\frac{\mathrm{d}}{\mathrm{d}x}\frac{1}{\tau_\alpha}\right)(X_{i,m}, \theta_m) + \sqrt{\frac{2h}{\tau_\alpha(X_{i,m}, \theta_m)}}W_{i,m} \tag{6.12}$$

$$\theta_{m+1} = \theta_m - \frac{h}{\tau_R(\theta_m)}(\theta_m - 1) - \frac{k_B}{\theta\eta}(F_{m+1} - F_m) \tag{6.13}$$

with

$$W_{i,m} \sim N(0,1) \text{ i.i.d.} \tag{6.14}$$

$$F_m = \frac{1}{2}\frac{1}{n}\sum_{j=1}^{n}(X_{j,m} - R)^2 - H^{\mathrm{NN}}_{n,m} \tag{6.15}$$

with $H^{NN}_{n,m}$ as described in Section 4.3.4, using $X_{1,m}, \ldots, X_{n,m}$. With $\tau_\alpha$ and $\tau_R$ as formulated in Equation (6.3) and Equation (6.4) respectively.

The values of the parameters used in the simulation are given in Table 4. Most values are derived from Table 1 in Semkiv et al. (2017). The parameters they used are given in Table 3. We use the superscript $^*$ to indicate that the parameter corresponds with their parameter.

Table 3: Parameters used by Semkiv et al. (2017)

| parameter | value |
|---|---|
| $c_1^*$ | 12.8 |
| $c_2^*$ | 34 |
| $K^*$ | $2 \cdot 10^{-20}$ |
| $\theta\eta_{,\theta}^*$ | $2.5 \cdot 10^5$ |
| $T_{g,b}^*$ | 213 |
| $T^*$ | 263 |
| $\beta^*$ | $9 \cdot 10^{-10}$ |
| $\tau_R^{0*}$ | 0.74 |
| $\tau_g^*$ | 100 |
| $d^*$ | $2 \cdot 10^{-8}$ |
| $n^*$ | $1.05 \cdot 10^{23}$ |

Since we re-scaled the model, we also re-scale some of the parameters. As described in Appendix A, we re-scale $c_2^*$ with a factor $T^*$. For $R^*$, they use an initial distribution with $\mathbb{E}[|R^*|] = 2.71 \cdot 10^{-8}$. We will use this value of $R^*$ to derive the value of our parameters. Since $z^* = \frac{R^* - d^*}{2}$, formulated by them, we get $C^* \overset{(6.5)}{=} \frac{T_{g,b}^*}{T^*}\left(1 + \frac{\beta^*}{z^*}\right) = \frac{267}{263}$. Because this is a dimensionless quantity, we do not have to re-scale it. Note that the model, in contrary to the simplified model, is not shift-invariant in $R$, due to the Formula (6.5) for $T_g$.

In their paper, $p_t(x)^*$, the probability density function of $X_t$ is normalized in such a way that integration over $x$ equals $n^*$. Since our probability density function $p_t(x)$ is normalized such that integration over $x$ equals 1, we need to compensate for this difference. By taking the normalization factor outside of the integral, we can write $\theta\eta_{,\theta} = \frac{\theta\eta_{,\theta}^*}{n^*}$. In their paper, they set $\theta\eta_{,\theta}$ to $2.5 \cdot 10^5$. If we copy this, the term in front of the integral in Equation (6.2) is of

the order of $10^{-6}$. This would mean that the integral term has almost no influence on $\theta$. To amplify the effect of the integral term we set $\theta\eta^*_{,\theta} = 1$.

Finally, $k_B$ is the Boltzmann constant. All our parameters are given in Table 4.

Table 4: Parameters of the simulation

| parameter | value |
|---|---|
| $c_1$ | 12.8 |
| $c_2$ | $\frac{34}{263}$ |
| $k_B$ | $1.380 \cdot 10^{-23}$ |
| $K$ | $2 \cdot 10^{-20}$ |
| $\theta\eta_{,\theta}$ | $\frac{1}{1.05 \cdot 10^{23}}$ |
| $C = \frac{T_{g,b}}{T}\left(1 + \frac{\beta}{z}\right)$ | $\frac{267}{263}$ |
| $\tau_R^0$ | 0.74 |
| $\tau_g$ | 100 |
| $R$ | 626 |

For $X_0$ we will use a normal distribution with mean $\mu_0$ and variance $\sigma_0^2$. For $\theta$ we use $\theta(0) = 1$. For numerical reasons, we bound the functions $\tau_\alpha$, $\tau_R$ and $\frac{\partial}{\partial x}\frac{1}{\tau_\alpha}$ in the following way:

$$0.01 \leq \tau_\alpha \leq 100 \tag{6.16}$$

$$0.01 \leq \tau_R \leq 100 \tag{6.17}$$

$$-100 \leq \frac{\partial}{\partial x}\frac{1}{\tau_\alpha} \leq 100 \tag{6.18}$$

### 6.3.2. Simulation 1: stationary distribution

For the first simulation, we analyze the behaviour of the model in the stationary distribution. This means that $\mu_0 = R$ and $\sigma_0^2 = 1$. We run the simulation from $t = 0$ to $t = 500$. We use a stepsize $h$ of 0.01. The results can be seen in Figure 13. To analyze whether or not the simulation is unbiased, we take a similar approach as we took in Section 5.3.2. Since we do not have an explicit formula for $X_t$, we can not calculate the covariance of $X_t$ and $X_s$. Therefore, we will assume that the results found in Section 5.3.2 still hold, and that $\theta(s)$ and $\theta(s+5)$ can be regarded as independent. We can now perform a Shapiro-Wilk test as described in Section 3.3 on the samples $\theta(5k)$, $k \in \{0, \ldots, 100\}$. The test gives a $p$-value of 0.2205, meaning that there is no clear evidence to reject the normality of the data. Therefore we can apply a $t$-test to our data. This gives a $p$-value of 0.2526. Thus we can conclude that our simulation is unbiased.

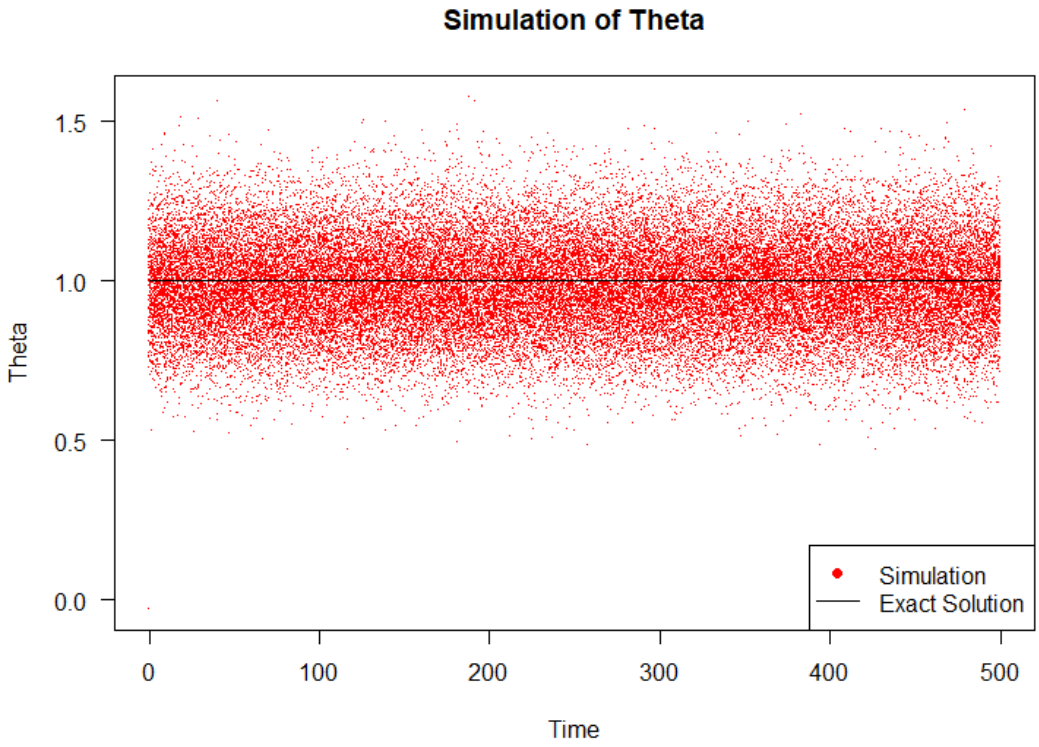


Figure 13: Simulation of $\theta$

6.3.3. Simulation 2: non-stationary distribution

For the second simulation, we analyze the model when $X_0$ is not distributed according to the stationary distribution. Instead, we let $X_0$ be a normal distribution with a mean four standard deviations away from the stationary distribution, so $\mu_0 = 630$. We also double the variance of $X_0$ by setting $\sigma_0^2 = 2$. This time, we only let the time run from $t = 0$ to $t = 0.2$ because we re-scaled $t$ in the model with a factor $\tau_g$. This has as a result that $\theta$ converges towards the stationary distribution before $t = 0.10$. To compensate for this, we set the stepsize to $h = 0.0001$. The results are shown in Figure 14. As expected, the model converges towards the stationary distribution. Therefore we can conclude that our simulation is in that sense consistent.

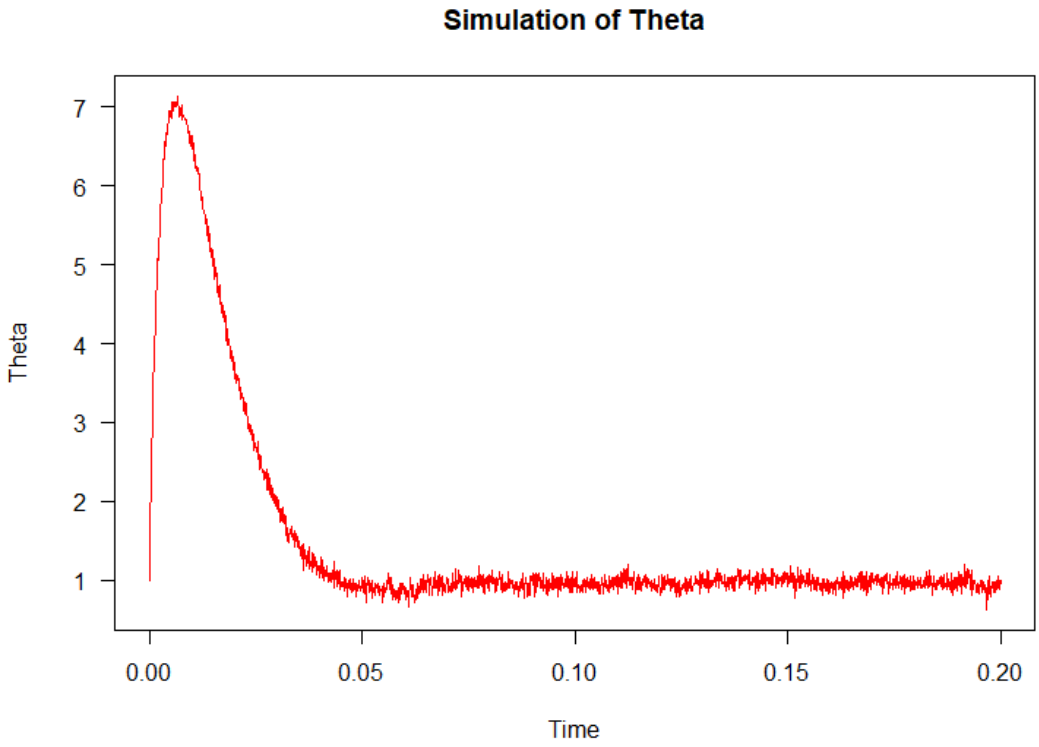


Figure 14: Simulation of $\theta$

## 6.4. Analysis of the samplesize and the stepsize

To analyze the effect of the samplesize and stepsize, we perform a similar analysis as we did in Section 5.4. We first analyze the effect of the samplesize. To do this, we fix the stepsize at $h = 0.0001$. With this stepsize, we repeat the simulation described in Section 6.3.3 for different samplesizes. For each samplesize, we run the simulation 100 times, and compute the mean and 95%-confidence intervals at $t = 0.2$. We do not know the exact distribution of $\theta$, so we assume that Formula (5.40) gives appropriate estimates. The results can be seen in Figure 15. Note that the x-axis has a logarithmic scale.

To further analyze the effect of the samplesize, we plot the variance of $\theta$ at $t = 0.2$ for the different samplesizes. The results can be seen in Figure 16. Note that both the x-axis and the y-axis have a logarithmic scale. The fitted line is given by:

$$\mathrm{Var}(\theta) = 5.5\frac{1}{n} \tag{6.19}$$

The fitted line suggests that the variance of $\theta$ in the full model decays at the same rate as in the simplified model when the samplesize $n$ increases. These results are, just as in Section 5.4, purely based on simulations. For analytical bounds on the variance we experience the same problem as we did for the simplified model. This time even more terms come into play, as $X_1, \ldots, X_n$, and thus $H_n^{\mathrm{NN}}$, now also depend on $\theta$.

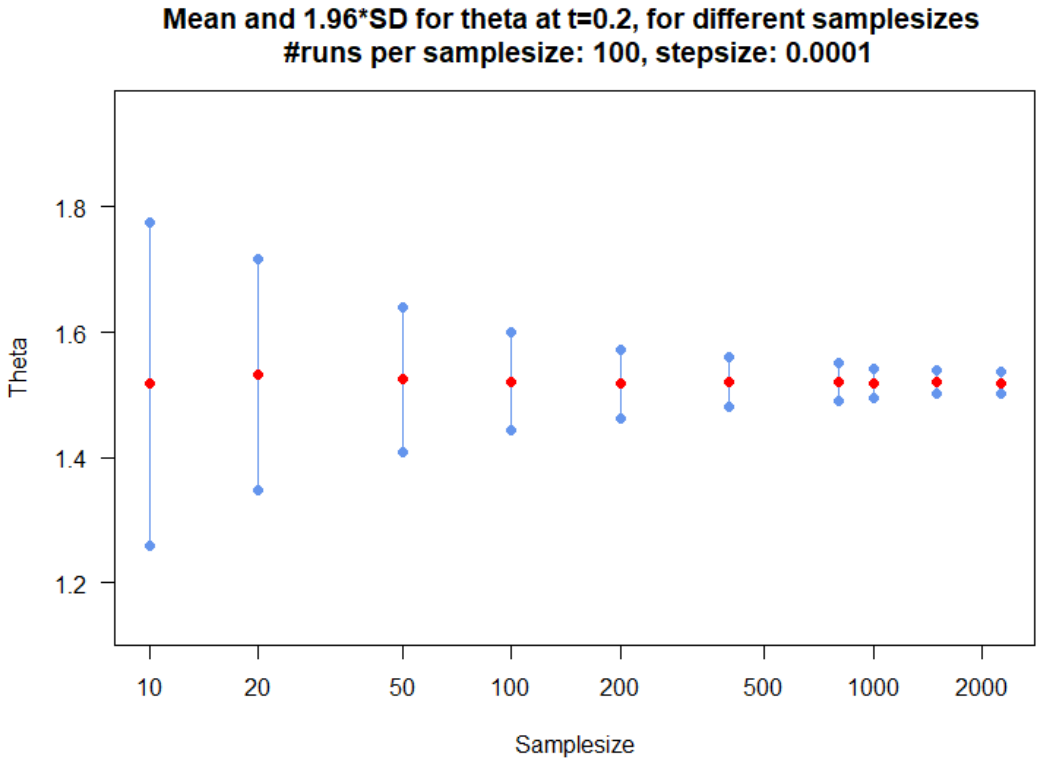


Figure 15: Mean and 95%-confidence intervals for different samplesizes

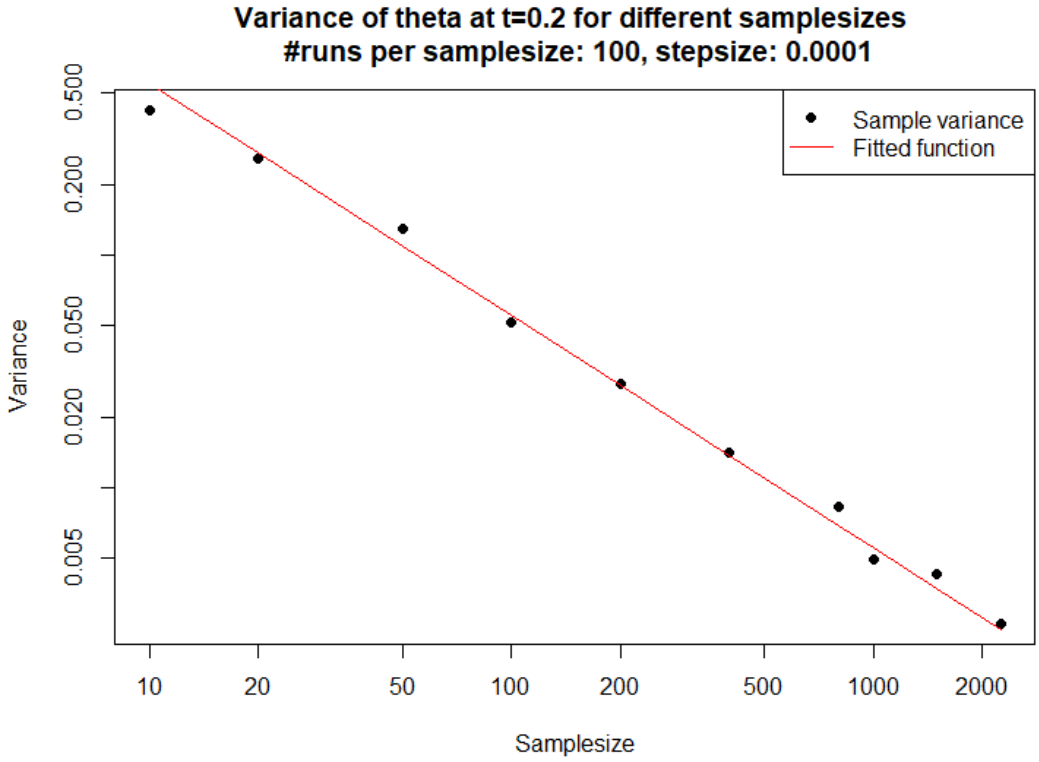


Figure 16: Variance of $\theta$ for different samplesizes

To analyze the stepsize, we follow the same procedure as for the samplesize, but instead of fixing the stepsize, we fix the samplesize. We fix the samplesize to $n = 1000$ and test different

stepsizes. The results can be seen in Figure 17. Note that again the x-axis has a logarithmic scale. Just as we had with the simplified model, the stepsize seems to have little impact on the variance of $\theta$. The variance of $\theta$ grows significantly with the largest stepsize we tested, suggesting that a too large stepsize does have an influence.

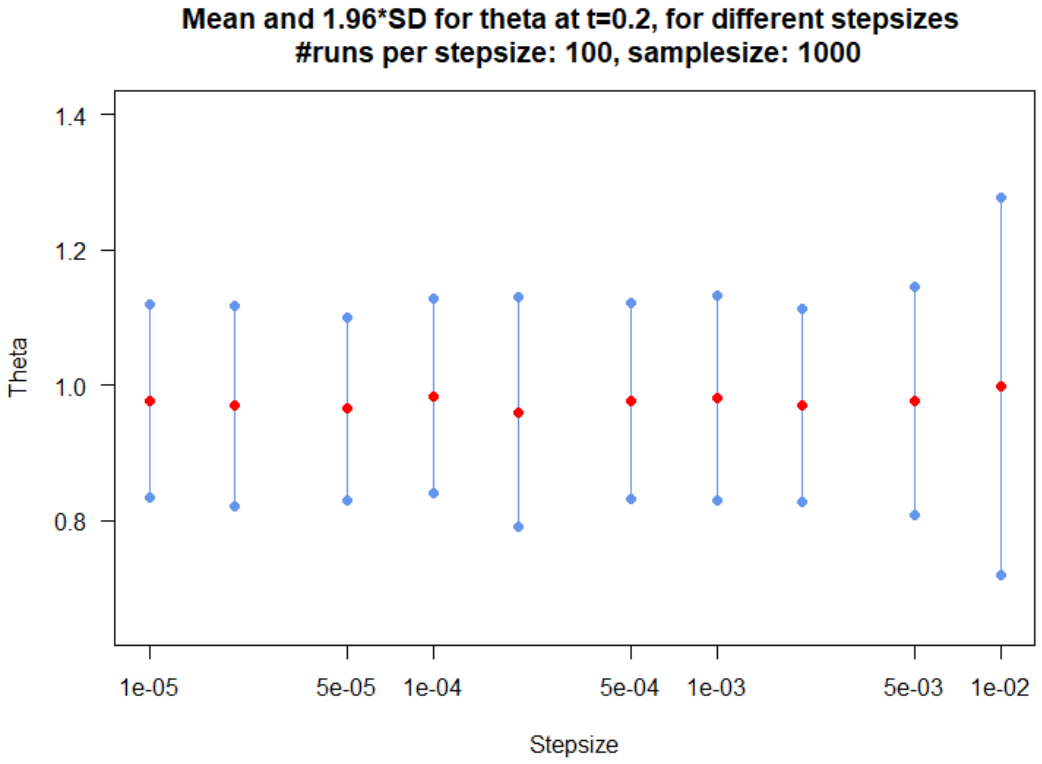


Figure 17: Mean and 95%-confidence interval for different stepsizes

## 6.5. Conclusion

For the full model, less exact results can be found in comparison to the simplified model. The stationary distribution of $X_t$, however, can still be found, which is a normal distribution with mean $R$ and variance 1.

The discrete model we build to simulate the process is unbiased when applied to the stationary distribution. For non-stationary distributions, the method appears to be consistent, as it converges to the stationary distribution.

Simulation show that the variance in the model decreases as the samplesize $n$ increases. The variance of $\theta$ appears to be of the order $\frac{1}{n}$. The stepsize has little influence on the variance of $\theta$, given that it is sufficiently small.

# 7. Conclusion

## 7.1. Conclusion and discussion

We analyzed four different methods to estimate the entropy of a random variable. To do this, we tested their performance against a standard normal distribution. The resubstitution method we used has the lowest mean square error of the methods examined, followed by the cross-validation method, the sample-spacing method and finally the nearest-neighbour method. The nearest-neighbour method turned out to be the only method without a significant bias for a samplesize of 1000.

We used the latter estimator to simulate a model of coupled differential equations. The model consists of a stochastic differential equation for $X(t)$, which depends on $\theta(t)$, and an ordinary differential equation for $\theta(t)$, which depends in a non-linear way on the distribution of $X(t)$.

We first analyzed a decoupled version of the model, for which an exact distribution for $X(t)$ could be found. Under the assumption that $X_0$ is normal distributed, $X(t)$ stays normally distributed with a mean and variance that shift towards the stationary distribution, which is a normal distribution with mean $R$ and variance 1. With this exact distribution, the ordinary differential equation could be rewritten to an explicit differential equation in terms of $\theta$ and $t$. This differential equation could then be solved by existing methods.

Knowing the exact solution, we compared it to our simulation, to see how it performs. The simulation of $\theta$ is unbiased in both the stationary distribution and in a non-stationary distribution, independent of the stepsize $h$ and samplesize $n$ according to our simulations. The variance of $\theta$ in the simulation is of the order of $\frac{1}{n}$. The stepsize $h$ has little influence on the variance of $\theta$

For the full - coupled - model, it no longer holds true that $X(t)$ stays normally distributed if $X_0$ is normally distributed. The stationary distribution does stay the same, being a normal distribution with mean $R$ and variance 1.

When applying the simulation to the full model, the simulation of $\theta$ stays unbiased for the stationary distribution. For non-stationary distribution, the simulation is consistent, in a sense that it converges towards the stationary distribution. Simulations suggest that the variance of $\theta$ is of the order of $\frac{1}{n}$ with $n$ being the samplesize. The stepsize $h$ has little influence on the variance of $\theta$, given that it is sufficiently small.

## 7.2. Future research

As pointed out in Section 5.4, further research can be conducted on analytically analyzing the model. Most of the results obtained in this paper rely on simulation and although this can give a good idea of how the model behaves, it does not function as a real proof.

More time can be spent at analyzing other methods to estimate the non-linear function of the density of $X(t)$ that appears in the ordinary differential equation. One such method would be to estimate the integral directly using a kernel density estimation for $p_t$.

Lastly, we only tested the estimation methods against a standard normal distribution. Since we know that $X(t)$ does not stay normally distributed in the full model, it can be useful to also test the estimation methods against other distributions. If other methods perform better at estimating the entropy of $X(t)$, they should be preferred.

# References

I.A. Ahmad and P.E.A. Lin. A nonparametric estimation of the entropy for absolutely continuous distributions (corresp.). *IEEE Transactions on Information Theory*, 22(3):372–375, 1976.

J. Beirlant, E.J. Dudewicz, L. Györfi, and E.C. Van der Meulen. Nonparametric entropy estimation: An overview. *International Journal of Mathematical and Statistical Sciences*, 2001.

S. Delattre and N. Fournier. On the kozachenko–leonenko entropy estimator. *Journal of Statistical Planning and Inference*, 185:69–93, 2017.

L Euler. *Institutionum calculi integralis*. impensis Academiae imperialis scientiarum, 1824.

P. Hall. Limit theorems for sums of general functions of m-spacings. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 96, pages 517–532. Cambridge University Press, 1984.

K. Itô. 109. stochastic integral. *Proceedings of the Imperial Academy*, 20(8):519–524, 1944.

A. V. Ivanov and M. N. Rozhkova. On properties of the statistical estimate of the entropy of a random vector with a probability density. *Problemy Peredachi Informatsii*, 17(3):34–43, 1981.

H. Joe. Estimation of entropy and other functionals of a multivariate density. *Annals of the Institute of Statistical Mathematics*, 41:683–697, 1989.

L.F. Kozachenko and N.N. Leonenko. Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, 23(2):9–16, 1987.

G.A. Pavliotis. *Stochastic processes and applications*. Springer, 2011.

M. Semkiv, P.D. Anderson, and M. Hütter. Two-scale model for the effect of physical aging in elastomers filled with hard nanoparticles. *Journal of Computational Physics*, 350:184–206, 2017.

S.S. Shapiro and M.B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3-4):591–611, 1965.

B.W. Silverman. *Density estimation for statistics and data analysis*. Chapman & Hall/CRC, 1986.

K. Soetaert, T. Petzoldt, and R. Woodrow Setzer. Solving differential equations in R: Package deSolve. *Journal of Statistical Software*, 33(9):1–25, 2010.

Student. The probable error of a mean. *Biometrika*, 6(1):1–25, 1908.

J.A. Thomas and T.M. Cover. Elements of information theory. *John Wiley & Sons, Inc.*, 6: 187–202, 1991.

# A. Derivation of the model

By reducing the dimensions from three to one, writing $X = Q$, and fixing $R$, the model proposed by Semkiv et al. (2017) as formulated by Equations (1.3) up to (1.11) can be written as:

$$\begin{aligned}
\mathrm{d}X_t &= -\frac{1}{\tau_\alpha}\left(X_t - R\right)\mathrm{d}t + \frac{k_B T}{k}\left(\frac{\mathrm{d}}{\mathrm{d}x}\frac{1}{\tau_\alpha}\right)\mathrm{d}t + \sqrt{\frac{2k_B T}{k\tau_\alpha}}\mathrm{d}W \\
\frac{\mathrm{d}\theta}{\mathrm{d}t} &= -\frac{1}{\tau_R}(\theta - T) + \frac{1}{\theta\eta_{,\theta}}\int_{\mathbb{R}}\frac{p_t}{k\tau_\alpha}\left(k(x-R) + k_B T\frac{\mathrm{d}p_t}{\mathrm{d}x}\frac{1}{p_t}\right)^2\mathrm{d}x
\end{aligned}$$

with:

$$\begin{aligned}
\tau_\alpha &= \tau_g \exp\left\{-c_1\frac{c_2(T-\theta) + T(\theta - T_g(z,\Sigma_{loc}))}{T(c_2 + \theta - T_g(z,\Sigma_{loc}))}\right\} \\
\tau_R &= \tau_R^0 \exp\left\{-c_1\frac{c_2(T-\theta) + T(\theta - T_g(z,0))}{T(c_2 + \theta - T_g(z,0))}\right\} \\
T_g(z,\Sigma_{loc}) &= T_{g,b}\left(1 + \frac{\beta}{z}\right) - \frac{\Sigma_{loc}}{K}
\end{aligned}$$

The formula for $\Sigma_{loc}$ is changed. This is because $(tr\mathbf{\Sigma})^2 - tr(\mathbf{\Sigma}\cdot\mathbf{\Sigma})$ becomes 0 if $\mathbf{\Sigma}$ is a one dimensional matrix (a single number). To still capture the effect, a new formula is formulated for the one dimensional model and given by:

$$\Sigma_{loc} = |R \cdot k(R - X_t)|$$

In order to make the model dimensionless, we use the following re-scaling: $X = \widetilde{X}\sqrt{\frac{k_B T}{k}}$, $R = \widetilde{R}\sqrt{\frac{k_B T}{k}}$, $p = \widetilde{p}\sqrt{\frac{k}{k_B T}}$ (Since the dimension of $p$ is the reciprocal of the dimension of $X$), $t = \widetilde{t}\tau_g$, $\theta = \widetilde{\theta}T$, $c_2 = \widetilde{c_2}T$, $T_g = \widetilde{T_g}T$, $\tau_\alpha = \widetilde{\tau_\alpha}\tau_g$ and $\tau_R = \widetilde{\tau_R}\tau_g$. Since $\mathbb{E}[(W_t)^2] = t$, we need to re-scale $W$ with a factor $\sqrt{\tau_g}$, so $W = \widetilde{W}\sqrt{\tau_g}$. Applying this re-scaling to the model gives:

$$\mathrm{d}\widetilde{X_t}\sqrt{\frac{k_B T}{k}} = -\frac{1}{\widetilde{\tau_\alpha}\tau_g}\left(\widetilde{X_t}\sqrt{\frac{k_B T}{k}} - \widetilde{R}\sqrt{\frac{k_B T}{k}}\right)\mathrm{d}\widetilde{t}\tau_g + \frac{k_B T}{k}\left(\frac{\mathrm{d}}{\mathrm{d}\widetilde{x}\sqrt{\frac{k_B T}{k}}}\frac{1}{\widetilde{\tau_\alpha}\tau_g}\right)\mathrm{d}\widetilde{t}\tau_g + \sqrt{\frac{2k_B T}{k\widetilde{\tau_\alpha}\tau_g}}\mathrm{d}\widetilde{W}\sqrt{\tau_g}$$

And similar for the other equations. Rewriting gives the following (dimensionless) model:

$$\begin{aligned}
\mathrm{d}\widetilde{X_t} &= -\frac{1}{\widetilde{\tau_\alpha}}\left(\widetilde{X_t} - \widetilde{R}\right)\mathrm{d}\widetilde{t} + \left(\frac{\mathrm{d}}{\mathrm{d}\widetilde{x}}\frac{1}{\widetilde{\tau_\alpha}}\right)\mathrm{d}\widetilde{t} + \sqrt{\frac{2}{\widetilde{\tau_\alpha}}}\mathrm{d}\widetilde{W} \\
\frac{\mathrm{d}\widetilde{\theta}}{\mathrm{d}\widetilde{t}} &= -\frac{1}{\widetilde{\tau_R}}\left(\widetilde{\theta} - 1\right) + \frac{k_B}{\theta\eta_{,\theta}}\int_{\mathbb{R}}\frac{\widetilde{p}}{\widetilde{\tau_\alpha}}\left((\widetilde{x} - \widetilde{R}) + \frac{\mathrm{d}\widetilde{p}}{\mathrm{d}\widetilde{t}}\frac{1}{\widetilde{p}}\right)^2\mathrm{d}x
\end{aligned}$$

with:

$$\begin{aligned}
\widetilde{\tau_\alpha} &= \exp\left\{-c_1 + c_1\frac{\widetilde{c_2}\widetilde{\theta}}{\widetilde{c_2} + \widetilde{\theta} - \widetilde{T_g}(z,\widetilde{\Sigma_{loc}})}\right\} \\
\widetilde{\tau_R} &= \frac{\tau_R^0}{\tau_g}\exp\left\{-c_1 + c_1\frac{\widetilde{c_2}\widetilde{\theta}}{\widetilde{c_2} + \widetilde{\theta} - \widetilde{T_g}(z,0)}\right\} \\
\widetilde{T_g}(z,\widetilde{\Sigma_{loc}}) &= \frac{T_{g,b}}{T}\left(1 + \frac{\beta}{z}\right) - \frac{\widetilde{\Sigma_{loc}}}{TK} = \frac{T_{g,b}}{T}\left(1 + \frac{\beta}{z}\right) - \frac{k_B}{K}|\widetilde{R}(\widetilde{R} - \widetilde{x})|
\end{aligned}$$

# B. R code

```
# Global settings
e <- exp(1) #allows e^x to be used instead of exp(x)
# Required libraries:
library(ks)
library(deSolve)

##-------------------------------------------------------------------##
##------------------------ Estimating Entroppy ----------------------##
##-------------------------------------------------------------------##

#Evaluate point in kerneldensity based on data with bandwith h
fhatPoint <- function(point,data,h){
  return(1/(NROW(data)*h)*sum(dnorm((point-data)/h,0,1)))
}

# Evaluate each point in Points in kerneldensity based on Data with
    bandwith h
fhatAll <- function(Points,Data){
  if(NCOL(Data)==1){
    bandweight <- bw.nrd0(Data) # Silverman's rule of tumb
    fhatX <- sapply(Points,fhatPoint,data=Data,h=bandweight)
    return(fhatX)
  }else{
    return(kde(Data,eval.points = Points)$estimate) # How does it
    work?
  }
}

# Estimate entropy with data with Resub method
ResubSingle <- function(data){
  H <- -1/NROW(data)*sum(log(fhatAll(data,data)))
  return(H)
}

# Estimate entropy for each column of Data with Resub method
ResubFull <- function(Data){
  result <- apply(Data,2,ResubSingle)
  return(result)
}

# Estimate entropy with data with Crossvall method
CrossValSingle <- function(data){
  fhats <- sapply(1:NROW(data),function(x) fhatAll(data[x],data[-x])
    )
  H <- -1/NROW(data)*sum(log(fhats))
  return(H)
}

```

```
46 # Estimate entropy for each column of Data with Crossvall method
47 CrossValFull <- function(Data){
48   result <- apply(Data,2,CrossValSingle)
49   return(result)
50 }
51
52 # Compute m-spacing on ordered data
53 MDistance <- function(data,M){
54   l <- NROW(data)
55   mdistance <- sapply(1:(l-M),function(i) (data[i+M]-data[i]) )
56   return(mdistance)
57 }
58
59 # Estimate entropy with data with samplespacing method
60 SampleSpaceSingle <- function(data,m){
61   data <- sort(data)
62   H <- 1/NROW(data)*sum(log(NROW(data)/m*MDistance(data,m)))-digamma
     (m)+log(m)
63   return(H)
64 }
65
66 # Estimate entropy for each column with samplespacing method
67 SampleSpaceFull <- function(Data,M){
68   result <- apply(Data,2,SampleSpaceSingle,m=M)
69   return(result)
70 }
71
72 # Compute nearest neighbours of X
73 NND <- function(X){
74   NND <- sapply(1:length(X), function(x) min(abs(X[x]-X[-x])))
75   return(NND)
76 }
77
78 # Estimate entropy with data with nn method
79 NNSingle <- function(data){
80   H <- 1/NROW(data)*sum(log(NND(data)))+log(2)-digamma(1)+log(NROW(
     data)-1)
81   return(H)
82 }
83
84 # Estimate entropy for each column with nn method
85 NNFull <- function(Data){
86   result <- apply(Data,2,NNSingle)
87   return(result)
88 }
89
90 # Generate the plot as used in section 4.4
91 PlotGood <- function(result,Exact,title){
92   plot(density(result),main = title)
93   abline(v=Exact,col="red")
```

```
94    abline(v=mean(result),col="green")
95    legend("topright",legend = c("Exact","Calculated mean"),col=c("red
       ","green"),lty=1)
96    print(paste("MSE: ",mean((result-Exact)^2) ))
97  }
98
99  # Example of using the code above
100
101 # Testing on normal data
102 mu <- 0
103 sd <- 1
104 Exact <- log(sqrt(2*pi*e)*sd)
105 Nsamples <- 1000
106 Nruns <- 100
107 X <- rnorm(Nsamples*Nruns,mu,sd)
108 X <- matrix(X,nrow=Nsamples,ncol=Nruns)
109 # Running Resub
110 result <- ResubFull(X)
111 PlotGood(result,Exact,paste("Estimated Entropy using Resubstitution,
       X~N(",mu,",",sd^2,")"))
112 # Running Crossval
113 result <- CrossValFull(X)
114 PlotGood(result,Exact,paste("Estimated Entropy using Crossvalidation
      , X~N(",mu,",",sd^2,")"))
115 # Running Samplespacing
116 M <- 1
117 result <- SampleSpaceFull(X,M)
118 PlotGood(result,Exact,paste("Estimated Entropy using Samplespacing,
     M=",M,", X~N(",mu,",",sd^2,")"))
119 # Running NN
120 result <- NNFull(X)
121 PlotGood(result,Exact,paste("Estimated Entropy using Nearest
     Neighbour, X~N(",mu,",",sd^2,")"))
122
123 ##-------------------------------------------------------------------##
124 ##------- Simulate simplfied model with tau_alpha constant -------##
125 ##-------------------------------------------------------------------##
126
127 # Simulation as described in section 5.3.1.2
128 SimTheta <- function(x0,h,T,R,Ct,Ck){
129   # Initialize variables
130   x <- x0
131   t <- 0
132   theta <- 1
133   FOld <- 0.5*mean((x-R)^2) - NNSingle(x)
134   timeList <- seq(0,T,by=h)
135   thetaList <- c(theta)
136   # Create progress bar
137   pb <- winProgressBar(title = "progress bar", min = 0,
138                        max = length(timeList), width = 300)
```

```
139   for(t in timeList[-length(timeList)]){
140     # Update x to next timestep
141     x <- x + -h*(x-R)+sqrt(2)*rnorm(length(x),mean=0,sd=sqrt(h))
142     # Calculate new F
143     FNew <- 0.5*mean((x-R)^2) - NNSingle(x)
144     # Update theta to next timestep
145     theta <- theta - Ct*h*(theta-1) - Ck*(FNew-FOld)
146     # Save all the required data
147     FOld <- FNew
148     thetaList <- c(thetaList,theta)
149     # Update progress bar
150     setWinProgressBar(pb,t/h,title = paste(round(t/T*100,0),"% done"
      ))
151   }
152   # Close progress bar and return results
153   close(pb)
154   return(list(timesteps = timeList,thetas = thetaList))
155 }
156
157 # Example of running the simulation
158 mu0 <- 0
159 var0 <- 1
160 R <- 0
161 Ct <- 0.1
162 Ck <- 0.1
163 start <- rnorm(1000,mu0,sqrt(var0))
164 result <- SimTheta(start,0.001,500,R,Ct,Ck)
165 # Generating plot of the results
166 plot(result$timesteps,result$thetas,pch=".",xlab="Time",las=1,
167      ylab="Theta",main="Simulation of Theta \n stepsize: 0.001",col=
      "red")
168 lines(c(0,500),c(1,1))
169 legend("topright",legend = c("Method 1","Method 2"),col=c("black","
      red"),lty=1)
170
171 RHS <- function(t,mu0,var0,R){
172   res <- (var0-1)*exp(-2*t)/((var0-1)*exp(-2*t)+1) - ((mu0-R)^2+var0
      -1)*exp(-2*t)
173 }
174
175 # Analyzing the results
176 var(result$thetas[result$timesteps%%5==0])
177 qqnorm(SimpleModel500$thetas[result$timesteps%%5==0])
178 qqline(SimpleModel500$thetas[result$timesteps%%5==0])
179 shapiro.test(SimpleModel500$thetas[result$timesteps%%5==0])
180 t.test(SimpleModel500$thetas[SimpleModel500$timesteps%%5==0],mu=1)
181
182 # "Exact" solution as described in section 5.3.1.1
183 parameters <- c(mu0 = 4, var0 = 2, R = 0, Ct = 0.1, Ck = 0.1)
184 state <- c(X = 1)
```

```
185 ODEFunc<-function(t, state, parameters) {
186   with(as.list(c(state, parameters)),{
187     # rate of change
188     dX <- -Ct*(X-1)-Ck*RHS(t,mu0,var0,R)
189     # return the rate of change
190     list(c(dX))
191   })
192 }
193 times <- seq(0,50,by=0.001)
194 output <- ode(y = state,
195               times = times,func = ODEFunc, parms = parameters)
196
197 # Generating plot to compare results
198 plot(output,type = "l",xlab="Time",las=1,
199      ylab="Theta",main="Simulation of Theta",ylim=c(1,max(result$
    thetas)))
200 points(result$timesteps,result$thetas,col="red",pch=20)
201 points(result$timesteps,result$thetas,col="red",type = "l")
202 lines(output)
203 legend("topright",legend = c("Method 1","Method 2"),col=c("black","
    red"),lty=1)
204
205 # Analysing the results
206 Relative <- sqrt(sum(abs(output[,"X"]-result$thetas)^2) / sum(output
    [,"X"]^2))
207 DeltaTheta <- (output[,"X"]-result$thetas)[result$timesteps%%5==0]
208 qqnorm(DeltaTheta)
209 qqline(DeltaTheta)
210 shapiro.test(DeltaTheta)
211 t.test(DeltaTheta,mu=0)
212
213
214 # Simulation for different samplesizes
215 thetaDist <- function(nRuns,h,T,samplesizes,param){
216   nVector <- samplesizes
217   meanVector <- c()
218   upperVector <- c()
219   lowerVector <- c()
220   R <- param$R
221   mu0 <- param$mu0
222   var0 <- param$var0
223   Ct <- param$Ct
224   Ck <- param$Ck
225   for(j in nVector){
226     results <- c()
227     for(i in 1:nRuns){
228       start <- rnorm(j,mu0,sqrt(var0))
229       resultRun <- SimTheta(start,h,5,R,Ct,Ck)
230       thetaT <- resultRun$thetas[length(resultRun$thetas)]
231       results <- c(results,thetaT)
```

```
232     }
233     print(paste(j," samples done")) # Optional to track progress
234     meann <- mean(results)
235     meanVector <- c(meanVector,meann)
236     upperVector <- c(upperVector,meann+1.96*sd(results))
237     lowerVector <- c(lowerVector,meann-1.96*sd(results))
238   }
239   return(list(means = meanVector, uppers = upperVector, lowers =
      lowerVector))
240 }
241
242 # Example of running simulation above
243 sampleList <- c(10,20,50,100,200,400,800,1000,1500,2250)
244 param <- list(R=0,mu0=4,var0=2,Ct=0.1,Ck=0.1)
245 myList2 <- thetaDist(100,0.01,5,sampleList,param)
246 plot(sampleList,myList2$means,ylim = c(min(myList2$lowers)*0.9,max(
      myList2$uppers)*1.1),log = 'x',col="red",
247       xlab = "Samplesize",ylab = "Theta",
248       main = "Mean and 1.96*SD for theta at t=5, for different
      samplesizes \n #runs per samplesize: 100, stepsize: 0.01")
249 points(sampleList,myList2$uppers,col="cornflowerblue")
250 points(sampleList,myList2$lowers,col="cornflowerblue")
251 segments(sampleList,myList2$lowers,sampleList,myList2$uppers,col = "
      cornflowerblue")
252 points(sampleList,myList2$means,col="red")
253
254 # Analyzing the variance
255 variance <- ((myList2$uppers-myList2$means)/1.96)^2
256 plot(sampleList,variance,log = 'x',xlab = "Samplesize", ylab = "
      Variance",
257       main = "Variance of theta at t=5 for different samplesizes \n #
      runs per samplesize: 100, stepsize: 0.01",pch=16)
258 lines(sampleList,0.172/sampleList,col="red")
259 legend("topright",legend = c("Sample variance","Fitted function"),
      col=c("black","red"),lty=c(NA,1),pch=c(16,NA))
260
261 plot(sampleList,variance,log = 'xy',xlab = "Samplesize", ylab = "
      Variance",
262       main = "Variance of theta at t=5 for different samplesizes \n #
      runs per samplesize: 100, stepsize: 0.01",pch=16)
263 lines(sampleList,0.172/sampleList,col="red")
264
265
266 # Simulation for different stepsizes
267 thetaStepsize <- function(nRuns,samplesize,T,stepsizes,param){
268   nVector <- stepsizes
269   meanVector <- c()
270   upperVector <- c()
271   lowerVector <- c()
272   R <- param$R
```

```
273   mu0 <- param$mu0
274   var0 <- param$var0
275   Ct <- param$Ct
276   Ck <- param$Ck
277   for(j in nVector){
278     results <- c()
279     for(i in 1:nRuns){
280       start <- rnorm(samplesize,mu0,sqrt(var0))
281       resultRun <- SimTheta(start,j,T,R,Ct,Ck)
282       thetaT <- resultRun$thetas[length(resultRun$thetas)]
283       results <- c(results,thetaT)
284     }
285     print(paste(j," stepsize done")) # Optional to track progress
286     meann <- mean(results)
287     meanVector <- c(meanVector,meann)
288     upperVector <- c(upperVector,meann+1.96*sd(results))
289     lowerVector <- c(lowerVector,meann-1.96*sd(results))
290   }
291   return(list(means = meanVector, uppers = upperVector, lowers =
      lowerVector))
292 }
293
294 # Example of running simulation above
295 stepsList <- c(0.001,0.002,0.005,0.01,0.02,0.05,0.1,0.2,0.5,1)
296 param <- list(R=0,mu0=4,var0=2,Ct=0.1,Ck=0.1)
297 myList1 <- thetaStepsize(100,1000,5,stepsList,param)
298 plot(stepsList,myList1$means,ylim = c(0.9*min(myList1$lowers) ,1.1*
      max(myList1$uppers)),log = 'x',col="red",
299      xlab = "Stepsize",ylab = "Theta",
300      main = "Mean and 1.96*SD for theta at t=5, for different
      stepsizes \n #runs per stepsize: 100, samplesize: 1000")
301 points(stepsList,myList1$uppers,col="cornflowerblue")
302 points(stepsList,myList1$lowers,col="cornflowerblue")
303 segments(stepsList,myList1$lowers,stepsList,myList1$uppers,col = "
      cornflowerblue")
304 points(stepsList,myList1$means,col="red")
305
306 ##-------------------------------------------------------------------##
307 ##---------------------- Full (dimensionless) model -----------------##
308 ##-------------------------------------------------------------------##
309
310 Tg <- function(x){
311   -kB/K*abs(R*(x-R))+C
312 }
313
314 TauAlpha <- function(theta,x){
315   Value <- exp(-c1+c1*( (c2*theta) / (c2+theta-Tg(x)) ))
316   upper <- min(Value,100)
317   return(max(upper,0.01))
318 }
```

```
319
320 TauR <-function(theta){
321   Value <- tauR0/taug*exp(-c1+c1*( (c2*theta) / (c2+theta-C) ))
322   upper <- min(Value,100)
323   return(max(upper,0.01))
324 }
325
326 TauAlphaDeriv <- function(theta,x){
327   Value <- 1/TauAlpha(theta,x)* (kB/K*R*sign(x-R)*c1*c2*theta) / (c2
      +theta-Tg(x))^2
328   upper <- min(Value,100)
329   return(max(upper,-100))
330 }
331
332 # Simultion as described in section 6.???
333 SimFull <- function(x0,h,T){
334   # Initialise variables
335   x <- x0
336   t <- 0
337   theta <- 1
338   FOld <- 0.5*mean((x-R)^2)-NNSingle(x)
339   timeList <- seq(0,T,by=h)
340   thetaList <- c(theta)
341   # Create progress bar
342   pb <- winProgressBar(title = "progress bar", min = 0,
343                        max = length(timeList), width = 300)
344   for(t in timeList[-length(timeList)]){
345     # Update x to next timestep
346     x <- x - h/TauAlpha(theta,x)*(x-R) + h*TauAlphaDeriv(theta,x) +
      sqrt(2/TauAlpha(theta,x))*rnorm(length(x),mean=0,sd=sqrt(h))
347     # Calculate new F
348     FNew <- 0.5*mean((x-R)^2) - NNSingle(x)
349     # Update theta to next timestep
350     theta <- theta - h/TauR(theta)*(theta-1) - kB/Eta*(FNew - FOld)
351     # Save all the required data
352     FOld <- FNew
353     thetaList <- c(thetaList,theta)
354     # Update progress bar
355     setWinProgressBar(pb, t/h, title=paste( round(t/T*100, 0),
356                                             "% done"))
357   }
358   # Close progress bar en return results
359   close(pb)
360   return(list(timesteps = timeList,thetas = thetaList))
361 }
362
363 # Example of running the simulation
364 # Starting Distribution
365 mu0 <- 630
366 var0 <- 2
```

```
# Parameters
R   <- 626                # rescaled
c1  <- 12.8               # normal
c2  <- 34/263             # rescaled
kB  <-1.380*10^(-23)      # normal
K   <- 2*10^(-20)         # normal
Eta <- 1/(1.05*10^(23))   # normal
C <- 267/263              # rescaled
tauR0 <-  0.74            # normal
taug <- 100               # normal
# System parameters and starting values
stepsize <- 0.0001
T <- 0.2
start <- rnorm(1000,mu0,sqrt(var0))
# Run the simulation
Run <- SimFull(start,h=stepsize,T)

# Plot theta
plot(Run$timesteps,Run$thetas,xlab="Time",las=1,
     ylab="Theta",main="Simulation of Theta",col="red",type = "l")
lines(c(0,T),c(1,1))
legend("bottomright",legend = c("Simulation","Exact Solution"),col=c
   ("red","black"),lty=c(NA,1),pch=c(16,NA))

# Analyse the data
shapiro.test(Run$thetas[Run$timesteps%%5==0])
t.test(Run$thetas[Run$timesteps%%5==0],mu=1)

# Simulation for different samplesizes
thetaFullDist <- function(nRuns,h,T,samplesizes){
  nVector <- samplesizes
  meanVector <- c()
  upperVector <- c()
  lowerVector <- c()
  for(j in nVector){
    results <- c()
    for(i in 1:nRuns){
      start <- rnorm(j,mu0,sqrt(var0))
      resultRun <- SimFull(start,h,T)
      thetaT <- resultRun$thetas[length(resultRun$thetas)]
      results <- c(results,thetaT)
    }
    print(paste(j," samples done")) # Optional to track progress
    meann <- mean(results)
    meanVector <- c(meanVector,meann)
    upperVector <- c(upperVector,meann+1.96*sd(results))
    lowerVector <- c(lowerVector,meann-1.96*sd(results))
  }
  return(list(means = meanVector, uppers = upperVector, lowers =
   lowerVector))
```

```
}

# Example of running simulation above
sampleList <- c(10,20,50,100,200,400,800,1000,1500,2250)
myList3 <- thetaFullDist(100,0.0001,0.2,sampleList)
par(las=1,pch=16)
plot(sampleList,myList2$means,ylim = c(min(myList2$lowers)*0.9,max(myList2$uppers)*1.1),log = 'x',col="red",
     xlab = "Samplesize",ylab = "Theta",
     main = "Mean and 1.96*SD for theta at t=0.2, for different samplesizes \n #runs per samplesize: 100, stepsize: 0.0001",las=1,pch=16)
points(sampleList,myList2$uppers,col="cornflowerblue")
points(sampleList,myList2$lowers,col="cornflowerblue")
segments(sampleList,myList2$lowers,sampleList,myList2$uppers,col = "cornflowerblue")
points(sampleList,myList2$means,col="red")

variance2 <- ((myList3$uppers-myList3$means)/1.96)^2
plot(sampleList,variance2,log = 'xy',xlab = "Samplesize", ylab = "Variance",
     main = "Variance of theta at t=0.2 for different samplesizes \n #runs per samplesize: 100, stepsize: 0.0001",pch=16)
lines(sampleList,5.5/sampleList,col="red")
legend("topright",legend = c("Sample variance","Fitted function"),col=c("black","red"),lty=c(NA,1),pch=c(16,NA))

# Simulation for different stepsizes
thetaFullStepsize <- function(nRuns,samplesize,T,stepsizes){
  nVector <- stepsizes
  meanVector <- c()
  upperVector <- c()
  lowerVector <- c()
  for(j in nVector){
    results <- c()
    for(i in 1:nRuns){
      start <- rnorm(samplesize,mu0,sqrt(var0))
      resultRun <- SimFull(start,j,T)
      thetaT <- resultRun$thetas[length(resultRun$thetas)]
      results <- c(results,thetaT)
    }
    print(paste(j," stepsize done")) # Optional to track progress
    meann <- mean(results)
    meanVector <- c(meanVector,meann)
    upperVector <- c(upperVector,meann+1.96*sd(results))
    lowerVector <- c(lowerVector,meann-1.96*sd(results))
  }
  return(list(means = meanVector, uppers = upperVector, lowers = lowerVector))
}
```

```
457
458 # Example of running simulation above
459 stepsList <- c(0.00001,0.00002,0.00005,
        0.0001,0.0002,0.0005,0.001,0.002,0.005,0.01)
460 myList4 <- thetaFullStepsize(100,1000,0.2,stepsList)
461 plot(stepsList,myList4$means,ylim = c(0.9*min(myList4$lowers) ,1.1*
        max(myList4$uppers)),log = 'x',col="red",
462       xlab = "Stepsize",ylab = "Theta",
463       main = "Mean and 1.96*SD for theta at t=0.2, for different
        stepsizes \n #runs per stepsize: 100, samplesize: 1000")
464 points(stepsList,myList4$uppers,col="cornflowerblue")
465 points(stepsList,myList4$lowers,col="cornflowerblue")
466 segments(stepsList,myList4$lowers,stepsList,myList1$uppers,col = "
        cornflowerblue")
467 points(stepsList,myList4$means,col="red")
```