AMSI
AUSTRALIAN MATHEMATICAL
SCIENCES INSTITUTE

National
Collaboration
in the Mathematical
Sciences

Drift Estimation on a Spatial Lattice

Justin Smallwood

February 26, 2014

Abstract

Diffusion processes arise in many places in the physical and economic sciences, for instance as models for stock prices, interest rates, cell movement and paths of molecular particles. The object of this paper is to review and derive several methods of estimation of these parameters, using a maximum likelihood and least-squares approach, and focusing on three specific processes, Brownian Motion with Drift, Geometric Brownian Motion, and the Ornstein-Uhlenbeck process. Connections with martingale theory are introduced and applied to parametric estimation of the drift.

1 Introduction

Diffusion processes arise in many places in the physical and economic sciences, for instance as models for stock prices, interest rates, cell movement and paths of molecular particles. Diffusion processes are characterised by several properties, heuristically they are random processes for which their future movement depends only on their current position. In addition, and important for use in modelling physical phenomena, is that the sample paths must be continuous functions of time almost everywhere[1]. Diffusions can be completely defined by their instantaneous mean, or drift, and instantaneous variance, or diffusion. The object of this paper is to review and derive several methods of estimation of these parameters, using a maximum likelihood and least-squares approach, and focusing on three specific processes, Brownian Motion with Drift, Geometric Brownian Motion, and the Ornstein-Uhlenbeck process. Connections with martingale theory are introduced and applied to parametric estimation of the drift.

[1] P($X(t)$ is continuous)=1.

1.1 Definition of Diffusion Process

Let us now provide a rigorous definition of a diffusion process. For completeness, we begin with the definition of a filtration,

Definition 1.1. *A filtration $\mathbb{F}$ is an indexed collection of σ-algebras $\mathcal{F}_t$, $t \in [0, T]$, such that for $s \leq t$, $\mathcal{F}_s \subseteq \mathcal{F}_t$.*

Definition 1.2. *Let $(\Omega, \mathcal{F}_t, \mathbb{P})$ be a probability triple with a filtration $\mathbb{F}$. A stochastic process is a collection of random variables $\{X(t)\}$ such that for any fixed t, $t \in [0, T]$, $X(t)$ is a random variable on $(\Omega, \mathcal{F}_T)$. A stochastic process is called adapted to filtration $\mathbb{F}$ if, for all $t \in [0, T]$, $X(t)$ is a random variable on $\mathcal{F}_t$, that is, if $X(t)$ is $\mathcal{F}_t$-measurable. (Klebaner, 2005)*

An important definition is that of a stopping time,

Definition 1.3. *A random variable τ is called a stopping time with respect to the filtration $\mathbb{F}$ if for each $t \in [0, T]$, the event $\{\tau \leq t\}$ is $\mathcal{F}_t$-measurable. That is, $\{\tau \leq t\} \in \mathcal{F}_t$.*

One can think of a stopping time in the context of a game, as a "rule" where one should cease to play the game. A stopping time can only depend on information available at any point in time, that is, it must be non-anticipatory, or $\mathcal{F}_t$-measurable. Now, we require that the diffusion process only take into account its current position when deciding upon its future movement, so we require that the process have the *Strong Markov Property*,

Definition 1.4. *A stochastic process $X(t)$ has the strong Markov property if, for any finite stopping time T, the regular conditional distribution of $X(T + t)$, $t \geq 0$ given $\mathcal{F}_T$, is $P_{X(t)}$, that is,*

$$P(X(T + t) \leq y | \mathcal{F}_T) = P(X(T + t) \leq y | X(T)) \ a.s.$$

(Klebaner, 2005)

Finally, we are in a position to rigorously define a diffusion process,

Definition 1.5. *A continuous time parameter stochastic process which possesses the (strong) Markov property and for which the sample paths $X(t)$ are (almost always) continuous functions of t is called a diffusion process. (Karlin and Taylor, 1981)*

1.2 Brownian Motion

The most well-known and important diffusion process is the Brownian Motion, or Wiener Process. This stochastic process is characterised by having stationary, independent increments, like a Poisson process, and furthermore these increments are normally distributed,

Definition 1.6. *Let* $\{B(t), \mathcal{F}_t\}, t \geq 0$, *where* $\{\mathcal{F}_t\}$ *is a filtration and* $B(t)$ *is an adapted process. Then* $B(t)$ *is a Standard Brownian Motion process if:*

1. $B(0) = 0$,

2. $B(t_2) - B(s_2)$, $B(t_1) - B(s_1)$, $t_2 > s_2 \geq t_1 > s_1$, *are independent random variables,*

3. $B(t) - B(s)$ *has the Normal distribution with mean* 0 *and variance* $t - s$. *(*$B(t) - B(s) \sim N(0, t-s)$*),* $t > s$,

4. $B(t)$, $t \geq 0$, *is an almost surely continuous function of* t.

The sample paths are almost surely continuous, however they are nowhere-differentiable[2]. As well as being an important modelling tool in its own right, Brownian Motion is the cornerstone of describing any diffusion process, through their representation as stochastic differential equations.

1.3 Stochastic Differential Equations

An important way of expressing a diffusion process is in *stochastic differential equation (SDE)* form. To this end, consider equations of the form

$$dX = \mu(X(t), t)dt + \sigma(X(t), t)dB(t),$$

where the functions $\mu(X(t), t)$, $\sigma(X(t), t)$ are given, and $B(t)$ is a standard Brownian Motion process. The function μ is called the drift of the the process, and σ is called the volatility.

[2] For a more detailed construction of the Wiener Process, refer to Grimmett and Stirzaker (2001).

An alternative way of defining the drift and diffusion is as the instantaneous mean and variance of the process,

$$\begin{aligned}\mu(x,t) &= \lim_{h\to 0}\frac{1}{h}E[\Delta X|X(t)=x],\\ \sigma^2(x,t) &= \lim_{h\to 0}\frac{1}{h}E[(\Delta X)^2|X(t)=x],\end{aligned}$$

where $\Delta X = X(t+h) - X(t)$.

If μ, σ do not depend on t, then the diffusion is (time) homogeneous.

1.4 Examples of Diffusion Processes

We will consider three diffusion processes, Brownian Motion with Drift, Geometric Brownian Motion, and the Ornstein-Uhlenbeck process. Let us consider each individually.

1.4.1 Brownian Motion with Drift

Brownian Motion with Drift differs slightly from regular Brownian Motion, in that the drift term is non-zero.

Definition 1.7. *Let $\{\mathcal{F}_t\}$ be a filtration and $\{X(t)\}$ an adapted process. Then $X(t)$ is a Brownian Motion process with drift if:*

1. $X(0) = 0$,
2. $X(t_2) - X(s_2)$, $X(t_1) - X(s_1)$, $t_2 > s_2 \geq t_1 > s_1$, *are independent random variables,*
3. $X(t) - X(s)$ *has the Normal distribution with mean* $\mu(t-s)$ *and variance* $\sigma^2(t-s)$. ($X(t) - X(s) \sim N(\mu(t-s), \sigma^2(t-s))$),
4. $X(t)$, $t \geq 0$, *is an almost-surely continuous function of* t.

In stochastic differential equation form, this process is expressed as

$$dX(t) = \mu\ dt + \sigma\ dB(t)$$

From this, it can be seen that the drift is μ, and the diffusion σ^2.

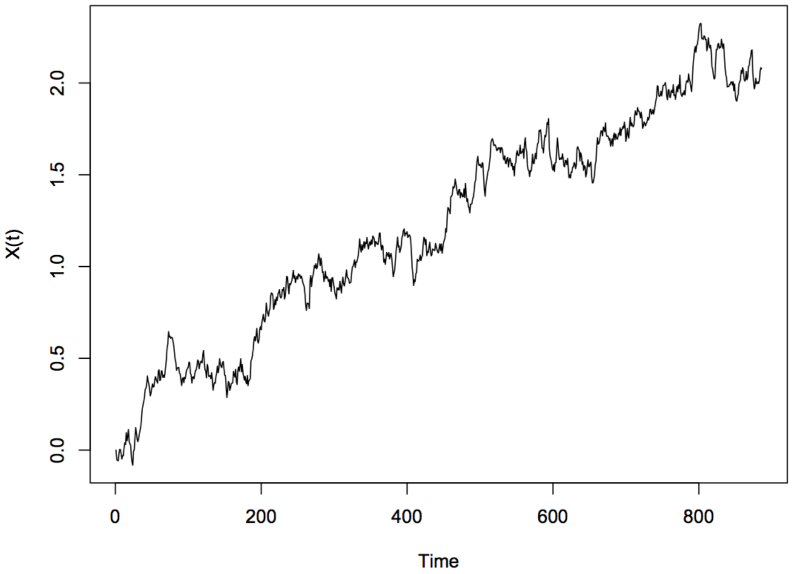

Figure 1: Sample path of a Brownian Motion with Drift process with $\mu = 2$, $\sigma = 1$.

1.4.2 Geometric Brownian Motion

One process used often in financial applications to model share prices is Geometric Brownian Motion. This process is particularly well-adapted to the task of modelling share prices due to the non-negativity of its sample paths.

Definition 1.8. *Let $X = \{X(t) : t \geq 0\}$ be a Brownian Motion process with drift μ and diffusion σ^2. The process defined by $e^{X(t)}$ is called Geometric Brownian Motion on state space $(0, \infty)$.*

In SDE form, we have,

$$dG(t) = (\mu + \frac{1}{2}\sigma^2)G(t)\ dt + \sigma G(t)\ dB(t),$$

where $B(t)$ is a Brownian Motion process.

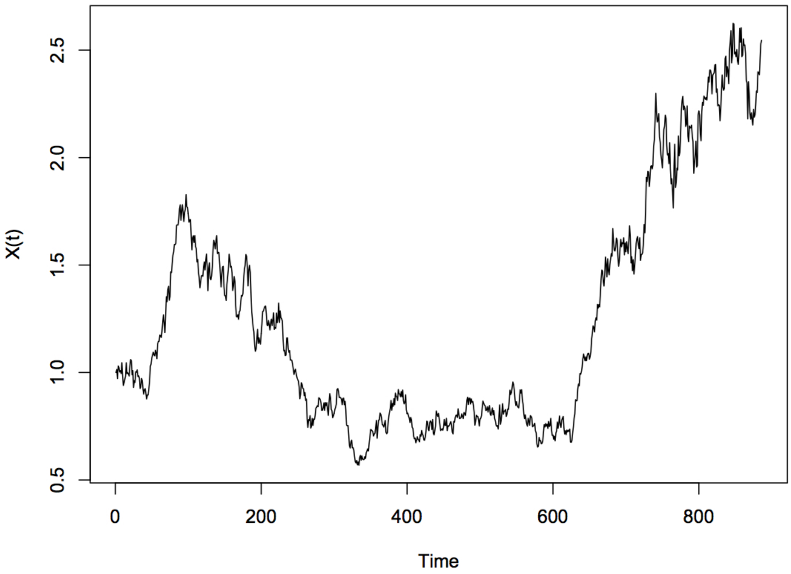

Figure 2: Sample path of a Geometric Brownian Motion process with $\mu = 2$, $\sigma = 1$.

1.4.3 Ornstein-Uhlenbeck Process

We now look at an example of a mean-reverting process, where the process is "pulled" towards a specific value.

Definition 1.9. *The Ornstein-Uhlenbeck Process has the SDE form*

$$dx_t = \theta(\mu - x_t)dt + \sigma dW(t),$$

where $\theta > 0$, μ and $\sigma > 0$ are taken to be parameters, and $W(t)$ denotes a Brownian Motion process.

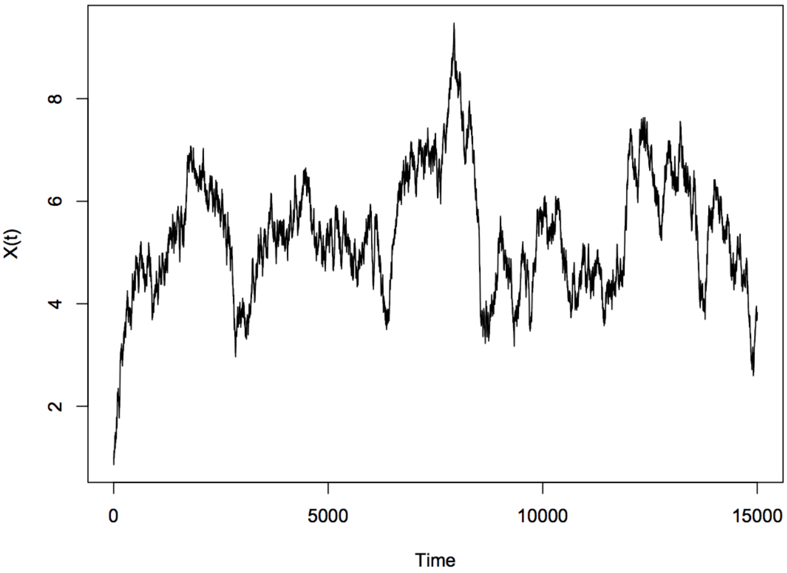

Figure 3: Sample path of an Ornstein-Uhlenbeck process with $\mu = 5$, $\theta = 4$, $\sigma^2 = 2$.

2 Spatial Estimation of Parameters

We now turn to estimation of the drift and diffusion parameters of a diffusion process. The first approach considered was that of the crossing tree.

2.1 The Crossing Tree

Jones and Shen (2004) introduced the notion of the crossing tree. Imagine a lattice of the form

$$\delta_0 + \delta\mathbb{Z}.$$

Now, imagine that we observe a continuous stochastic process on this lattice. We define a crossing as

Definition 2.1. *A crossing is defined as* $T_{i+1} = inf\,\{t > T_i : |X(t) - X(T_i)| \geq \delta\}$.

Heuristically, a crossing is simply the time at which the process crosses a lattice line on either side of the previous crossing point, however if it crosses the same line as the previous crossing point, it is not considered another crossing. An example of this is given in Figure 4.

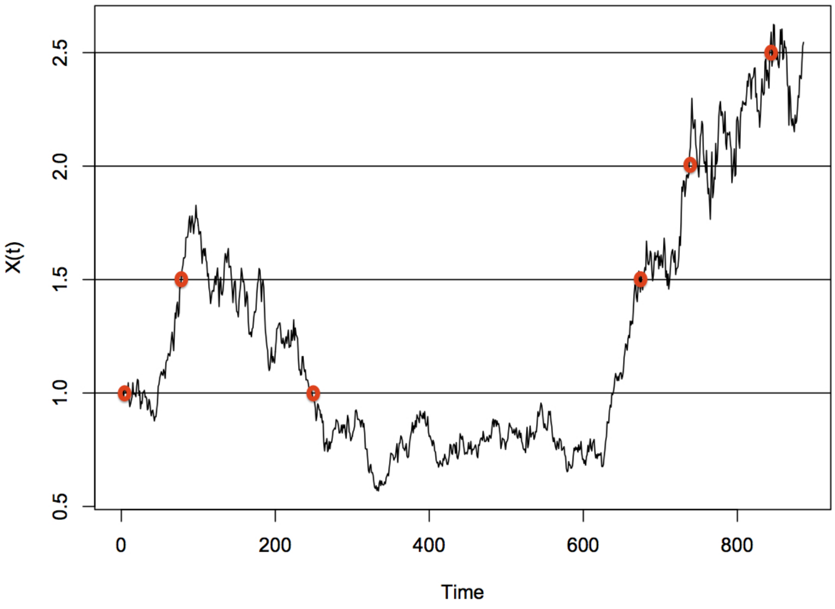

Figure 4: Crossings of a Geometric Brownian Motion process.

2.2 Scale Measure and Scale Function

An important definition is that of the scale measure, assuming the process is time homogeneous. Its use will become clear through the derivation of the hitting probabilities in the next section.

Definition 2.2.

The scale measure is defined as

$$s(x) = exp^{-\int^{x} \frac{2\mu(\epsilon)}{\sigma^2(\epsilon)} d\epsilon}.$$

The scale function is the indefinite integral of this scale measure,

$$S(x) = \int^x s(\nu)d\nu$$

(Karlin and Taylor, 1981)

2.3 Derivation of Hitting Probabilities

A stochastic process has a certain probability of hitting the upper boundary of an interval before the lower boundary. The crossing tree method uses this information, along with the observed number of up-crossings, Q, and total number of crossings N to estimate the probability of hitting the upper boundary first. This information is then used to estimate the parameters of the process. This probability is defined as follows,

Definition 2.3. *Define $T(a)$ as the first time that the process reaches the value a. This is known as the "hitting time" of a.*

To find the hitting probability of the upper boundary before the lower boundary, we must compute

$$u(x) = P\{T(b) < T(a)|X(0) = x\},\ a < x < b.$$

Karlin and Taylor (1981) derive a differential equation governing this probability. Let $a < x < b$, and choose a h such that the probability of attaining either of the boundaries in the interval $(t, t+h)$ is negligible. Now, if we condition on the location of the process at time h, $X(h)$, then the probability of hitting the upper boundary before the lower boundary at time h is $u(X(h))$. Using the law of total expectation, we find

$$u(x) = E[u(X(h))|X(0) = x] + o(h).$$

Now write $X(h) = \Delta X + x$, and assume that we can expand $u(\Delta X + x)$ in a Taylor Series around $\Delta X = 0$,

$$u(x) = E[u(x) + \Delta X u'(x) + \frac{1}{2}(\Delta X)^2 u''(x) + o(h)|X(0) = x] + o(h).$$

Now, it is known that the infinitesimal parameters of the process are $\lim_{h\to 0}\frac{1}{h}E[\Delta X|X(0) = x] = \mu(x)$, $\lim_{h\to 0}\frac{1}{h}E[(\Delta X)^2|X(0) = x] = \sigma^2(x)$, so

$$\begin{aligned} u(x) &= E[u(x)|X(0) = x] + u'(x)E[\Delta X|X(0) = x] + \frac{1}{2}u''(x)E[(\Delta X)^2|X(0) = x], \\ u(x) &= u(x) + \mu(x)h\frac{du}{dx} + \frac{1}{2}\sigma^2(x)h\frac{d^2u}{dx^2} + o(h). \end{aligned}$$

Now, dividing by h and letting $h \to 0$, we find,

$$0 = \mu(x)\frac{du}{dx} + \frac{1}{2}\sigma^2(x)\frac{d^2u}{dx^2}.$$

The boundary conditions are $u(a) = 0$ and $u(b) = 1$ for obvious reasons.

2.3.1 Solving the differential equation

Karlin and Taylor (1981) solve this equation to yield the hitting probability of the upper boundary in terms of the Scale function,

$$u(x) = \frac{S(x) - S(a)}{S(b) - S(a)}.$$

2.4 Concrete examples of hitting probabilities

2.4.1 Brownian Motion With Drift

The Scale function for a Brownian Motion process $X = \{X(t), t \geq 0\}$ with drift μ and variance σ^2 is

$$S(x) = e^{-2x\mu/\sigma^2}.$$

On our spatial lattice, with boundaries δ apart, the probability of hitting the upper boundary before the lower boundary, given that you start exactly between the two boundaries is

$$u(x) = \frac{S(x) - S(a)}{S(b) - S(a)},$$

we let $\delta = a - x = x - b$,

$$
\begin{aligned}
u(x) &= \frac{e^{-2\frac{x\mu}{\sigma^2}} - e^{-2\frac{(x-\delta)\mu}{\sigma^2}}}{e^{-2\frac{(x+\delta)\mu}{\sigma^2}} - e^{-2\frac{(x-\delta)\mu}{\sigma^2}}}, \\
&= \frac{1 - e^{2\frac{\delta\mu}{\sigma^2}}}{e^{-2\frac{\delta\mu}{\sigma^2}} - e^{2\frac{\delta\mu}{\sigma^2}}}.
\end{aligned}
$$

Now, let us substitute $\rho = \frac{\mu}{\sigma^2}$,

$$
u(x) = \frac{1 - e^{2\delta\rho}}{e^{-2\delta\rho} - e^{2\delta\rho}}.
$$

If we now multiply the fraction by $\frac{e^{-2\delta\rho}}{e^{-2\delta\rho}}$,

$$
\begin{aligned}
u(x) &= \frac{e^{-2\delta\rho} - 1}{e^{-4\delta\rho} - 1}, \\
&= \frac{e^{-2\delta\rho} - 1}{(e^{-2\delta\rho} + 1)(e^{-2\delta\rho} - 1)}, \\
&= \frac{1}{e^{-2\delta\rho} + 1}
\end{aligned}
$$

2.4.2 Geometric Brownian Motion

The scale measure for Geometric Brownian Motion is given by

$$
s_Y(y) = e^{-2\int^y \frac{(\mu+\frac{1}{2}\sigma^2)\xi}{\sigma^2\xi^2}d\xi},
$$

letting $\rho = \mu/\sigma^2$,

$$
\begin{aligned}
&= e^{-\int^y (2\rho+1)\frac{1}{\xi}d\xi}, \\
&= e^{-\log(y)(2\rho+1)}, \\
&= (\frac{1}{y})^{2\rho+1}.
\end{aligned}
$$

The scale function is accordingly given by

$$
\begin{aligned}
S_Y(y) &= \int^y s(\xi)d\xi \\
&= \frac{-1}{2\rho}(\frac{1}{y})^{2\rho}.
\end{aligned}
$$

The hitting probability can now be easily found,

$$
\begin{aligned}
u(x) &= \frac{S(x) - S(x-\delta)}{S(x+\delta) - S(x-\delta)}, \\
&= \frac{x^{-2\rho} - (x-\delta)^{-2\rho}}{(x+\delta)^{-2\rho} - (x-\delta)^{-2\rho}}
\end{aligned}
$$

2.4.3 Ornstein-Uhlenbeck Process

Firstly, the error function is introduced,

Definition 2.4. *The error function is defined as*

$$
erf(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt.
$$

The scale measure for the Ornstein-Uhlenbeck process has the form

$$
\begin{aligned}
s(x) &= e^{-\int^x \frac{2\mu(\epsilon)}{\sigma^2(\epsilon)} d\epsilon}, \\
s(x) &= e^{-\frac{\theta x}{\sigma^2}(2\mu - x)}.
\end{aligned}
$$

Integrating this gives the scale function,

$$
\begin{aligned}
S(x) &= \int^x e^{-\frac{\theta t}{\sigma^2}(2\mu - t)} dt, \\
S(x) &= \frac{\sqrt{\pi}\sigma e^{-\frac{\mu^2\theta}{\sigma^2}} \mathrm{erf}(\frac{\sqrt{\theta}(x-\mu)}{\sigma})}{2\sqrt{\theta}} + c,
\end{aligned}
$$

where erf represents the error function.

Now, the hitting probabilities of the upper boundary have the form

$$
\begin{aligned}
u(x) &= \frac{S(x) - S(x-\delta)}{S(x+\delta) - S(x-\delta)}, \\
&= \frac{\mathrm{erf}(\frac{\sqrt{\theta}(x-\mu)}{\sigma}) - \mathrm{erf}(\frac{\sqrt{\theta}(x-\delta-\mu)}{\sigma})}{\mathrm{erf}(\frac{\sqrt{\theta}(x+\delta-\mu)}{\sigma}) - \mathrm{erf}(\frac{\sqrt{\theta}(x-\delta-\mu)}{\sigma})}
\end{aligned}
$$

2.5 Maximum Likelihood Estimation of Parameters

2.5.1 Brownian Motion with Drift

Brownian motion with drift has a simple, closed form expression for the maximum likelihood estimate, which can be obtained by maximising the likelihood function analytically.

Theorem 2.1. *Let $X(t)$ be a Brownian Motion with drift process, with unknown drift μ and diffusion σ^2. Let a crossing tree of width δ be given, and let Q be the number of up-crossings, and N be the total number of crossings. Then the maximum likelihood estimator for $\rho = \frac{\mu}{\sigma^2}$ is*

$$\hat{\rho} = \frac{1}{2\delta} \log\left(\frac{Q}{N-Q}\right).$$

Proof.

$$\begin{aligned}
L(\rho) &= \prod_{i=1}^{Q} \frac{1}{e^{-2\delta\rho}+1} \prod_{j=1}^{N-Q} (1 - \frac{1}{e^{-2\delta\rho}+1}), \\
\log(L(\rho)) &= \sum_{i=1}^{Q} \log(\frac{1}{e^{-2\delta\rho}+1}) + \sum_{j=1}^{N-Q} \log((\frac{e^{-2\delta\rho}}{e^{-2\delta\rho}+1})), \\
\frac{dL(\rho)}{d\rho} &= \sum_{i=1}^{N} \frac{2\delta e^{-2\delta\rho}}{e^{-2\delta\rho}+1} - 2\delta, \\
0 &= N\frac{2\delta e^{-2\delta\rho}}{e^{-2\delta\rho}+1} - (N-Q)2\delta, \\
\frac{N-Q}{N}(e^{-2\delta\rho}+1) &= e^{-2\delta\rho}, \\
\frac{N-Q}{N}(1+e^{2\delta\rho}) &= 1, \\
e^{2\delta\rho} &= \frac{Q}{N-Q}, \\
2\delta\rho &= \log\left(\frac{Q}{N-Q}\right), \\
\hat{\rho} &= \frac{1}{2\delta}\log\left(\frac{Q}{N-Q}\right).
\end{aligned}$$

□

2.5.2 Geometric Brownian Motion

A closed-form likelihood function does not exist for the Geometric Brownian Motion and the Ornstein-Uhlenbeck process, as the hitting probabilities depend upon the path taken, however, for a given sample, we can construct a likelihood function that can be numerically optimised.
Let x_k be the kth crossing point, and let $\kappa = 2\rho$.

$$\begin{aligned}\log(L(p)) = \sum_{k=1}^{N} (&\mathbf{1}(x_k > x_{k-1}) \log(x_{k-1}^{-\kappa} - (x_{k-1} - \delta)^{-\kappa}) + \mathbf{1}(x_k < x_{k-1}) \log((x_{k-1} + \delta)^{-\kappa} - x_{k-1}^{-\kappa}) \\ &- \log((x_{k-1} + \delta)^{-\kappa} - (x_{k-1} - \delta)^{-\kappa}))\end{aligned}$$

and we now optimise this numerically to obtain estimates of μ and σ.

2.5.3 Ornstein-Uhlenbeck Process

We take the following log-likelihood function, where $u(x)$ is defined in section 2.4.3,

$$\log(L(\mu, \theta, \sigma)) = \sum_{k=1}^{N} (\mathbf{1}(x_k > x_{k-1}) \log(u(x_k)) + \mathbf{1}(x_k < x_{k-1}) \log(1 - u(x_k))$$

and we now optimise this numerically to obtain estimates of μ, θ and σ.

3 Simulation

3.1 Spatial Simulation

The processes were simulated on a spatial lattice by using the closed form expressions for the hitting probabilities of the upper boundary of the interval. This amounted to simulating a series of biased coin tosses, with a probability u of stepping up, and $(1 - u)$ of stepping down.

3.2 Brownian Motion and Geometric Brownian Motion at regular points in time

Brownian Motion with Drift was a simple process to simulate, as the distribution of $B(t+\Delta t) - B(t)$ is known to be $N(\mu\Delta t, \sigma^2\Delta t)$. A standard normal random variable was simulated, and then scaled,

$$B(t+\Delta t) - B(t) = \mu\Delta t + \sigma\sqrt{\Delta t}N(0,1).$$

The sample path is then calculated by a cumulative sum.
Geometric Brownian Motion is easily obtained from this process, by taking $\exp\{B(t)\}$, where $B(t)$ is a Brownian Motion with Drift process.

3.3 Ornstein-Uhlenbeck Process at regular points in time

Glasserman (2003) shows that a sample from an Ornstein-Uhlenbeck process at evenly spaced periods of time is an AR(1) process. The following equation can be used to simulate the OU process for any time step Δt,

$$S_{t+1} = e^{-\theta\Delta t}S_t + (1 - e^{-\theta\Delta t})\mu + \sigma\sqrt{\frac{(1 - e^{-2\theta\Delta t})}{2\theta}}Z,$$

where $Z \sim N(0,1)$.

4 Results for spatial estimators

4.1 Brownian Motion with Drift Spatial Estimators

The Brownian Motion with Drift estimator performed well, with the maximum likelihood estimator confidence interval for the expected value of the estimator including the true value, as seen in Figure 5.
The parameters used were:

- $T = 50$
- $\mu = -1$
- $\sigma^2 = 1$

- $\epsilon = 0.0011$
- $\delta = 0.1$

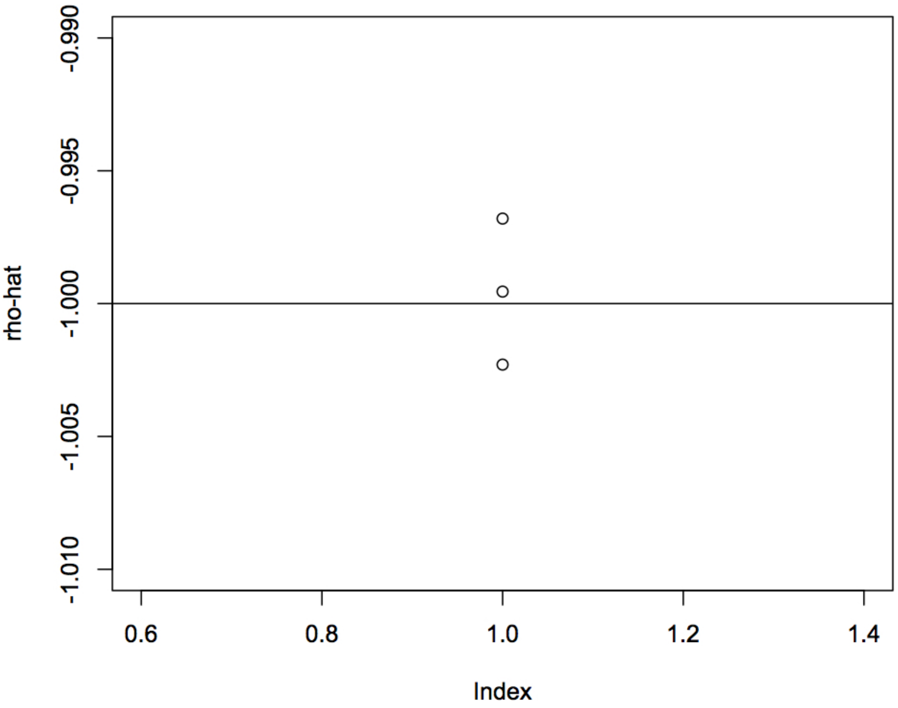

Figure 5: Confidence Interval for ρ with true value marked. 25 observations.

4.2 Geometric Brownian Motion Spatial Estimators

The Geometric Brownian Motion estimator performed well, with the maximum likelihood estimator confidence interval including the true value, as seen in Figure 6.
The parameters used were:

- $T = 5$
- $\mu = -10$
- $\sigma^2 = 3$
- $\epsilon = 0.001$

- $\delta = 0.1$

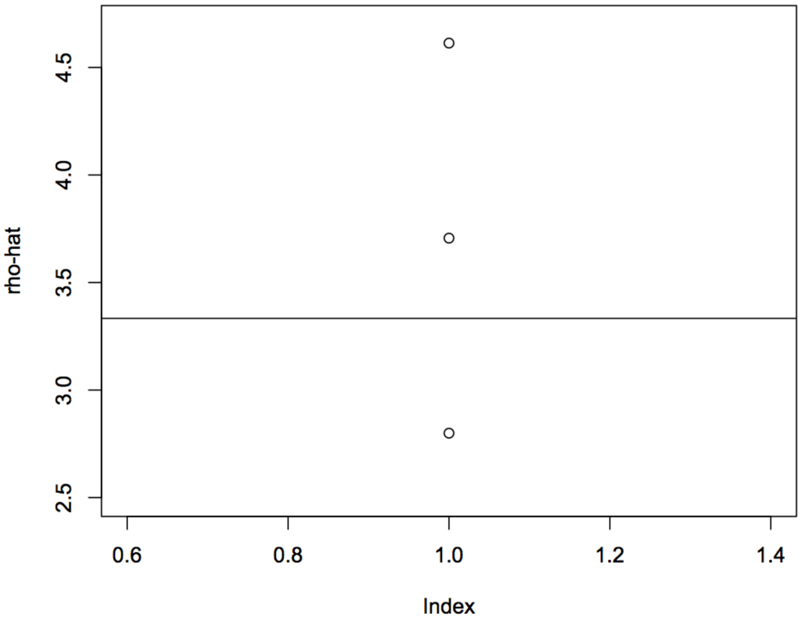

Figure 6: Confidence Interval for ρ with true value marked. 25 observations.

4.3 Ornstein-Uhlenbeck Spatial Estimators

Three parameters were estimated for the Ornstein-Uhlenbeck process, μ, σ and θ. All three estimators performed well, with the true value falling within a 95% confidence interval, as seen in the following figures. The parameters used were:

- $T = 5$
- $\mu = 5$
- $\sigma^2 = 1$
- $\theta = 3$
- $\delta = 1$
- reps $= 20$

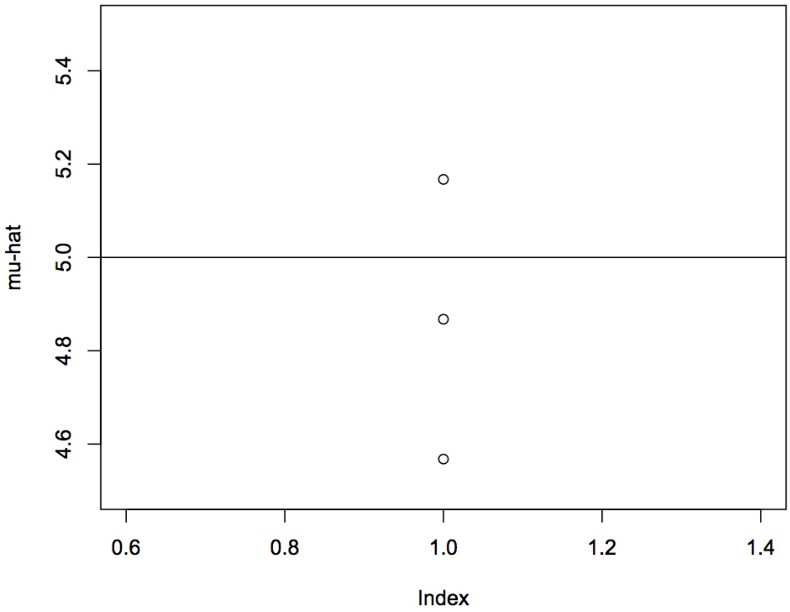

Figure 7: Confidence Interval for μ with true value marked. 18 observations.

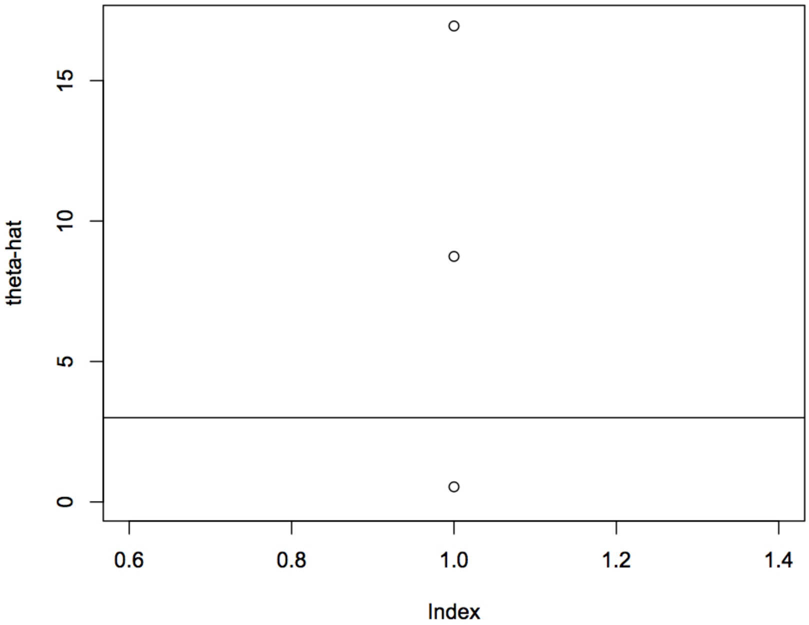

Figure 8: Confidence Interval for θ with true value marked. 18 observations.

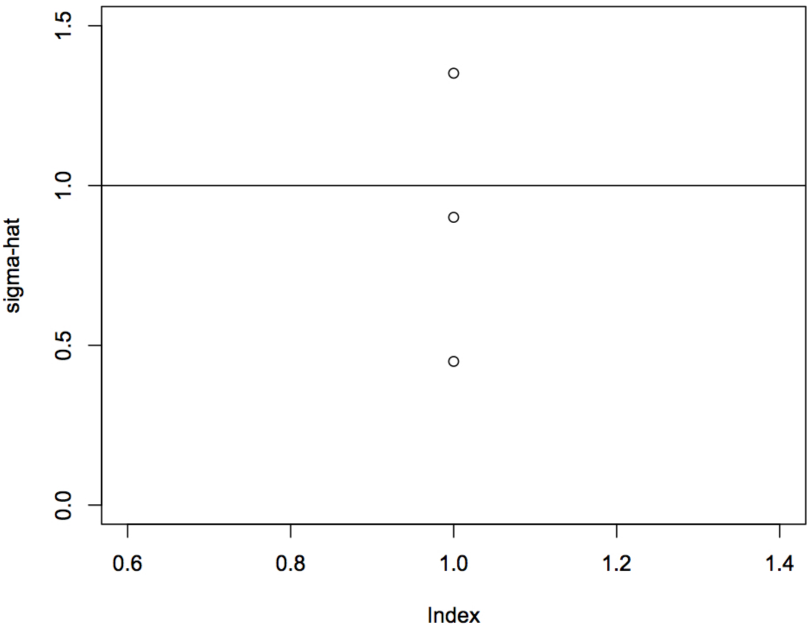

Figure 9: Confidence Interval for σ with true value marked. 18 observations.

5 Monte Carlo Testing of Spatial Estimators

We tested the spatial estimators in a point-in-time setting, by simulating a process at regularly spaced points in time, and then applying the crossing-tree methodology. We tried this across several different δ values.The results were generally disappointing, and it was difficult to generate many repetitions as this process was very computationally demanding, with simulations requiring hours to run. This was due to the fact that we needed to generate thousands of process values, and then numerically optimise a likelihood function.

5.1 Ornstein-Uhlenbeck Process

The results and confidence intervals for the Ornstein-Uhlenbeck process are shown in the next figures. Due to computational demands, only 20 observations of the estimators were able to be calculated. This calls into question whether the Central Limit Theorem applies for the confidence intervals, and with more observations this would become more reliable.

The parameters used were:

- $T = 5$
- $\theta = 2$
- $\mu = 5$
- $\sigma^2 = 2$
- $\delta = 0.75$
- $\epsilon = 0.0088$
- reps $= 20$

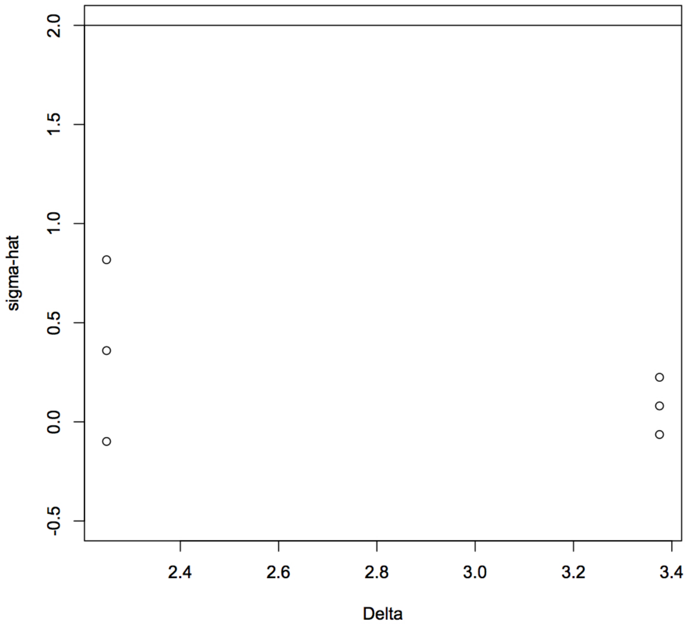

Figure 10: Confidence Interval for σ with true value marked. 20 observations.

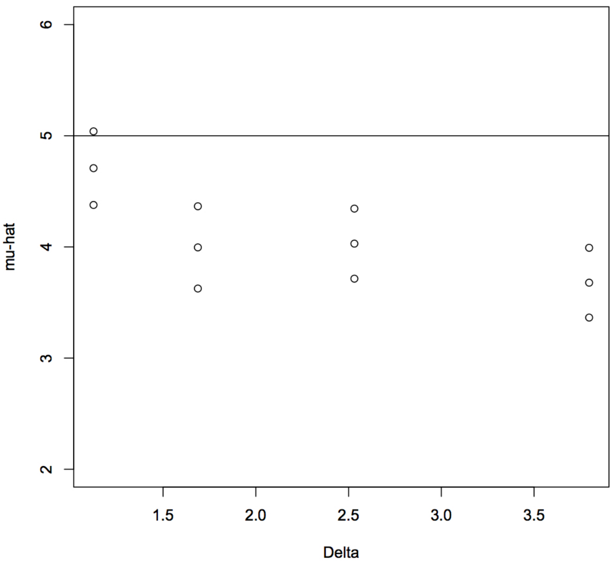

Figure 11: Confidence Interval for μ with true value marked. 20 observations.

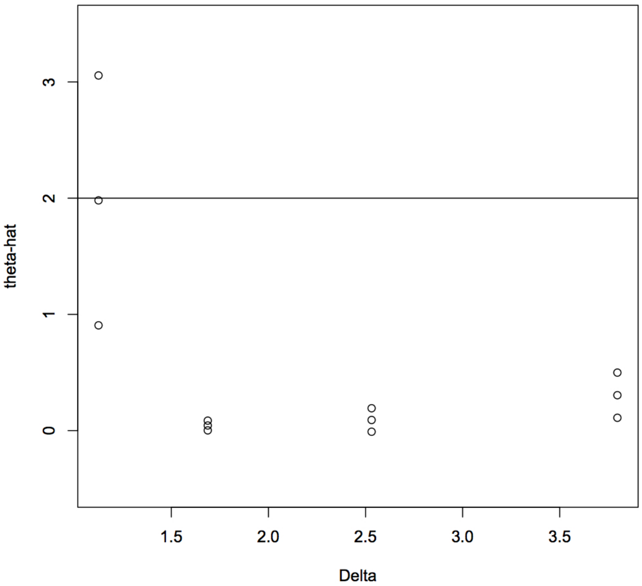

Figure 12: Confidence Interval for θ with true value marked. 20 observations.

A feature of these plots is the widening confidence intervals as delta increases, non-withstanding the left-most delta. This widening occurs because of the fact there are fewer crossings for larger delta, and so the estimator becomes more variable. The smallest delta has a large confidence interval because of a large standard deviation, which is the case because there may be crossings that it is missing due to the sampling at regular points in time. This creates more variability, and the probability of missing a crossing decreases significantly as delta increases.

6 Estimation when the process is observed at regular points in time

If the process is not observed spatially, but instead at regularly spaced points in time, there are estimators readily available.

6.1 Brownian Motion with Drift

The regular point in time maximum likelihood estimator is introduced through the next theorem,

Theorem 6.1. *Let $Z(t_i)$ be a Brownian Motion process with drift μ and volatility σ^2, sampled at constant time intervals $t_i - t_{i-1} = \Delta t$, and let $R(t)$ be defined as $R(t_i) = Z(t_i) - Z(t_{i-1})$. Let $r = (r_1, r_2, ..., r_n)$ be a sample from $R(t)$, then the maximum likelihood estimator for μ is $\hat{\mu} = \frac{Z(t_n)-Z(t_0)}{n\Delta t}$. Furthermore, the maximum likelihood estimate for σ^2 is $\hat{\sigma^2} = \frac{\frac{n-1}{n}s_r^2}{\Delta t}$, where s_r^2 denotes the sample variance of the sample from R.*

Proof.

Due to the independence of increments,

$$R(t) \sim N(\mu\Delta t, \sigma^2\Delta t),$$

so the likelihood function is

$$\begin{aligned} L(r, \mu, \sigma^2) &= \prod_{i=1}^{n} f(r_i|\mu, \sigma^2), \\ &= (\frac{1}{\sigma^2 2\pi\Delta t})^{\frac{n}{2}} exp^{\frac{\sum_{i=1}^{n}(r_i-\mu\Delta t)^2}{2\sigma^2\Delta t}}. \end{aligned}$$

Taking the logarithm yields

$$\log(L(r, \mu\sigma^2) = -\frac{n}{2}\log(2\sigma^2\pi\Delta t) + \frac{\sum_{i=1}^{n}(r_i - \mu\Delta t)^2}{2\sigma^2\Delta t},$$

now differentiating with respect to μ,

$$\frac{\partial \log(L)}{\partial \mu} = -\sum_{i=1}^{n}(r_i - \mu\Delta t),$$

and we let the left-hand side equal zero to find the maximum, so

$$\begin{aligned} 0 &= n\bar{r} - n\hat{\mu}\Delta t, \\ \hat{\mu} &= \frac{\bar{r}}{\Delta t}. \end{aligned}$$

Due to the definition of $R(t)$, this is a telescoping sum that cancels to leave

$$\hat{\mu} = \frac{Z(t_n) - Z(0)}{t_n}.$$

Now, differentiating with respect to σ^2, and letting the derivative equal 0,

$$\begin{aligned}\frac{\partial \log(L)}{\partial \sigma^2} &= -\frac{n}{2\sigma^2} + \frac{\sum_{i=1}^{n}(r_i - \mu\Delta t)^2}{2\sigma^4 \Delta t} = 0,\\ 0 &= -\frac{n\sigma^2}{2} + + \frac{\sum_{i=1}^{n}(r_i - \mu\Delta t)^2}{2\Delta t},\end{aligned}$$

rearranging this and substituting the expression for $\hat{\mu}$ yields

$$\hat{\sigma^2} = \frac{\sum_{i=1}^{n}(r_i - \bar{r})^2}{n\Delta t}.$$

The summation can be recognised as part of the sample variance of the sample r. Therefore,

$$\hat{\sigma^2} = \frac{(n-1)s_r^2}{n\Delta t}.$$

□

6.2 Geometric Brownian Motion

An analogous result for Geometric Brownian Motion follows,

Theorem 6.2. *Let $X(t)$ be a Brownian Motion process with drift μ and volatility σ^2, and let $G(t) = e^{X(t)}$ be a Geometric Brownian Motion process. Define $R(t_i) = \frac{G(t_i)}{G(t_{i-1})}$ so that $\log(R(t_i)) = X(t_i) - X(t_{i-1}) \sim N(\mu\Delta t, \sigma^2\Delta t)$. This means that $R(t_i) \sim LN(\mu\Delta t, \sigma^2\Delta t)$, where LN stands for the log-normal distribution. Then the maximum likelihood estimates for the parameters μ and σ^2 are*

$$\begin{aligned}\hat{\mu} &= \frac{\log(\frac{G(t_n)}{G(t_0)})}{n\Delta t},\\ \hat{\sigma^2} &= \frac{\sum_{i=1}^{n}(\log(r_i) - \frac{\sum_{i=1}^{n}\log(r_i)}{n})^2}{n\Delta t}.\end{aligned}$$

Proof.

The log-normal distribution has the density

$$f(r;\mu,\sigma^2)=\frac{1}{r\sqrt{2\pi\sigma^2}}exp-\frac{(\log(r)-\mu)^2}{2\sigma^2}.$$

Taking the Likelihood function yields

$$\begin{aligned}
L(r,\mu,\sigma^2) &= (\frac{1}{2\sigma^2\pi\Delta t})^{\frac{n}{2}}\prod_{i=1}^{n}\frac{1}{r_i}exp^{-\frac{\sum_{i=1}^{n}(\log(r_i)-\mu\Delta t)^2}{2\sigma^2\Delta t}},\\
\log(L) &= -\frac{n}{2}\log(2\sigma^2\pi\Delta t)-\sum_{i=1}^{n}\log(r_i)-\frac{\sum_{i=1}^{n}(\log(r_i)-\mu\Delta t)^2}{2\sigma^2\Delta t}\\
\frac{\partial\log(L)}{\partial\mu} &= \sum_{i=1}^{n}(\log(r_i)-\mu\Delta t)=0,\\
\hat{\mu} &= \frac{\sum_{i=1}^{n}\log(r_i)}{n\Delta t},
\end{aligned}$$

because of the definition of $R(t_i)$, this is telescoping sum that yields

$$\hat{\mu}=\frac{\log(\frac{G(t_n)}{G(t_0)})}{n\Delta t}.$$

Now, turning our attention to $\hat{\sigma^2}$,

$$\begin{aligned}
\frac{\partial\log(L)}{\partial\sigma^2} &= -\frac{n}{2\sigma^2}+\frac{\sum_{i=1}^{n}(\log(r_i)-\mu\Delta t)^2}{2\sigma^4\Delta t}=0,\\
\hat{\sigma^2} &= \frac{\sum_{i=1}^{n}(\log(r_i)-\mu\Delta t)^2}{n\Delta t},\\
\hat{\sigma^2} &= \frac{\sum_{i=1}^{n}(\log(r_i)-\frac{\sum_{i=1}^{n}\log(r_i)}{n})^2}{n\Delta t}.
\end{aligned}$$

□

6.3 Least-Squares Estimation of Ornstein-Uhlenbeck Process

Maximum likelihood estimation for the Ornstein-Uhlenbeck Process is very difficult, with the algebra becoming very messy. For this reason, the least-squares approach was chosen, and estimators derived using the fact established in section 3.3 that a sample at regularly spaced points in time from an Ornstein-Uhlenbeck Process is an autoregressive process of order 1.

Theorem 6.3. *Let S_t be a sample from an Ornstein-Uhlenbeck process, sampled at regularly spaced points in time of length Δt. Then the least-squares estimates of the parameters of the process are $\hat{\mu} = \frac{\hat{a}}{1-\hat{b}}$, $\hat{\theta} = -\frac{\log(\hat{b})}{\Delta t}$ and $\hat{\sigma^2} = \frac{2\log(\hat{b})se(\epsilon)^2}{\Delta t(\hat{b}^2-1)}$, where $\hat{a}$ and $\hat{b}$ are the ordinary least-squares regression estimators.*

Proof.

An AR(1) process has the form

$$S_{t+1} = c + \varphi S_t + \epsilon_t,$$

we can find estimates for the parameters by regressing S_{t+1} against S_t. From the equation above,

$$S_{t+1} = e^{-\theta\Delta t}S_t + (1 - e^{-\theta\Delta t})\mu + \sigma\sqrt{\frac{(1-e^{-2\theta\Delta t})}{2\theta}}Z,$$

matching this with the usual regression form

$$S_{t+1} = a + bS_t + \epsilon_t,$$

where $\epsilon_t \sim N(0, \sigma_N^2)$, we find the following least-squares estimates for the parameters,

$$\begin{aligned}
b &= e^{-\theta\Delta t},\\
\hat{\theta} &= -\frac{\log(\hat{b})}{\Delta t},\\
a &= (1 - e^{-\theta\Delta t})\mu,\\
\hat{\mu} &= \frac{\hat{a}}{(1 - e^{-\hat{\theta}\Delta t})},
\end{aligned}$$

substituting $\hat{\theta}$ yields

$$\hat{\mu} = \frac{\hat{a}}{1 - \hat{b}}.$$

The estimate for σ^2 is as follows,

$$\begin{aligned} se(\epsilon) &= \sigma \sqrt{\frac{(1 - e^{-2\theta \Delta t})}{2\theta}}, \\ \hat{\sigma^2} &= \frac{2\hat{\theta} se(\epsilon)^2}{(1 - e^{2\hat{\theta}\Delta t})}, \end{aligned}$$

substituting in $\hat{\theta}$,

$$\begin{aligned} &= -\frac{2\log(\hat{b}) se(\epsilon)^2}{(1 - \hat{b}^2)\Delta t}, \\ \hat{\sigma^2} &= \frac{2\log(\hat{b}) se(\epsilon)^2}{\Delta t(\hat{b}^2 - 1)} \end{aligned}$$

□

6.3.1 Results for Brownian Motion with Drift

The results for the Brownian Motion with Drift was very good, with the true values of the parameters falling within the 95% confidence interval around the maximum likelihood estimates.
The parameters used were:

- $T = 10$
- $\mu = 2$
- $\sigma^2 = 1$
- $\epsilon = 0.0039$

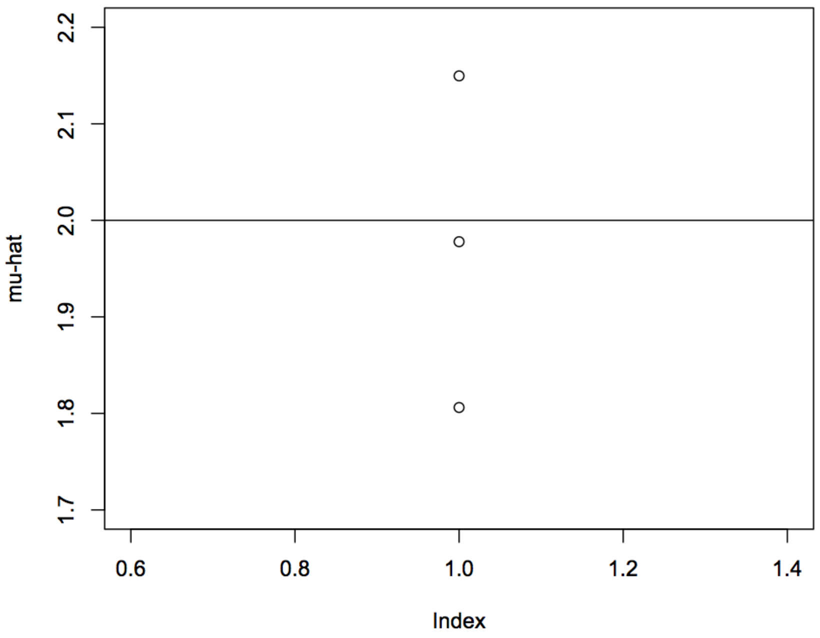

Figure 13: Confidence interval for μ, with the true value marked by a horizontal line.

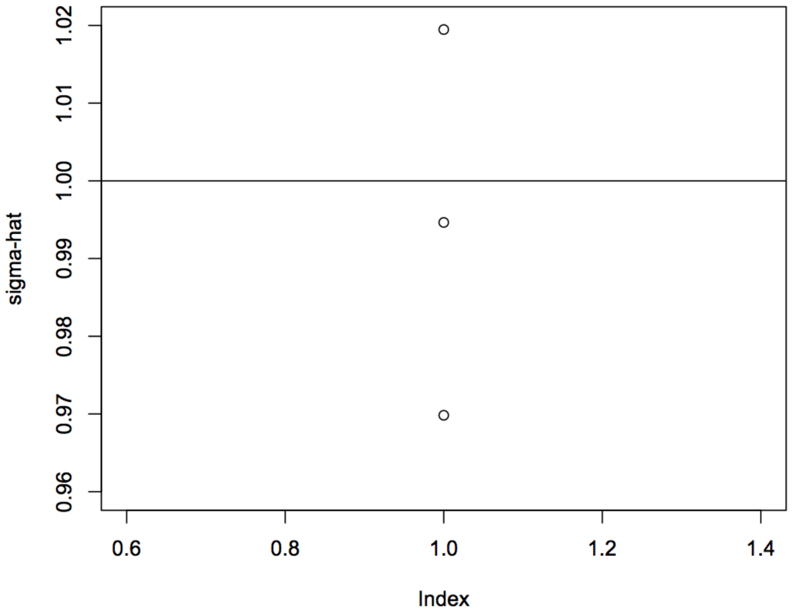

Figure 14: Confidence Interval for σ, with the true value marked by a horizontal line.

6.3.2 Results for Geometric Brownian Motion

The results for the Geometric Brownian Motion was very good, with the true values of the parameters falling within the 95% confidence interval around the maximum likelihood estimates.
The parameters used were:

- $T = 5$
- $\mu = -10$
- $\delta = 0.5$
- $\sigma^2 = 1$
- reps $= 20$

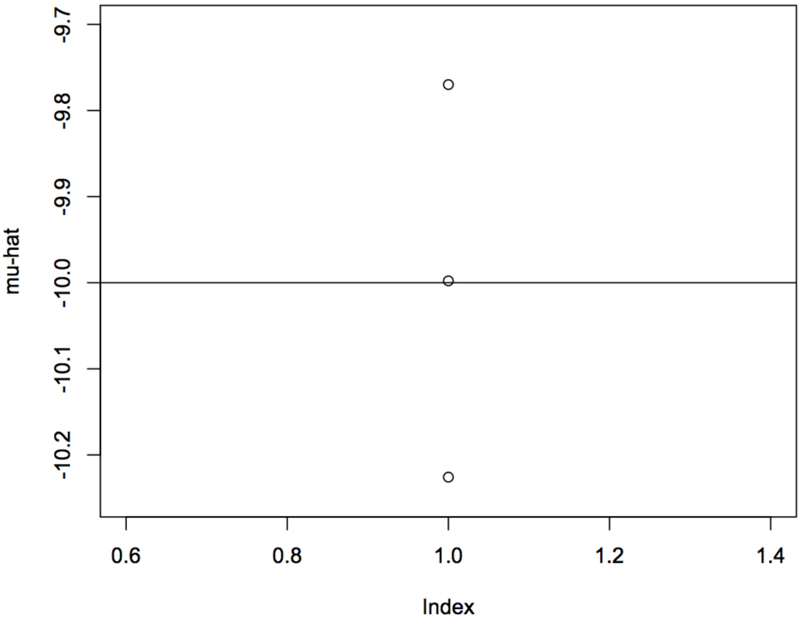

Figure 15: Confidence interval for μ, with the true value marked by a horizontal line.

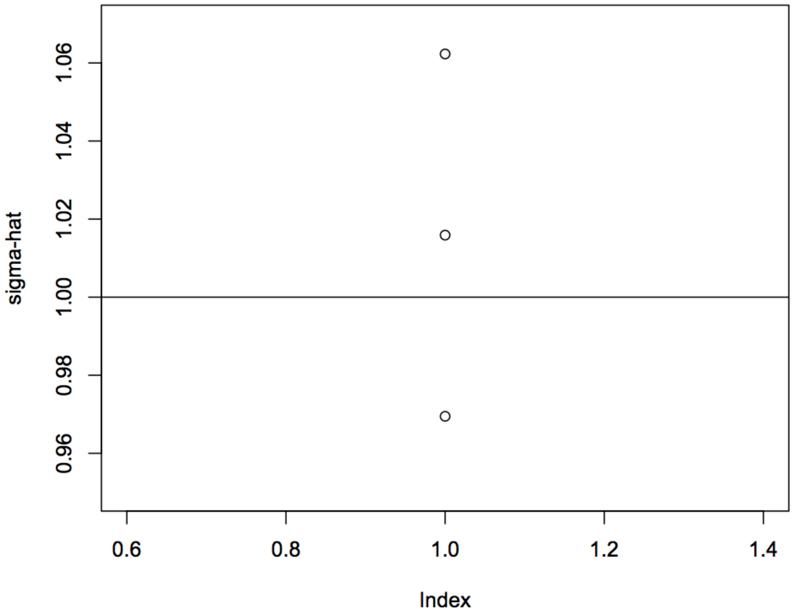

Figure 16: Confidence Interval for σ, with the true value marked by a horizontal line.

6.3.3 Results for Ornstein-Uhlenbeck Process

The results for the Ornstein-Uhlenbeck Process was very good, with the true values of the parameters falling within the 95% confidence interval around the maximum likelihood estimates.
The parameters used were:

- $T = 100$
- $\mu = 5$
- $\sigma^2 = 2$
- $\epsilon = 0.01$
- $\theta = 4$

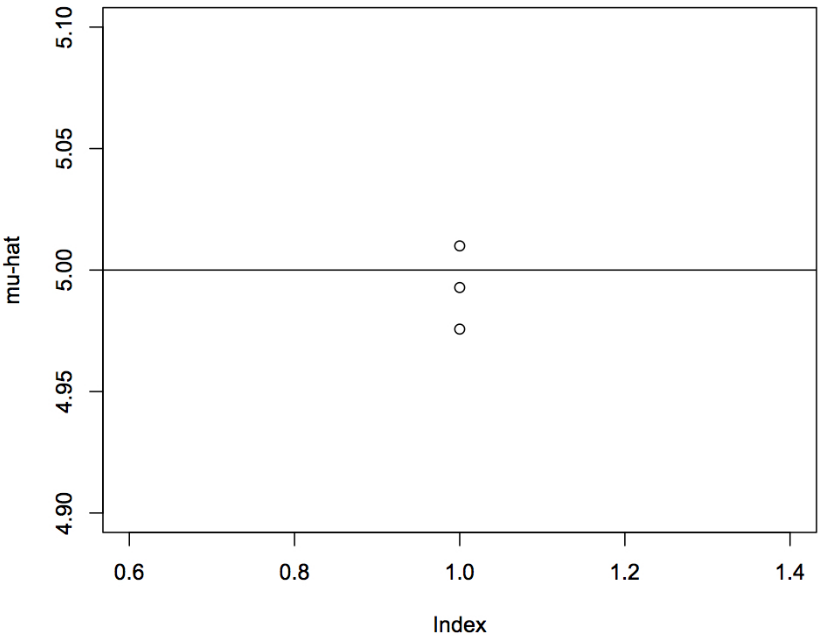

Figure 17: Confidence interval for μ, with the true value marked by a horizontal line.

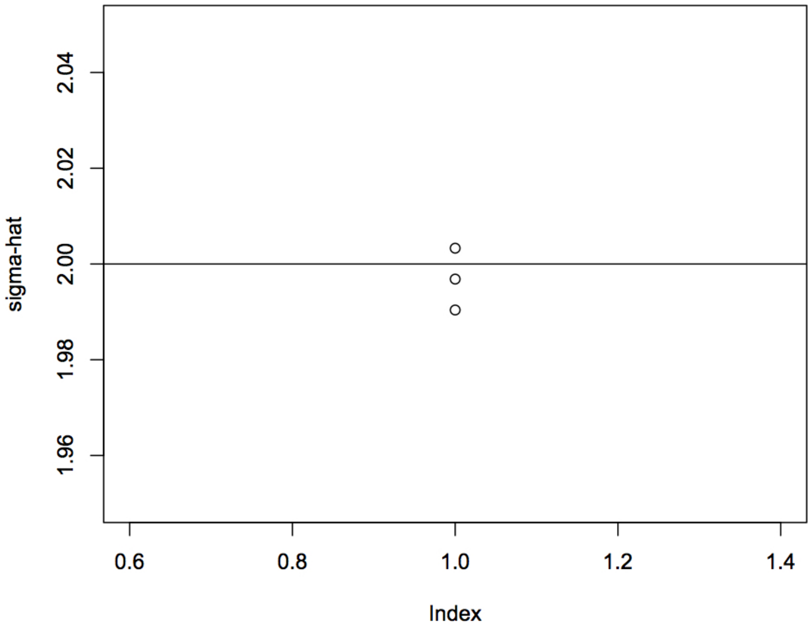

Figure 18: Confidence Interval for σ, with the true value marked by a horizontal line.

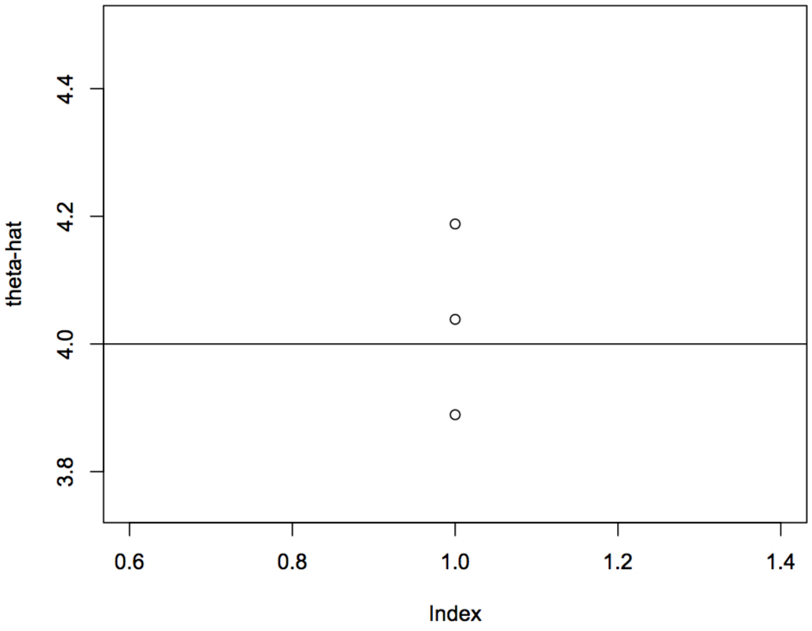

Figure 19: Confidence Interval for θ, with the true value marked by a horizontal line.

7 Martingale Theory

We now turn our attention to estimators that use information on the amount of time between crossings. Our main tool here is Doob's Optional Stopping Theorem.

7.1 Optional Stopping Theorem

Definition 7.1. *Let $X(t)$ be a diffusion process. Let T_{ab} be*

$$T_{ab} = inf\{t|X(t) - \mu t \notin [a,b]\},$$

then T_{ab} is called the first passage time of the process $X(t)$ of the interval $[a,b]$.

Theorem 7.1. *Let $M(t)$ be a martingale.*

1. *If $\tau \leq K < \infty$ is a bounded stopping time then $E[M(\tau)] = E[M(0)]$.*
2. *If $M(t)$ is uniformly integrable, then for any stopping time τ, $E[M(\tau)] = E[M(0)]$.*

7.2 Brownian Motion with Drift Expected Hitting Times

An estimator for μ in the case of Brownian Motion with Drift is derived. This approach works well in the case of Brownian Motion with Drift, but is very difficult to extend to Geometric Brownian Motion or the Ornstein-Uhlenbeck process.

Theorem 7.2. *Let $X(t)$ be a Brownian Motion process with drift μ and diffusion σ^2. Then $X(t) - \mu t$ is a martingale. Furthermore, the expected hitting time of the boundaries $[a, b]$ is*

$$E[T] = \frac{E[X(T)]}{\mu}$$

Proof.

We firstly prove that $X(t) - \mu t$ is a martingale,

$$\begin{aligned} E[X(t+1) - \mu(t+1)|\mathcal{F}_t] &= E[X(t+1)|\mathcal{F}_t] - \mu(t+1), \\ &= X(t) + \mu - \mu(t+1), \\ &= X(t) + \mu t. \end{aligned}$$

Therefore, $X(t) - \mu t$ is a martingale. Now, as $X(t) - \mu t$ is bounded between $[a, b]$, the martingale is uniformly integrable, justifying the use of the optional stopping theorem. This means

$$\begin{aligned} E[X(T) - \mu T] &= E[X(0) - \mu(0)], \\ E[X(T)] - \mu E[T] &= E[X(0)]. \end{aligned}$$

Now, $X(0) = 0$, so, rearranging,

$$E[T] = \frac{E[X(T)]}{\mu}.$$

□

8 Further Work

While we have estimators for both spatial, and point in time estimation of parameters, it would be excellent to be able to combine these into an estimator that used both sets of information concurrently. Generalised Method of Moments was identified as a possible technique for this purpose, however, due to time restraints, little progress was made in this regard. A further project could look at the suitability of Generalised Method of Moments, and derive necessary ”moment conditions” for the three processes featured in this report.

9 Acknowledgements

I would like to thank my supervisor, Dr. Owen Jones, for his guidance and patience. I would also like to thank AMSI and the University of Melbourne for this opportunity. I highly recommend this program to all prospective applicants.

References

Paul Glasserman. *Monte Carlo Methods in Financial Engineering.* Springer, 2003.

Geoffrey Grimmett and David Stirzaker. *Probability and Random Processes.* Oxford University Press, 2001.

O.D. Jones and Y. Shen. Estimating the hurst index of a self-similar process via the crossing tree. *Signal Process*, 2004.

Samuel Karlin and Howard M. Taylor. *A Second Course in Stochastic Processes.* Academic Press, Inc., 1981.

Fima Klebaner. *Introduction to Stochastic Calculus with Applications.* Imperial College Press, 2nd edition, 2005.

A R Code

A.1 Spatial Estimators

A.1.1 Brownian Motion with Drift

```
#This program uses a spatial approach to estimate
# the parameters of a Brownian Motion with drift process.

# Set Parameters
delta=2;
crossing = 0;
drift = -2;
diffusion = 2;
reps <- 25
p=drift/diffusion;

#Calculate Hitting Probability
f=1/(exp(-2*delta*p)+1);
estimators <- c(rep(0,reps))

for ( i in 1:reps) {
C=vector();
steps=50000;
for (j in 1:steps) {
#Coin tosses, will we step up or down?
r=runif(1);
if(r<f){
    C=append(C,1);
}
if(r>f) {
    C=append(C,0);
}
}
Q=0;
N=steps;
for (k in 1:length(C)) {
    if (C[k]==1) {
        Q=Q+1;
    }
}
Q
N
uhat=Q/N
#Estimator derived from the theory.
MLE=(1/(2*delta))*log(Q/(N-Q));
estimators[i] = MLE
}
```

A.1.2 Geometric Brownian Motion

```
#This program uses a spatial approach to estimate the parameters of a
# Geometric Brownian Motion process.
#Hitting probabilities
F<-function(x,p,d) {
return(((1/x)^(2*p)-(1/(x-d))^(2*p))/((1/(x+d))^(2*p)-(1/(x-d))^(2*p)))
}
likelihood=1

#Parameters
estimators=c(rep(0,reps))
d <- 0.2
drift <- 10
diffusion <- 3
p=drift/diffusion
start <- 5
x <- start
X <- vector()
C<- vector()
steps=5000
reps <- 20

estimators=c(rep(0,reps))
for (i in 1:reps) {
x <- start
X <- vector()
```

```
X <- append(X,x)
C<- vector()
p=drift/diffusion
for (j in 1:steps) {
   #Don’t allow process below 0.
   if(x-d==0 || x-d<0 || x==0 || x<2*d) {
    if(j!=1) {
        X<-append(X,x);
    }
    x<- x+d;
    C<- append(C,1);
    } else {
    #Coin tosses, up or down?
ra<-runif(1);
if(ra<F(x,p,d)) {
        C<-append(C,1)
     if(j!=1) {
        X<-append(X,x);
    }
    x<-x+d;
 } else {
  if(ra>F(x,p,d)) {
    C<-append(C,0)
     if(j!=1) {
        X<-append(X,x);
    }
    x<-x-d  }
    }
    }
    }
Q<-0;
N<-steps
#Add up all up-crossings
for (j in 1:length(C)) {
    if (C[j]==1) {
        Q<- Q+1;
    }
}

#The likelihood function that we maximise
likelihoodf <- function(par,C,d,X) {
likelihood = 0
for (j in 1:length(C)) {
if(C[j]==1) {
    x<-X[j];
    if(x==d || x-d<0 || x-d==0 || x-d<d) {
        likelihood=likelihood;
   } else {
    likelihood<-likelihood + log(F(x,par[1],d));
    }
} else {
if(C[j]==0){

    x<-X[j];
    if(x==d || x-d<0 || x-d==0 || x-d<d){
        likelihood<-likelihood;
   } else {
    likelihood<-likelihood + log((1-F(x,par[1],d)));
    }
    }
    }
    }
return(-likelihood)
}
# We use 'Brent' at R's suggestion.
MLE =
optim(par=c(1),likelihoodf,C=C,d=d,X=X,method=c("Brent"),lower=-100,upper=100)
estimators[i] = MLE$par[1]
}
```

A.1.3 Ornstein-Uhlenbeck Process

```
# This program uses a spatial approach to estimate the parameters of an
#Ornstein-Uhlenbeck Process.
```

```
# Complex error function
erfz    <- function(z)
{
    if (is.null(z)) return( NULL )
    else if (!is.numeric(z) && !is.complex(z))
        stop("Argument ’z’ must be a numeric or complex scalar or vector.")

    a0 <- abs(z)
    c0 <- exp(-z * z)
    z1 <- ifelse (Re(z) < 0, -z, z)

i <- a0 <= 5.8
work.i <- i
cer <- rep(NA, length = length(z))
    if ( sum(work.i) > 0) {
        cs <- z1
        cr <- cs
        for (k in 1:120) {
            cr[work.i] <- cr[work.i] * z1[work.i] * z1[work.i]/(k + 0.5)
            cs[work.i] <- cs[work.i] + cr[work.i]
            work.i <- work.i & (abs(cr/cs) >= 1e-15)
    if (sum(work.i) == 0) break
        }
        cer[i] <- 2 * c0[i] * cs[i]/sqrt(pi)
    }
work.i <- !i
    if( sum(work.i) > 0) {
        cl <- 1/z1
        cr <- cl
        for (k in 1:13) {
            cr[work.i] <- -cr[work.i] * (k - 0.5)/(z1[work.i] * z1[work.i])
            cl[work.i] <-  cl[work.i] + cr[work.i]
            work.i <- work.i & (abs(cr/cl) >= 1e-15)
    if (sum(work.i) == 0) break
        }
        cer[!i] <- 1 - c0[!i] * cl[!i]/sqrt(pi)
    }
cer[ Re(z) < 0] <- -cer[ Re(z) < 0]
    return(cer)
}


# Imaginary error function
erfi <- function(z) {
if(Re(z)!=0){-1i * erfz(1i * z)}
else {0}
}
#Calculate the hitting probability of the upper boundary.
hitting <- function(theta, sigma, mu, crossing, delta) {
return(Re((erfi((sqrt(abs(theta))/sigma)*(crossing-mu))
-erfi((sqrt(abs(theta))/sigma)*(crossing-delta-mu)))
/(erfi((sqrt(abs(theta))/sigma)*(crossing+delta-mu))
-erfi((sqrt(abs(theta))/sigma)*(crossing-delta-mu)))))
}
#Parameters
#Start point of the process.
crossing <- -2
mu <- 5
sigma <- 1
theta <- 3
delta <- 1
reps <- 20
estimators <- array(dim=c(3,reps))
for (k in 1:reps) {
#This will hold the up and down crossings.
C <- vector()
#This will count the number of up crossings.
Q = 0
#Set how many steps to generate.
reps <- 1000
#This will hold the crossings. This will be used with C to generate the
# likelihood function.
```

```
crossings <- vector()
for (i in 1:reps) {
crossings <- append(crossings,crossing)
u = hitting(theta,sigma,mu,crossing,delta)
if(runif(1)<Re(u)) {
C <- append(C,1)
crossing = crossing + delta
Q=Q+1
} else {
C <- append(C,0)
crossing = crossing - delta
}
}

likelihoodf <- function(C,crossings,par,delta,mu) {
likelihood <- 0
for(i in 1:length(C)) {
if(C[i]==1){
likelihood = likelihood + log(hitting(par[2],par[1],par[3],crossings[i],delta))
} else {
likelihood = likelihood+log((1-hitting(par[2],par[1],par[3],crossings[i],delta)))
}
}
return(-Re(likelihood))
}
#Optimise the resulting likelihood function with respect to the unknown parameters.
model<-optim(
par=c(runif(1,0.1,10),runif(1,0.1,3),runif(1,-5,5)),likelihoodf,C=C,
crossings=crossings,delta=delta)
estimators[1,k] = model$par[1]
estimators[2,k] = model$par[2]
estimators[3,k] = model$par[3]
}
```

A.2 Point in time estimators

A.2.1 Brownian Motion with Drift

```
#This program estimates parameters of brownian motion using
# point in time maximum likelihood estimators.

#Generates brownian motion process.
bmtime<-function(drift,diffusion,Time,delta) {
  eps = delta*sqrt(pi/(2*1024*diffusion))
  Y = c(0, cumsum(rnorm(ceiling(Time/eps), drift*eps, sqrt(diffusion*eps))))
  return(Y)
}
#Parameters
drift <- 2
diffusion <- 1
Time <- 10
reps <- 10
reps2 <- 20
deltavec<- c(0.1*1.5^(1:reps))


estimatorsmu <- array(dim=c(1,reps2))
estimatorssigma<-array(dim=c(1,reps2))
for(k in 1:reps2) {
eps = deltavec[1]*sqrt(pi/(2*1024*diffusion))
Z<-bmtime(drift,diffusion,Time,deltavec[1])
p=drift/diffusion;
R<- vector()
for (j in 1:(length(Z)-1)) {
R<-append(R,Z[j+1]-Z[j])
}
estimatorsmu[1,k]<-mean(R)/eps
estimatorssigma[1,k] <-(length(R)-1)*var(R)/(length(R)*eps)

}
```

```
meansmu <- mean(estimatorsmu[1,])
meanssigma <- mean(estimatorssigma[1,])
sdmu <- sd(estimatorsmu[1,])
sdsigma <- sd(estimatorssigma[1,])

uppermu <- meansmu-qt(0.025,reps-1)*sdmu/sqrt(reps2)
lowermu <- meansmu+qt(0.025,reps-1)*sdmu/sqrt(reps2)
uppersigma <- meanssigma-qt(0.025,reps-1)*sdsigma/sqrt(reps)
lowersigma <- meanssigma+qt(0.025,reps-1)*sdsigma/sqrt(reps)
```

A.2.2 Geometric Brownian Motion

```
#This program uses a regular point in time approach to estimate the parameters
# of a geometric brownian motion process.

#Generates brownian motion process.
bmtime<-function(drift,diffusion,Time,eps,delta) {
  eps = delta*sqrt(pi/(2*1024*diffusion))
  Y = c(0, cumsum(rnorm(ceiling(Time/eps), drift*eps, sqrt(diffusion*eps))))
  return(Y)
}
#Parameters
delta<- 0.5
drift <- -10
diffusion <- 1
Time <- 5
reps <- 20
d<-vector();
# We want the expected amount to move to be less than delta/16.
eps =  delta*sqrt(pi/(2*1024*diffusion))

#container to store the estimators.
estimators <- array(dim=c(2,reps));
for(k in 1:reps) {

#Generate the Geometric Brownian Motion
Z<-exp(bmtime(drift,diffusion,Time,eps,delta))
p=drift/diffusion;
startdelta<-2
R<- vector()
for (j in 1:(length(Z)-1)) {
R<-append(R,Z[j+1]/Z[j])
}
R<-log(R)
sum=0
for (i in  1:length(R)) {
sum=sum+R[i]
}

sum2 = 0
for(j in 1:length(R)) {
sum2=sum2 + (R[j]-sum/length(R))^2

}
mu <- sum/(Time)
sigma <- sum2/Time
estimators[1,k] = mu
estimators[2,k] = sigma
}
```

A.2.3 Ornstein-Uhlenbeck Process

```
#This program uses a regular point in time appraoch to estimate the parameters
# of an Ornstein-Uhlenbeck Process.

generateOU <- function(start,mu,sigma,theta,deltat,steps) {
X<-c(start)
for(i in 1:steps) {
nek <- exp(-theta*deltat)*X[length(X)]+(1-exp(-theta*deltat))*mu
+sigma*sqrt((1-exp(-2*theta*deltat))/(2*theta))*rnorm(1)
X<- append(X,nek)
}
return(X)
}
```

```
#Parameters of the process
theta <- 4
mu <- 5
sigma <- 2
deltat <- 0.01
start <- -3
Time <- 100
steps <- Time/deltat
reps <- 25
#Form of the process is theta(mu-x)*deltat+sigma*sqrt(deltat)*rnorm
estimators <- estimators <- array(dim=c(3,reps))
for(k in 1:reps) {
X<-generateOU(start,mu,sigma,theta,deltat,steps)
Y<-X[2:length(X)]
X<- X[1:(length(X)-1)]
#Fit a linear model to the process.
model <- lm(Y~X)
int <- model$coefficients[1]
slope <- model$coefficients[2]
fit=summary(model)

#These are the estimators derived from the theory.
muestimatelm <- int/(1-slope)
thetaestimatelm <- -log(slope)/deltat
sigmaestimatelm <- sqrt((2*log(slope)*(fit$sigma)^2)/(deltat*(slope^2-1)))

estimators[1,k] <- muestimatelm
estimators[2,k] <- thetaestimatelm
estimators[3,k] <- sigmaestimatelm }
means <- c(rep(0,3))
stand <- c(rep(0,3))
for(k in 1:3) {
means[k] <- mean(estimators[k,])
stand[k] <- sd(estimators[k,])
}
# Create confidence intervals for the expected value of the estimators.
```

```
upper <- means-qt(0.025,reps-1)*stand/sqrt(reps)
lower <- means+qt(0.025,reps-1)*stand/sqrt(reps)
```