Universidad Nacional Autónoma de México

Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas

Estimation of discretely observed Markov processes

TESIS

para obtener el grado de Doctor en Ciencias

Autor:
FERNANDO BALTAZAR LARIOS

Director:
MOGENS BLADT PETERSEN

2011, Ciudad Universitaria, México

Acknowledgements

Thanks to my supervisor Mogens Bladt for giving me the opportunity to work with him, thanks for your patience, dedication, knowledge, for their great support and assistance.

Special thanks to Michael Sørensen for the support and assistance given to this work.

Thanks to Pablo, Eliane, Ramsés and Bo for contributions to this work.

To CONACYT for providing me with financial support.

Contents

Introduction

Markov processes are among the most important stochastic processes from both theory and applications. In this work We consider parameter estimation of Markov processes when they have been observed discretely.

The objective of this work is propose methods for obtaining estimates of parameters in Markov jump processes (MJP) and diffusion models when the data is a discrete time sample of the integral of the process.

In the first chapter, We introduce the concept of Markov process, we describe their main properties. We define a Markov jump process, study their infinitesimal generators and the principal properties. The limiting properties of these systems are similar to those of Markov chains, and these results are stated and finally we present an algorithm for simulation of Markov jump processes.

We present an overview of the diffusion processes and its main results, the estimation of their parameters when it has been continuously and discretely observed. We introduce the concept of integrated diffusion process and we propose a method for estimating the parameters in this case. Numerical examples are considered, We describe a diffusion bridge and describe a algorithm to simulate diffusion bridges.

The case when the Markov jump process is observed only at discrete time points has been studied in the last decade. An important application of MJP is in credit risk modelling.

In the second chapter, we present an algorithm for the estimation of transition rates by a Markov Chain Monte Carlo (MCMC) approach to observations from several Markov jump processes which conditional on an underlying Markov jump process are independent with the same transition rates. We apply our results to analysis of credit rating transition. The algorithm generalizes the results in [Bladt & Sørensen (2009] on estimation of transition rates by an MCMC approach to several observations.

In the rest of the work, we consider estimation for both general one-dimensional diffusion process and integrated diffusion processes. Likelihood based estimation (including Bayesian) for discretely observed diffusion processes has been investigated in the last decade and because of its importance in this work we return to this study assuming that we have other observations for the estimation.

In the chapter three, we consider the estimation of the parameters of diffusion processes when they have been continuously and discretely observed and We propose a method for estimating the parameters in this case. We consider maximum likelihood estimation in the situation where We do not observe the diffusion process itself directly, but instead observe integrals of the process over disjoint time-intervals. Moreover, the data are, assumed to be contaminated by measurement errors. The data can be viewed as incomplete observations of a model with a tractable likelihood function. Therefore we propose a simulated EM-algorithm to obtain maximum likelihood estimates of the parameters in the diffusion model. Numerical examples are considered.

Integrated diffusion processes play an important role in finance as models for realized volatility, see e.g. [Andersen & Bollerslev (1998], [T. Andersen & Labys (2001], [Bollerslev & Zhou (2002], and [Barndorff-Nielsen & Shephard (2002].

These processes are also used in fields of engineering and the sciences. An important example is provided by the records of the concentration of oxygen isotopes in ice-core data from Greenland and Antarctica, see e.g. [Ditlevsen, Ditlevsen & Andersen (2002]. Such data are used to investigate the paleo-climate.

In the last chapter, we apply the method developed in the last section of the previous chapter to the Ornstein-Uhlenbeck process where the parameters of an Ornstein-Uhlenbeck model are estimated from a set of integrated paleo-temperature data obtained from an ice-core from Greenland and a similar investigation for the Cox-Ingersoll-Ross(CIR)/square root process is presented.

Chapter 1

Markov Processes

In this chapter, we introduce the concept of Markov process which is a stochastic process with the Markov property, or memorylessness. It is one for which conditional on the present state of the system, its future and past are independent. Also we describe the main properties of Markov processes. We define a Markov jump process and study their properties. We present an overview of the diffusion processes and its main results, the estimation of their parameters when it has been continuously and discretely observed. We introduce the concept of integrated diffusion process and we propose a method for estimating the parameters in this case. Numerical Examples are considered. We describe a diffusion bridge and describe a algorithm to simulation of diffusion bridges.

1.1 Introduction

Markov processes are among the most important stochastic processes for both theory and applications. A Markov jump process is a stochastic process which remains in a state for an exponentially distributed time whose rate depends on the state.

Markov jump processes are natural candidates for modeling systems in real time such as production and inventory systems, computer and telecommunications networks, and miscellaneous input-output systems. Many continuous-time processes have discrete-time analogues; for instance, a birth-death process is continuous-time analogues of discrete-time random walks. One's choice of a continuous-time model for a system typically depends on how realistic it is, its easy in addressing the issues at hand, or in computing quantities of interest.

A Markov jump process is a continuous-time Markov process on a countable state space whose sample paths are right-continuous and piecewise-constant with finite lengths, and the number of jumps in any finite time is finite. This type of Markov process is represented by the sequence of states it visits and the sojourn times at the visits.

In the rest of chapter, we define a Markov jump process, we study their infinitesimal generator and the principal properties. The limiting properties of these systems are similar to those of a Markov chain, and these results are stated and finally we present an algorithm for simulation of Markov jump processes.

When we want to model a stochastic Markov process in continuous time it is almost impossible to specify in some reasonable manner a consistent set of finite dimensional distributions. The one exception is the family of Gaussian processes with specified means and covariances. It is much more natural and profitable to take an evolutionary approach. A rich and useful class of such Markov processes are the diffusion process.

The history of diffusion processes begins with the botanist Brown, who in 1826-1827 observed that grains of pollen suspended in a water displayed a certain type of erratic motion, which did not fit any of the contemporary mathematical models. This motion came to be known as the Brownian motion. Einstein, in 1905, used physical principles to do mathematical analysis of this motion and Wiener provided a rigorous mathematical foundation for the Brownian motion.

1.2 Definitions

We start with some important Definitions.

Definition 1.2.1 *Let T denote the time set under consideration and let* $(\Omega, \mathscr{F}, \mathbb{P})$ *be a common underlying probability space. A stochastic process $X = \{X_t\}_{t \in T}$ is a function of two variables*

$$X : T \times \Omega \to \mathbb{R},$$

with values in the state space E, where

1. *$X_t = X(t, *) : \Omega \to \mathbb{R}$ is a random variable for each $t \in T$,*
2. *$X(*, \omega) : T \to \mathbb{R}$ is a realization or sample path for each $\omega \in \Omega$.*

Depending on T being a discrete or a continuous time set, we call the stochastic process a discrete or a continuous time process. Assume that X has sample paths in a subspace $\mathscr{H}$ of the space $F(T, E)$ of all measurable functions from T to E.

Definition 1.2.2 *The finite-dimensional distributions of $X = \{X_t\}_{t \in T}$ are the push forward measures on the product space $\mathbb{R}^k$ for $k \in \mathbb{N}$ defined by*

$$P(X_{t_1}(\omega) \in A_1, \ldots, X_{t_k}(\omega) \in A_k) = F_{t_1, \ldots, t_n}(A_1, \ldots, A_n),$$

for all $t_i \in T$, $A_i \subset E$ in the state space of X, $\omega \in \Omega$ and F on the product space.

Now, we present some important characteristics of the stochastics processes.

1. A stochastic process $X = \{X_t\}_{t \in T}$ for which the random variables $X_{t_j+1} - X_{t_j}$ with $j = 1, \ldots, k-1$ are independent for any finite combination of time instants $t_1 < \ldots < t_k$ in T is a stochastic process with **independent increments**.

2. A stochastic process is $\{X_t\}_{t\in T}$ on a space E is stationary if, for any $s_1 < \ldots < s_n$, we have

$$(X_{s_1+t}, \ldots, X_{s_n+t}) \overset{d}{=} (X_{s_1}, \ldots, X_{s_n}),$$

for all $t \in T$, where $\overset{d}{=}$ is identically distributed.

3. A stationary process X is **ergodic** if each of its invariant events has probability 0 or 1, (an event A is invariant for X if there is a $B \in \mathscr{H}$ such that $\{X \circ \theta_t \in A\} = B, t \in T$, where $\{\theta_t\}_{t\in T}$ is the time-shift operator on T and $\circ$ denotes the convolution operator.

4. The process X is **reversible** it is stationary and $X_t \circ R_t^- \overset{d}{=} X_t$ for all $t \geq 0$, where R^- is the time-reversal on $T(R^- t = -t)$. Note that X is reversible if and only if $(X_t \circ \theta_t \circ R_t^- \overset{d}{=} X, t \in T$.

Definition 1.2.3 *Suppose that* X *and* Y *are stochastic processes on* $(\Omega, \mathscr{F}, \mathbb{P})$. *Then we say that* $\{X_t\}$ *is a version of (or a modification of)* $\{Y_t\}$ *if*

$$P(\{\omega; X_t(\omega) = Y_t(\omega)\}) = 1$$

for all $t \in T$ *.Note that if* X *is a version of* Y *, then* X *and* Y *have the same finite-dimensional distributions.*

Here, we consider some probability space $(\Omega, \mathscr{F}, \mathbb{P})$ and $T = \mathbb{R}^+$, a family $X = \{X_t\}_{t\geq 0}$ of random variables $X_t : \Omega \to E$ is called a continuous-time stochastic process on the state space E.

The index t admits the convenient interpretation as time: if $X_t = x$, the process is said to be in state x at time t. For some given $\omega \in \Omega$, the E-valued set $\{X_t(\omega) : t \geq 0\}$ is called a realization (trajectory, sample path) of the stochastic process X associated with ω.

Now, we define a concept which is very import in the study of stochastics processes.

Definition 1.2.4 *Let* $(\Omega, \mathcal{F}_t, \mathbb{P})$ *be a filtered probability space. Then a random variable* $\tau : \Omega \to I$ *is called a stopping time if* $\{\tau \leq t\} \in \mathcal{F}_t$ *for all* t *in* I.

Definition 1.2.5 ***(Markov Process)*** *A continuous-time stochastic process* $X = \{X_t\}_{t\geq 0}$ *on a countable state space* E *is a Markov process if for any* $t_0 < t_1 < \ldots < t_k < t_{k+1}$ *and* $C \subset E$*, it satisfies the Markov property:*

$$P(X_{t_{k+1}} \in C | X_{t_k}, \ldots, X_{t_0}) = P(X_{t_{k+1}} \in C | X_{t_k}).$$

Intuitively, this means that knowing the entire history of the process does not contain any more information than knowing its last value. Markov processes describe the time-evolution of random systems that do not have any memory.

A Markov process is a time-homogeneous process when the last probability does only depend on the time increment $t_{k+1} - t_k$, but not on t_k.

Definition 1.2.6 *The function*

$$p : \mathbb{R}_+^2 \times E \times E \to [0,1]$$

defined by

$$p_{xy}(s,t) = P(X_t = y | X_s = x)$$

with $s < t$ *and* $s,t \in \mathbb{R}_+$ *, is called the transition function.*

The values $p_{xy}(s,t)$ are the conditional probabilities that the process is in state y at time t given that it was in state x at time s. Given a homogeneous Markov process, then we have that

$$p_{xy}(s,t) = p_{xy}(t-s) = P(X_{t-s} = y | X_0 = x)$$

Definition 1.2.7 *The probability distribution* μ_0 *satisfying*

$$\mu_0(x) = P(X_0 = x)$$

is called the initial distribution. If there is a state $x \in E$ *such that* $\mu_0(x) = 1$*, then* x *is called the initial state.*

We have that the transition probabilities of a Markov process satisfy the Chapman-Kolmogorov equation

$$p_{xz}(s,v) = \int_y p_{xy}(s,t) p_{yz}(t,v) \tag{1.1}$$

for all $s < t < v$.

Now, we define two important concepts in the study of stochastics processes, a stochastic semigroup and stopping time.

Let K be a Banach space and $\mathscr{B}(K,K)$ the set of bounded operators of K in K.

Definition 1.2.8 *A family* $\{G(t)\}_{t\geq 0} \subset \mathscr{B}(K,K)$ *is a semigroup if:*

1. $G(0) = I$
2. $G(s+t) = G(t)G(s)$ *for all* $s,t \geq 0$.
   And when also satisfy
3. *If* $\lim_{t\to 0} \|G(t) - I\| = 0$*, we say that it is a uniformly continuous semigroup, where* $\|G\| = sup_{f\in K, \|f\|=1} \|Gf\|$*, where* $\|\|$ *is the usual*

1.3 Markov Jump Processes

In the following, we assume that the process is time-homogeneous Markov process with finite state space $E = \{1, 2, \ldots, m\}$, thus the term Markov process will always refer to a homogeneous Markov process.

Definition 1.3.1 *A matrix* $P(t) = \{p_{xy}(t))\}_{x,y\in E}$ *i.e. the* $m \times m$ *matrix with entries* $p_{xy}(t)$ *will be called a Markov transition matrix.*

The family $\{P(t)\}_{t\geq 0}$ is called the transition semigroup of the Markov process.

Theorem 1.3.1 *The family $\{P(t)\}_{t\geq 0}$ is a stochastic semigroup, that is, it satisfies the following:*

1. *$P(0) = I$, the identity matrix.*
2. *$P(t)$ is a stochastic matrix, that is:*
   (a) *$p_{ij}(t) \geq 0$ for all $i, j \in E$.*
   (b) *$\sum_{j=1}^{m} p_{ij}(t) = 1$ for all $i \in E$.*
3. *The Chapman-Kolmogorov equation, $P(s+t) = P(s)P(t)$.*

Proof. For 2) we consider $\mathbf{1}$ a row vector of ones, we have that

$$(P(t))_i \mathbf{1}^t = \sum_{j=1}^{m} p_{ij}(t) = P(\bigcup_{j\in E}\{X_t = j\}|X_0 = i) = 1,$$

where $(P(t))_i$ is the ith row of $P(t)$.

3) Using the Markov property

$$\begin{aligned} p_{ij}(s+t) &= P(X_{t+s} = j|X_0 = i) \\ &= \textstyle\sum_{k=1}^{m} P(X_{t+s} = j|X_s = k, X_0 = i)P(X_s = k|X_0 = i) \\ &= \textstyle\sum_{k=1}^{m} p_{ik}(s)p_{kj}(t). \end{aligned}$$

And 1) is implied by 3) taking $s = t = 0$. ■

The probabilistic behavior of a finite state, continuous time Markov process $X = \{X_t\}_{t\geq 0}$ is determined by the knowledge of stochastic semigroup $\{P(t)\}_{t\geq 0}$ and the distribution of X_0.

Now, the set of all stochastic matrices with respect to a fixed dimension m will be denoted by

$$\mathscr{P} = \{P(t) = (p(t)_{ij})_{i,j} \in \mathbb{R}^{m\times m}|0 \leq p(t)_{ij} \leq 1 \ \text{ and } \ \sum_j p(t)_{ij} = 1, t \geq 0\}.$$

Definition 1.3.2 *The semigroup $\{P(t)\}_{t\geq 0}$ is called standard if $P(t) \to 0$ as $t \downarrow 0$, which is to say that*

$$\lim_{t\to 0^+} p_{ij}(t) = \begin{cases} 1 & if \quad i = j \\ 0 & if \quad i \neq j \end{cases}$$

holds for all $i, j \in E$.

Note that the semigroup is standard if and only if its elements $p_{ij}(t)$ are continuous functions of t.

There are some subtleties in the realm of continuous time processes that are not present in the discrete-time case. These steam from the fact that the uncountable union of measurable sets need not be measurable anymore.

For Example, the mapping $X_t(*) : \Omega \to E$ is measurable for every $t \in \mathbb{R}^+$, i.e., $\{\omega \in \Omega | X_t(\omega) \in A\} \in \mathscr{F}$ for every measurable subset $A \subset E$. However,

$$\{\omega \in \Omega | X_t(\omega) \in A, t \in \mathbb{R}^+\} = \bigcap_{t \in \mathbb{R}^+} \{\omega \in \Omega | X_t(\omega) \in A\}$$

need not be in $\mathscr{F}$ in general.

We will therefore impose some regularity conditions on the Markov process in order to exclude pathological cases. Throughout this work, we assume that

$$p_{i,j}(0) = \delta_{ij} \tag{1.2}$$

where

$$\delta ij = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{if} \quad i \neq j \end{cases}$$

This guarantees that no transition can take place at zero time and we consider only Markov processes with standard semigroup of transitions probabilities, this guarantees that the realizations of $X = \{X_t\}_{t \geq 0}$ are right continuous functions, more precisely, it implies that the Markov process is stochastically continuous, separable and measurable on compact intervals. Moreover, there exists a separable version, being stochastically equivalentto this process. Due to the fact that the state space is discrete, continuity from the right of the sampling functions implies that they are step functions, that is, for almost all $\omega \in \Omega$ and all $t \geq 0$ there exists $\Delta_t(t, \omega) > 0$ such that

$$X_{t+\tau}(\omega) = X_t(\omega)$$

for $\tau \in [0, \Delta_t(t, \omega))$.

First for τ with $\tau < \infty$, then for unbounded τ by an easy argument. This fact motives the name Markov jump process. Then we can describe a Markov jump process as

Definition 1.3.3 *A Markov jump process is a continuous-time Markov process $X = \{X_t\}_{t \geq 0}$ with countable state space E which starts in an initial state x_0 at time $\tau_0 = 0$ and stays in this state until some time τ_1 when it makes a transitions to a different state x_1. It stays in this state until a later time $\tau_2 > \tau_1$ at which is jumps to different state x_2. Then if $\tau_1, \tau_2, \ldots$ are the set of jump times, $X_t = x_0$ for $t \in [0, \tau_1)$,$X_t = x_1$ for $t \in [\tau_1, \tau_2)$ and so on. We assume that* $\lim_{n \to \infty} \tau_n = \infty$.

We are interested in analyzing the distribution function governing the time that the Markov jump process stays in some state $i \in E$. We consider the time that the Markov jump process $X = \{X_t\}_{t \geq 0}$ spend in a certain state, when the process is in a state x, the time that it stays in this state is a random variable governed by some distribution function.

Suppose that $X_t = i$, the future development of X_{t+s}, for $s > 0$, goes roughly as follows. Let

$$\Delta(t) = inf\{s > 0 | X_{t+s} \neq i\} \tag{1.3}$$

be thetime until the process changes its state, then $\Delta(t)$ is a holding time. Obviously, $\Delta(t)$ is a stopping time. Hence, conditioned on X_t and $\Delta(t) < \infty$, the next jump will occur at time $t + \Delta(t)$. Otherwise the Markov process will not leave the state X_t anymore.

Proposition 1.3.1 *Consider some Markov jump process $X = \{X_t\}_{t\geq 0}$ being in state $i \in E$ at time $t \geq 0$. Then, there exists $\lambda(i) \geq 0$, independent of the time t, such that*

$$P(\Delta(t) > s|X_t = i) = \exp(-\lambda(i)s)$$

for all $s > 0$ and $\lambda(i)$ is called the jump rate associated with the state $i \in E$.

Proof. Since that Markov jump process is homogeneous $P(\Delta(t) > s|X_t = x) = P(\Delta(0) > s|X_0 = i)$ and we define $h(s) = P(\Delta(0) > s|X_0 = i)$, then we have

$$\begin{aligned} h(t+s) &= P(\Delta(0) > t+s|X_0 = i) = P(\Delta(0) > t, \Delta(t) > s|X_0 = i) \\ &= P(\Delta(0) > t|X_0 = i)P(\Delta(t) > s|\Delta(0) > t, X_0 = i) \\ &= h(t)P(\Delta(t) > s|, X_t = i) \\ &= h(t)h(s) \end{aligned}$$

Also, $h(*)$ is continuous at zero, since the transition probabilities were assumed to be continuous at zero. Moreover, $0 \leq g(*) \leq 1$, which finally implies that the only solution must be

$$h(s) = \exp(-\lambda(i)s)$$

with $\lambda(x) \in [0, \infty]$ given by $\lambda(i) = -\log(P(\Delta(0) > 1|X_0 = i))$. ■

Definition 1.3.4 *If $i \in E$ with associated jump rate $\lambda(i)$. Then, i is called*

1. *Permanent,if $\lambda(i) = 0$.*
2. *Stable, if $0 < \lambda(i) < \infty$.*
3. *Instantaneous, if $\lambda(i) = \infty$ (not present for Markov processes with right continuous sample paths).*

If $X_t = i$ at time t. If i is

1. Permanent, then
$$P(X_s = i|X_t = i) = 1$$
for every $s > t$, hence the Markov jump process stays in x forever.
2. Stable, then
$$P(0 < \Delta(t) < \infty|X_t = i) = 1.$$
3. Instantaneous, then
$$P(\Delta(t) = 0|X(t) = i) = 1,$$
hence the Markov jump process exists the state as soon as it enters it.

Since we consider only Markov jump process with right continuous samples paths, then the state space E does not contain instantaneous states.

For ease of notation we define the following random variables:

Definition 1.3.5 *We consider a Markov jump process $X = \{X_t\}_{t\geq 0}$ then we define*

1. $\tau_0 = 0$.
2. $\tau_k =$ *the time at which the kth jump occurs.*
3. $x_k =$ *the state visited during* $[\tau_k, \tau_{k+1})$.
4. $\Delta_k = \tau_{k+1} - \tau_k$, *i.e. the time spend in state* x_k
5. $N(t) =$ *the largest interger n, for which* $\tau_n < t$.

Definition 1.3.6 *Let $X = \{X_t\}_{t\geq 0}$ be a Markov jump process. Then, the Markov process is called regular if*

$$\tau_\infty := \sup_{k\in\mathbb{N}} \tau_k = \infty.$$

The next Proposition states that the time, at which the Markov process jumps next, and the state, it jumps into, are independent.

Proposition 1.3.2 *Let $X = \{X_t\}_{t\geq 0}$ be a regular Markov jump process on E with $\tau_{k+1} < \infty$ a.s. Then, conditioned on $X_{\tau_k} = i$, the random variables Δ_{k+1} and $X_{\tau_{k+1}}$ are independent, i.e.,*

$$P(\Delta_{k+1} > t, X_{\tau_{k+1}} = j|X_{\tau_k} = i) = P(\Delta_{k+1} > t|X_{\tau_k} = i)P(X_{\tau_{k+1}} = j|X_{\tau_k} = i)$$

Proof. By Bayes rule we have,

$$\begin{aligned}&P(\Delta_{k+1} > t, X_{\tau_{k+1}} = j|X_{\tau_k} = i)\\ &\qquad =P(\Delta_{k+1} > t|X_{\tau_k} = i)P(X_{\tau_{k+1}} = j|X_{\tau_k} = i, \Delta_{k+1} > t).\end{aligned}$$

By applying Markov property we have

$$\begin{aligned}&P(X_{\tau_{k+1}} = j|X_{\tau_k} = i, \Delta_{k+1} > t)\\ &\qquad =P(X_{\tau_{k+1}} = j, X_s = i, \tau_k \leq s < \tau_{k+1}|X_{\tau_k+t} = i)\\ &\qquad =P(X_{\tau_{k+1}+\Delta(\tau_k+t)} = j, X_s = i, \tau_k \leq s < \tau_{k+1} + \Delta(\tau_k + t)|X_{\tau_k} = i))\\ &\qquad =P(X_{\tau_{k+1}} = j|X_{\tau_k} = i).\end{aligned}$$

■

We can now define the embedded process of the Markov jump process.

Definition 1.3.7 *We call the embedded process of the Markov jump process $X = \{X_t\}_{t\geq 0}$ to the process $(Y_n, \tau_n)_{n\in\mathbb{N}}$ where $\{Y_n\}_{n\in\mathbb{N}}$ is a time homogeneous Markov chain on the state space E in terms of the following transition probability $\boldsymbol{P} = \{p_{ij}\}_{ij\in E}$. If i is permanent, set $p_{ii} = 1$. Otherwise, if i is stable, set*

$$p_{ij} = P(X_{\tau_1} = j|X_0 = i) \tag{1.4}$$

and consequently $p_{ii} = 0$

The evolution of Markov jump processes can be described in very much the same terms as those used for a discrete-time process (Markov chains). For discrete-time processes we write the n-step transition probabilities in matrix form and expressed them in terms of the one-step matrix $\mathbf{P}$. In a Markov jump process there is no exact analogue of $\mathbf{P}$ since there is no implicit unit length of time. The infinitesimal calculus offers one way to plug this gap. We shall see that there exists a matrix Q, called the infinitesimal generator of the Markov jump process, which takes over the role of $\mathbf{P}$. An alternative way of approaching the equation of a Markov jump process is to consider the embedded process obtained by listing the changes of states of the original process.

1.3.1 Infinitesimal Generator

In this section we present to the characterization of Markov jump processes that is not present for the discrete-time case. It is in terms of infinitesimal changes of the transition probabilities and based on the notion of generators. We again take a Markov jump process that satisfies the same conditions as in the previous section.

Suppose that the Markov jump process is in state i at time t, i.e. $X_t = i$, we are interested in the things that may happen in the small time interval $(t, t+h)$, we have the following cases:

1. The process may stay in the same state at time $t+h$, this happen with probability $p_{ii}(h) + o(h)$, the error term taking into account the possibility that the process moves out of i and back to i in the interval.

2. The process may move to new state j with probability $p_{ij}(h) + o(h)$, we are assuming here that the probability of two or more transitions in the interval $(t, t+h)$ is $o(h)$.

Then we are interested in the behaviour of $p_{ij}(h)$ for small h, for this we we consider the follow Proposition.

Proposition 1.3.3 *Consider the semigroup $P(t)$ of a Markov jump process. Then, the limit*

$$\lim_{h\to 0^+} \frac{P(h) - I}{h}$$

i.e. the limits

$$\lim_{h\to 0^+} \frac{1 - p_{ii}(h)}{h} = q_i$$

$$\lim_{h\to 0^+} \frac{p_{ij}(h)}{h} = q_{ij} \tag{1.5}$$

with $i \neq j$ exists and defines the infinitesimal generator, where $0 \leq q_{ij} < \infty$ and $0 \leq q_i \leq \infty$.

Proof. See [Karlin & Taylor (1975]. ■

Now, considering the relation

$$1 = p_{ii}(h) + \sum_{i \neq j} p_{ij}(h)$$

dividing by h and letting h decrease to zero yield the relation

$$q_i = \sum_{i \neq j} q_{ij}.$$

Now, we consider the $m \times m$ matrix

$$Q = \begin{pmatrix} -q_1 & q_{12} & \cdots & q_{1m} \\ q_{21} & -q_2 & \cdots & q_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ q_{m1} & q_{m2} & \cdots & -q_m \end{pmatrix}$$

By (1.5) we have

$$\lim_{h \to 0^+} \frac{P(h) - I}{h} = Q \tag{1.6}$$

Using Theorem 1.3.1 and 1.6 we have

$$\frac{P(t+h) - P(t)}{h} = \frac{P(t)(P(h) - I)}{h} = \frac{P(h) - I}{h} P(t), \tag{1.7}$$

we take limit when h drecrease to zero and we get that the derivate of $P(t)$ is given by

$$P'(t) = P(t)Q = QP(t) \tag{1.8}$$

where $P'(t)$ denotes the matrix whose elements are $p'_{ij}(t)$.

Equation (1.8) can be solved under the initial condition $P_0 = I$ by standard methods of system of ordinary differential equations to yield the formula

$$P(t) = \exp(tQ) = \sum_{n=0}^{\infty} \frac{t^n Q^n}{n!}. \tag{1.9}$$

with $\exp(*)$ denoting the matrix exponential function. Q is called the infinitesimal generator of Markov process $X = \{X_t\}_{t \geq 0}$.

A matrix $Q \in \mathbb{R}^{m \times m}$ generates a continuous-time Markov process if and only if all off-diagonal entries are nonnegative and the sum over each row equals zero and the set of all generators with respect to a fixed dimension m will be denoted by

$$\mathscr{Q} = \{Q = (q_{ij})_{i,j} \in \mathbb{R}^{m \times m} | q_{ij} \geq 0 \text{ if } i \neq j, q_{ii} = -\sum_{i \neq j} q_{ij}\}.$$

Theorem 1.3.2 *Let Q be the infinitesimal generator of the Markov jump process $X = \{X_t\}_{t \geq 0}$ with state space $E = \{1, 2, \ldots, m\}$. Set $P(t) = \exp(tQ)$. Then the semigroup $\{P(t)\}_{t \geq 0}$ has the following properties:*

1. *$\{P(t)\}_{t \geq 0}$ is the unique solution to the* ***Kolmororov forward equation***

$$\frac{d}{dt} P(t) = P(t)Q,$$

with $P(0) = I$, i.e. $p_{ij}(t)$ satisfy

$$p'_{ij}(t) = \sum_k p_{ik}(t) q_{kj}.$$

2. *$\{P(t)\}_{t\geq 0}$ is the unique solution to the* ***Kolmororov backward equation***

$$\frac{d}{dt}P(t) = QP(t),$$

with $P(0) = I$, i.e. $p_{ij}(t)$ satisfy

$$p'_{ij}(t) = \sum_{k} q_{ik}p_{kj}(t).$$

3. *For $k = 0, 1, 2, \ldots,$ we have*

$$\frac{d^k}{dt}|_{t=0}P(t) = Q^k.$$

Proof. The results are immediate of (1.8). ■

Here, we introduce another important family of parameters for Markov jump process

Definition 1.3.8 *The transition rates of the Markov jump process $X = \{X_t\}_{t\geq 0}$ are*

$$q_{ij} = q_i p_{ij}.$$

for $i \neq j$.

In this Definition q_{ij} are the elements of the infinitesimal generator and this relation follows of backward and forward equations.

The following Theorem will serve as the basis for constructing a density on the space of sample paths of Markov jump process.

Theorem 1.3.3 *Let Q the infinitesimal generator of the Markov jump process $X = \{X_t\}_{t\geq 0}$ then*

1. *Conditioned on the Markov jump process is in the state i at time t, the random variable $\Delta(t)$ (holding time) is exponentially distributed with parameter q_i, i.e.*
$$P(\Delta(t) > s|X_t = i) = exp(-q_i s).$$

2. *If $X_t = i$ and $q_i > 0$, there is, with probability 1, a sample function discontinuity for some $t > 0$, and in fact, a first discontinuity which is a jump. If $0 < s \leq \infty$, the conditional probability that the first discontinuity in $[t, t+s)$ is a jump to j, given that $X_t = i$ and that there is a discontinuity in $[t, t+s)$, is $\frac{q_{ij}}{q_i}$.*

Proof. 1) By Proposition 1.3.1 we have that the distribution function $F_\Delta(t)$ of Δt is exponential and so $1 - F_\Delta(t) = e^{-\lambda_i t}$ where $\lambda_i = F'_\Delta(0) = q_i$.

2) Suppose that $0 < h, t$ and that the process jumps only once in $(t, t+h]$, then

$$P(\text{jump to } j \text{ in some time in}(t, t+h)|X_t = i) \backsimeq \frac{p_{ij}(h)}{1 - p_{ii}} \rightarrow \frac{q_{ij}}{q_i}$$

as $h \downarrow 0$. ■

1.3.2 Markov Properties

In this section we prove important properties about the behavior of a finite state continuous time Markov jump process. Our first observation is a characterization of a Markov jump process in terms of elementary properties of its embedded process.

Theorem 1.3.4 *A Markov jump process $X = \{X_t\}_{t\geq 0}$ on E with embedded process $(Y_n, \Delta_{t_n})_{n\in\mathbb{N}}$ satisfies*

1. *$\{Y_n\}_{n\in\mathbb{N}}$ is a discrete-time Markov chain on E with transition probabilities $P = \{p_{i,j}\}_{i,j\in E}$.*

2. *For non negative $t_1, t_2, \ldots, t_k$*

$$P(\Delta_1 \leq t_1, \ldots, \Delta_k \leq t_k | Y_n, n \geq 1) = \prod_{n=1}^{k} P(\Delta_t \leq t_n | Y_n)$$

and for $i \in E$ and for each $n \geq 1$ we have

$$P(\Delta_n \leq t | Y_n = i) = 1 - e^{-q_i t}$$

with $i \in E$ and $t \geq 0$.

Remark 1.3.1 *If we consider a Markov jump process $X = \{X_t\}_{t\geq 0}$ on E with embedded process $(Y_n, \Delta_{t_n})_{n\in\mathbb{N}}$ then $(Y_n, \Delta_{t_n})_{n\in N}$ is a discrete-time Markov chain on $E \times \mathbb{R}^+$ with transition probabilities*

$$P(Y_{n+1} = j, \Delta_{n+1} \leq t | Y_n = i, \Delta_n) = p_{ij}(1 - e^{-q_i t}).$$

Proof. To prove the Proposition, it suffices by Remark 1.3.1 show that, for $i, j \in E, s_k, t > 0, n \geq 0$

$$P(Y_{n+1} = j, \Delta_{n+1} > t | Y_k, \Delta_k = s_k, k \leq n, Y_n = i) = p_{ij} e^{-q_i t}, \tag{1.10}$$

for Markov properties.

Using the Markov property of X at the time

$$\tau_{n+1} = \sum_{k=0}^{n} \Delta_k,$$

the probability on the left-hand side of (1.10) equals

$$\begin{aligned} P(X_{\tau_{n+1}} = j, \Delta_{n+1} > t | X_{\tau_n} = i) = P(X_{\tau_{n+1}} = j | X_{\tau_n} = i) \\ \times P(\Delta_{n+1} > t | X_{\tau_n} = i, X_{\tau_{n+1}} = j), \end{aligned}$$

since

$$P(X_{n+1} = j | X_{\tau_n} = i) = p_{ij}$$

and

$$\begin{aligned} &P(\Delta_{\tau_{n+1}} > t | X_{\tau_n} = i, X_{\tau_{n+1}} = j) \\ &\quad = P(X_u = i, u \in [\tau_n, \tau_n + t] | X_v = i, v \in [\tau_n, t), X_{\tau_{n+1}} = j) \\ &\quad = P(X_u = i, u \in [0, t] | X_v = i, v \in [0, t), X_{\Delta_1} = j) \\ &\quad = P(\Delta_1 > t | X_0 = i) = e^{-q_i t}, \end{aligned}$$

we have that (1.10) is true, which prove the Theorem. ■.

The following Proposition give a explicit form for the transition probabilities of a Markov jump process.

Proposition 1.3.4 *If the Markov jump process $X = \{X_t\}_{t\geq 0}$ on E is such that its embedded process satisfies conditions of Theorem 1.3.4, then its transition probabilities satisfy*

$$p_{ij}(t) = e^{-q_i t}\delta_{ij} + \int_o^t \sum_{k\neq i} p_{kj}(t-s)q_{ik}e^{-q_i s}ds.$$

where

$$\delta_{ij} = \begin{cases} 1 & \textit{if} \quad i = j \\ 0 & \textit{if} \quad i \neq j \end{cases}$$

Proof. For fixed $i, j \in E$ and conditioning on $N(t)$ of the Definition 1.3.5, we have

$$\begin{array}{l} P(X_{t+u} = j|X_t = i, X_s, s < t) = \\ \sum_{n=0}^{\infty}(\alpha_n(t,u) + \beta_n(t,u))P(N(t) = n|X_t = i, X_s, s < t) \end{array}$$

where

$$\begin{aligned} \alpha_n(t,u) &= P(X_{t+u} = j, t_{n+1} > t + u|F_n(i,t)) \\ \beta_n(t,u) &= P(X_{t+u} = j, t_{n+1} \leq t + u|F_n(i,t)) \\ F_n(i,t) &= \{N(t) = n, X_t = i, X_s, s < t\} \end{aligned}$$

Since that the residual time $t_{n+1} - t$ at time t is exponentially distributed with rate q_i, then

$$\beta_n(t,u) = \delta_{ij}P(t_{n+1} - t > u|F_n(i,t)) = \delta_{ij}e^{-q_i u}.$$

Now, conditioning on X_{n+1} and t_{n+1} and using Remark 1.3.1

$$\begin{aligned} \beta_n(t,u) = & \sum_{k\neq i}\int_0^u P(X_{t+u} = j|t_{n+1} = t + v, X_{n+1} = k, F_n(i,t)) \\ & \times P(X_{n+1} = k, t_{n+1} - t \in dv|F_n(i,t)) \\ = & \sum_{k\neq i}\int_0^u p_{kj}(u-v)q_i p_{ik}e^{-q_i v}dv \end{aligned}$$

using the transition rates, we have

$$\beta_n(t,u) = \sum_{k\neq i}\int_0^u p_{kj}(u-v)q_{ik}e^{-q_i v}dv$$

Substituting these expressions for and noting that they are independent of n and t, we have

$$P(X_{t+u} = j|X_t = i, X_s, s < t) = e^{-q_i t}\delta_{ij} + \sum_{k\neq i}\int_o^u p_{kj}(u-v)q_{ik}e^{q_i v}dv,$$

Since this expression is true for all $t \geq 0$, by setting $t = 0$ on the left-hand side, the right-hand side must equal $p_{ij}(u)$, which completes the proof. ■.

Let $\mathbb{F}_t = \sigma(X_s, s \leq t)$. The σ-algebra $\mathbb{F}_t$ consists of the measurable sets A such that

$$A \bigcap \{\tau \leq t\} \in \mathbb{F}_t$$

for all $t \geq 0$

Definition 1.3.9 *A non-negative random variable τ is called a stopping time for $\{X_t\}_{t\geq 0}$ if $\{\tau \leq t\} \in \mathbb{F}_t$ for all t.*

Definition 1.3.10 *A Markov jump process $\{X_t\}_{t\geq 0}$ on a state space E satisfies the **strong Markov property** if, for any stopping time τ , being finite a.s.,*

$$P(X_{s+\tau} \in A|X_\tau = x, X_t, t < \tau) = P(X_s \in A|X_0 = x)$$

for every $A \in \mathscr{F}$, whenever both sides are well-defined.

1.3.3 Stationary distributions and limiting probabilities

The asymptotic behaviour of $X = \{X_t\}_{t\geq 0}$ for a large t is closely bound with the existence of stationary distributions.

Definition 1.3.11 *The vector $\pi = (\pi_1, \ldots, \pi_m)$ is a stationary distribution of Markov jump process $X = \{X_t\}_{t\geq 0}$ with stochastic semigroup $\{P(t)\}_{t\geq 0}$ if*

1. *$\pi_i \geq 0$ for $i = 1, \ldots, m$.*
2. *$\sum_{i=1}^{m} \pi_i = 1$*
3. *$\pi = \pi P(t)$ for all $t \geq 0$.*

More generally, any measure π on E that satisfies $\pi = \pi P(t)$, $t \geq 0$, is an invariant measure for X_t.

Remark 1.3.2 *A vector π which satisfies $\pi = \pi P(t)$ for all $t \geq 0$ is called a stationary distribution for it makes the process stationary. That is, if we set the initial distribution of X_0 to be such a π, then the distribution of X_t will also be π for all $t \geq 0$ i.e.*

$$P(X_t = j) = \pi_j$$

for all $j \in E$ and all $t > 0$.

To see this, set the initial distribution of X_0 to be π_0 and compute $P(X_t = j)$ by conditioning on X_0. This gives

$$\begin{aligned} P(X_t = j) &= \sum_{i\in E} P(X_t = j|X_0 = i)P(X_0 = i) \\ &= \sum_{i\in E} p_{ij}(t)\pi_i = (\pi P(t))_i = \pi_i. \end{aligned}$$

Proposition 1.3.5 *For the Markov jump process X, the following statements are equivalent.*

1. *X is a stationary process.*

2. $X_t \overset{d}{=} X_0$*,* $t > 0$*.*

3. The distribution of X_0 *is a stationary distribution.*

Proof. The basic idea of the proof following of Remark 1.3.2, for details see [Grimmett & Stirzaker (2001].

The next Theorem give a relation of the stationary distribution and infinitesimal generator.

Theorem 1.3.5 *We have that* $\pi = \pi P(t)$ *for all* t *if and only if* $\pi Q = \bar{0}$*, where* $\bar{0}$ *is a vector of zeros.*

Proof. Remembering that $Q^0 = I$ then

$$\begin{aligned} \pi Q = \bar{0} \quad &\Leftrightarrow \quad \pi Q^n = \bar{0} \\ &\Leftrightarrow \quad \sum_{n=1}^{\infty} \frac{t^n}{n!} \pi Q^n = \bar{0} \\ &\Leftrightarrow \quad \pi \sum_{n=0}^{\infty} \frac{t^n}{n!} \pi Q^n = \pi \\ &\Leftrightarrow \quad \pi P(t) = \pi \end{aligned}$$

■

We should convince yourself that the implications are true in both directions in each of the lines above.

Thus, we see that the condition $\pi = \pi P(t)$ for all $t \geq 0$, which would be quite difficult to check, reduces to the much simpler condition $\pi Q = \bar{0}$, in terms of the generator matrix Q. The equations $\pi Q = \bar{0}$ are a set of m linear equations which, together with the normalization constraint $\sum_{i \in E} \pi_i = 1$, determines the stationary distribution π if one exists.

The jth equation in $\pi Q = \bar{0}$ is given by

$$0 = -q_j \pi_j + \sum_{i \neq j} q_{ij} \pi_i$$

which is equivalent to

$$q_j \pi_j = \sum_{i \neq j} q_{ij} \pi_i \tag{1.11}$$

Now, we can give the following interpretation:

1. On the left hand side, π_j is the long run proportion of time that the process is in state j, while q_j is that rate of leaving state j when the process is in state j. Thus, the product $q_j \pi_j$ is interpreted as the long run rate of leaving state j.

2. On the right hand side, q_{ij} is the rate of going to state j when the process is in state i, so the product $q_{ij} \pi_i$ is interpreted as the long run rate of going from state i to state j. Summing over all $i \neq j$ then gives the long run rate of going to state j. That is, the equation (1.11) is interpreted as the long run rate out of state j equal to the long run rate into state j.

For this reason the equations $\pi Q = \bar{0}$ are called the Global Balance Equations, or just Balance Equations, because they express the fact that when the process is made stationary, there must be equality, or balance, in the long run rates into and out of any state.

Now, we now consider the limiting probabilities

$$\lim_{t \to \infty} p_{ij}(t)$$

Theorem 1.3.6 *Let $X = \{X_t\}_{t \geq 0}$ be a Markov jump process, if a stationary distribution π exists, then it is unique and*

$$\lim_{t \to \infty} p_{ij}(t) = \pi_i$$

for all i.

Proof. See [Grimmett & Stirzaker (2001].

1.3.4 Examples

In this section we describe some standard Examples of Markov jump process.

Example 1.3.1 ***Poisson Process.*** *Consider an independent and identically distributed sequence $\{\Delta_k\}_{k \in \mathbb{N}}$ of exponential random variable with parameter $\lambda > 0$ and define recursively the sequence of random variable $\{\tau_k\}_{k \in \mathbb{N}}$ by*

$$\tau_{k+1} = \tau_k + \Delta_k$$

for $k \leq 1$. Here, τ_k is called the kth event time and Δ_k the inter-event time. Then, the sequence of random variables $\{X_t\}_{t \geq 0}$ defined by

$$X_t = \sum_{k=0}^{\infty} 1_{\{\tau_k \leq t\}} = \max\{k \geq 0 | \tau_k \leq t\}.$$

for $t \geq 0$ and with $X_0 = 0$. $\{X_t\}_{t \geq 0}$ is called a homogeneous Poisson process with intensity λ.

By construction, X_t is counting the number of events up to time t and is a Markov jump process with one-step transition probabilities

$$p_{ij} = \begin{cases} 1 & \text{if} \quad j = i + 1 \\ 0 & \text{if} \quad j \neq i + 1 \end{cases}$$

And exponential sojourn rates $q_i = \lambda$, which are obviously P-regular.

Example 1.3.2 ***Birth-Death Process.*** *Suppose that X_t represents the number of discrete items in a population at time t, where births and deaths in the population occur as follows. Whenever the population state is 0, the time to the next birth is exponentially distributed with rate λ_0. Also, whenever there are $i \geq 1$ items in the population, the time to the next birth is exponentially distributed with rate λ_i , and the time to the next death is exponentially distributed with rate μ_i. These times are independent and independent of the rest of the process. Assume the birth and death rates are bounded.*

Under these conditions, it follows as in the preceding Example that X_t is a Markov jump process on $\mathbb{Z}^+$ with transition rates

$$q_{ij} = \begin{cases} \lambda_i & \text{if} \quad j = i+1 \\ \mu_i & \text{if} \quad j = i-1 \end{cases}$$

The X_t is called a birth-death process with birth rates λ_i and death rates μ_i. Its exponential sojourn rates are

$$q_i = \lambda_i + \mu_i,$$

where $\mu_0 = 0$, and its one-step transition probabilities are

$$p_{ij} = \begin{cases} \frac{\lambda_i}{\lambda_i+\mu_i} & \text{if} \quad j = i+1 \\ \frac{\mu_i}{\lambda_i+\mu_i} & \text{if} \quad j = i-1 \end{cases}$$

There are a variety of queueing processes and general input-output processes that are birth-death processes.

1.3.5 Generating Markov Jump Processes

In this section we present an algorithm to simulation of Markov jump process.

The generation of Markov jump processes is quite similar to the generation of Markov chains. Suppose $X = \{X_t\}_{t\geq 0}$ is a Markov jump process on state space $E = \{1, 2, \ldots, m\}$ with infinitesimal generator $Q = \{q_{ij}\}_{i,j\in E}$, where $Q \in \mathscr{Q}$ and μ_0 is the initial distribution.

We suppose that $Y = \{Y_n\}_{n\in\mathbb{N}}$ is the embedded Markov chain associated to X. Remembering that the time spent in each state $i \in E$ is exponentially distributed with a parameter that may depend on i. The one-step transition matrix, say $P = \{p_{ij}\}_{i,j\in E}$, of Y, can be found directly from Q. Following the results in the previous sections we have that

$$p_{ij} = \begin{cases} \frac{q_{ij}}{q_{ii}} & \text{if} \quad i \neq j \\ 0 & \text{if} \quad i = j \end{cases}$$

We can observe that the probabilities are simply proportional to the rates.

Remembering that the holding times as $\Delta_1, \Delta_2 \ldots$ and the jump times as $\tau_1, \tau_2 \ldots$ we can simulate a typical realization of a Markov jump process with finite state space using the next algorithm.

Algorithm generating Markov Jump Process

1. Initialize τ_0. Draw Y_0 from the initial distribution μ_0. Set $X_0 = Y_0$ and $n = 0$.

2. Set $i = Y_n$

3. Draw Δ_{n+1} from $\text{Exp}(q_{ii})$.

4. Set $\tau_{n+1} = \tau_n + \Delta_{n+1}$.
5. Set $X_t = Y_n$ for $\tau_n \leq t < \tau_{n+1}$.
6. Draw Y_{n+1} from the distribution corresponding to the Y_n-th row of P.
7. Set $n = n + 1$
8. Go to Step 2.

Simulation of a Markov Jump process.

Here, we present a simple simulation of a realization of the Markov jump process X on the state space $E = \{1, 2, 3\}$.

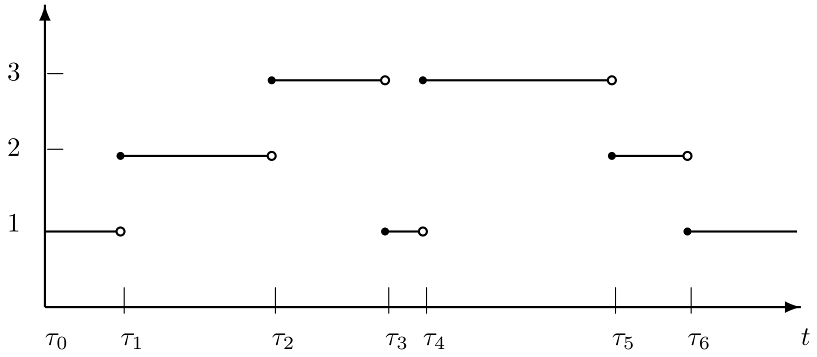
We suppose that the infinitesimal generator is known:

$$Q = \begin{pmatrix} -0.10 & 0.05 & 0.05 \\ 0.10 & -0.20 & 0.10 \\ 0.15 & 0.50 & -.30 \end{pmatrix}.$$

Then, the corresponding transition matrix P of the embedded Markov chain associate Y is:

$$P = \begin{pmatrix} 0.911 & 0.046 & 0.043 \\ 0.093 & 0.827 & 0.080 \\ 0.130 & 0.120 & 0.750 \end{pmatrix}.$$

We use the previous algorithm for the simulation,the picture 1.1 is an illustration that a realization of X in the time interval $[0, 10]$.



Picture 1.1:
A Typical realization of a Markov jump process.

1.4 Diffusion Processes

In this section we describe a class of stochastic processes called the diffusion processes. These are continuous-time, continuous state-space processes and their sample paths are everywhere continuous but nowhere differentiable. Since diffusions are defined through stochastic differential equations, we give a brief introduction to stochastic differential equations. We will discuss some major properties of diffusion processes and some common Examples of diffusion processes.

1.4.1 Definitions.

We start considering the following Definition.

Definition 1.4.1 *The continuous-time stochastic process* $X = \{X_t\}_{t\geq 0}$ *with continuous state space defined on the probability space* $(\Omega, \mathscr{B}(\mathbb{R}), P)$ *where,* $\mathscr{B}(\mathbb{R})$ *is the Borel* σ*-algebra on* $\mathbb{R}$ *, is a Markov process if it satisfies the following Markov property:*

$$P(X_t \in B | X_{t_1} = x_1, \ldots, X_{t_n} = x_n, X_s = x) = P(X_t \in B | X_s = x)$$

for all Borel subsets $B \subset \mathbb{R}$*, time instants* $0 \leq t_1 \leq \ldots \leq t_n \leq s \leq t$ *and all* $x_1, \ldots, x_n, x \in \mathbb{R}$ *for which the conditional probabilities are defined.*

For fixed $s, t \geq$ and x in the state space, the transition probability $P(X_t \in B | X_s = x)$ is a probability measure on the σ-algebra $\mathscr{B}(\mathbb{R})$ of Borel such that

$$P(X_t \in B | X_s = x) = \int_B p_{xy}(s,t)dy$$

for all $B \in \mathscr{B}(\mathbb{R})$. The quantity $p_{xy}(s,t)$ is the transition density. It plays a similar role as the transition matrix in Markov jump process.

Definition 1.4.2 *Let* $f : \mathbb{R} \to \mathbb{R}$ *be a bounded measurable function. The Markov process* $X = \{X_t\}_{t\geq 0}$ *is* ***ergodic*** *if*

$$\lim_{T\to\infty} \frac{1}{T} \int_0^T f(X_t)dt = \int_{-\infty}^{\infty} f(y)\bar{p}(y)dy \tag{1.12}$$

where

$$\bar{p}(y) = \int_{-\infty}^{\infty} p_{xy}(s,t)\bar{p}(x)dx$$

is stationary probability density.

This means that the time average limit coincide with the spatial average. Then is ergodic if its statistical properties can be deduced from a single, sufficiently long sample of the process.

Definition 1.4.3 *A one-dimensional* ***diffusion process*** $\{X\}_{t\geq 0}$ *is a continuous time stochastic process which possesses the strong Markov property and for which the sampler paths are (almost always) continuous functions of* t.

Every diffusion process whose state space is an interval $I = [r, s]$ satisfies the following condition:

$$\lim_{h \downarrow 0} \frac{1}{h} P(|X_{t+h} - x| > \epsilon | X_t = x) = 0 \tag{1.13}$$

for all $x \in I$ and every $\epsilon > 0$.

Almost all diffusion process are characterized by two basic conditions which augment (1.13) and describe the mean and variance of the infinitesimal displacements. We have the existence of the limits

$$\lim_{h \downarrow 0} \frac{1}{h} E(\Delta_h X_t | X_t = x) = \mu(x, t) \tag{1.14}$$

and

$$\lim_{h \downarrow 0} \frac{1}{h} E((\Delta_h X_t)^2 | X_t = x) = \sigma^2(x, t) \tag{1.15}$$

where $\Delta_h X_t = X_{t+h} - X_t$ and $x \in I$. The functions $\mu(x, t)$ and $\sigma^2(x, t)$ are called the infinitesimal parameters of the process, in particular $\mu(x, t)$ called the drift parameter and $\sigma^2(x, t)$ the diffusion parameter.

Generally, $\mu(x, t)$ and $\sigma^2(x, t)$ are continuous functions of x and t, and a regular process has $\sigma^2(x, t)$ positive in I and $t > 0$.

In other words, condition (1.13) prevents the diffusion process from having instantaneous jumps. From (1.14) and (1.15) one can see that $\mu(x, t)$ and $\sigma^2(x, t)$ are respectively the instantaneous rate of change of the mean and the instantaneous rate of change of the squared fluctuations of the process, given that $X_t = x$.

Definition 1.4.4 *The* ***hitting time*** *of the process $\{X_t\}_{0 \leq t \leq \tau}$ to the level z by*

$$T_z = \begin{cases} \infty & \text{if} \qquad X_t \neq z \\ inf\{t \geq 0; X_t = z\} & \text{in otherwise} \end{cases}$$

for $0 \leq t \leq \tau$.

Definition 1.4.5 *A diffusion process $\{X\}_{t \geq 0}$ whose state space is an interval $I = [r, s]$ is* ***regular*** *if*

$$P(T_z < \infty | X_0 = x) > 0$$

for all $x, z \in I$.

Definition 1.4.6 ***Conservative Process Diffusion with Killing****. A diffusion with killing $\{X_t\}_{0 \leq t \leq \tau}$ is a process whose paths behave those of a regular diffusion until a possibly random, possibly infinite time τ when the process is killed. If $\tau = \infty$ from all starting points, the process is said to be conservative then, a regular process is conservative if*

$$P(X_t \in I | X_0 = x) = P(\tau > t | X_0 = x) = 1$$

for all $t \geq 0$ and $x \in I$.

A characterization of a diffusion process.

Definition 1.4.7 *A standard process is a strong Markov process $\{X\}_{t\geq 0}$ whose paths posses the following regularity properties:*

1. *X_t is right continuous.*
2. *Left limits of X_t exist.*
3. *X_t is continuous from the left through stopping times (quasi-left continuity).*

A sufficient condition that a standard processes be a diffusion is the fulfillment of the **Dynkin condition**:

$$\lim_{h\downarrow 0}\frac{1}{h}P(|X_{t+h}-X_t|>\epsilon|X_t=x)=0$$

for all $x\in I$ and every $\epsilon>0$.

Theorem 1.4.1 *Let $\{X_t\}_{t\geq 0}$ be a standard process and suppose the Dynkin condition holds. Then $\{X_t\}_{t\geq 0}$ is a diffusion process.*

Proof See [Karlin & Taylor (1981].

The following Lemma gives a criterion to determine when the Dynkin condition is satisfied

Lemma 1.4.1 *If a standard process satisfies the infinitesimal moment condition*

$$\lim_{h\downarrow 0}\frac{1}{h}E(|\Delta_h X_t|^p|X_t=x)=0$$

for some $p>2$ uniformly for x in any compact subinterval of I and t finite, then the Dynkin condition is satisfied.

Proof. It is direct uses the Chebyshev inequality. ■

Diffusion processes are almost surely continuous functions of time, but they need not to be differentiable. Without going into the mathematical details, the continuity of a stochastic process can be defined in terms of continuity with probability one, mean square continuity and continuity in probability or distribution.

Theorem 1.4.2 *Let the stochastic process $\{X_t\}_{t\geq 0}$ be a diffusion process for which $\mu(x,t)$ and $\sigma^2(x,t)$ are moderately smooth. The forward evolution of its transition density $p_{xy}(s,t)$ is given by the* ***Fokker-Planck equation*** *also known as the* ***Kolmogorov forward equation***

$$\frac{\partial p}{\partial t}+\frac{\partial}{\partial y}\mu(y,t)p-\frac{1}{2}\frac{\partial^2}{\partial y^2}\sigma^2(y,t)p=0$$

for a fixed initial state (x,s).

The backward evolution of the transition density $p_{xy}(s,t)$ is given by the ***Kolmogorov backward equation***

$$\frac{\partial p}{\partial s} + \mu(x,s)\frac{\partial p}{\partial x} + \frac{1}{2}\sigma^2(y,t)\frac{\partial^2 p}{\partial x^2} = 0 \tag{1.16}$$

for a fixed initial state (y,t).

Proof. We proof only (1.16). Consider the approximate time discrete continuous state process with two equally probable jumps from (x,s) to $(x+\mu\Delta s \pm \sigma\sqrt{\Delta s}, s+\Delta s)$, which is consistent with (1.14) and (1.15). The approximate transition probability is then given by

$$\hat{p}_{xy}(s,t) = \frac{1}{2}\hat{p}_{x+\mu\Delta s+\sigma\sqrt{\Delta s}y}(s+\Delta s,t) + \frac{1}{2}\hat{p}_{x+\mu\Delta s-\sigma\sqrt{\Delta s}y}(s+\Delta s,t)$$

Taking Taylor expansions up to the first order in Δs around $(x,s;y,t)$ leads to

$$\frac{\partial \hat{p}}{\partial s}\Delta s + \mu\frac{\partial \hat{p}}{\partial x} + \frac{1}{2}\sigma^2\frac{\partial^2 \hat{p}}{\partial x^2}\Delta s + \mathcal{O}(\Delta s)^{3/2} = 0$$

Since the discrete time process converges in distribution to the diffusion process, we obtain the backward Kolmogorov equation when $\Delta s \to 0$. ■

Transformations of Processes

A continuous strictly increasing function g may be used to transform an arbitrary stochastic process X into a new process defined by $Y_t = g(X_t)$, if X is a diffusion then Y is a diffusion too.

Theorem 1.4.3 *Let $\{X_t\}_{t\geq 0}$ be a regular diffusion whose state space is an interval $I=[r,s]$ with parameters $\mu(x,t)$ and $\sigma^2(x,t)$. Let g be a strictly monotone function on I with continuous second derivative on I. Then $Y_t = g(X_t)$ defines a regular diffusion process on the interval $I^* = [g(r), g(s)]$ and Y has parameters*

$$\mu_Y(y,t) = \frac{1}{2}\sigma^2(x,t)g''(x) + \mu(x,t)g'(x)$$

and

$$\sigma_Y^2(y,t) = \sigma^2(x,t)(g'(x))^2.$$

Remark 1.4.1 *These transformations of diffusion are subsumed in what is known as the Itô transformations formula, which will be discussed later.*

Proof See [Karlin & Taylor (1981].

1.4.2 Examples of diffusion processes.

Wiener processes or Brownian motion.

In 1828 the Scottish botanist Robert Brown observed that pollen grains suspended in liquid performed an irregular motion. The motion was later explained by the random collisions with the molecules of the liquid. To describe the motion mathematically it is natural to use the concept of a stochastic process $W_t(\omega)$, interpreted as the position at time t of the pollen grain ω.

The Wiener process was proposed by Wiener as mathematical description of Brownian motion. This physical process characterizes the erratic motion (i.e. diffusion) of a grain pollen on a water surface due to the fact that is continually bombarded by water molecules. The resulting motion can be viewed as a scaled random walk on any finite time interval and is almost surely continuous, with probability 1.

Definition 1.4.8 *A **standard Wiener process** is a continuous-time Gaussian Markov process $W = \{W_t\}_{t \geq 0}$ with stationary and independent increments which satisfies:*

1. *$W_0 = 0$ with probability 1,*
2. *W_t is almost surely continuous,*
3. *$W_t - W_s \sim N(0, t-s)$,*

for all $0 \leq s \leq t$.

It is a homogeneous Markov process because its transition probability is given by

$$p_{xy}(s,t) = \frac{1}{\sqrt{2\pi(t-s)}} \exp\left\{ \frac{-(y-x)^2}{2(t-s)} \right\}.$$

The standard Wiener process is also a diffusion process with drift $\mu(x,s) = 0$ and diffusion coefficient $\sigma^2(x,s) = 1$. Since $\Delta_h W = W_h - W_0$ is Normally distributed with mean zero and variance h, we have that

$$E(\Delta_h W | W_0 = x) = 0$$

and

$$E((\Delta_h W)^2 | W_0 = x) = h$$

Hence, the Kolmogorov forward and backward equations are given by

$$\frac{\partial p}{\partial t} - \frac{1}{2} \frac{\partial^2 p}{\partial y^2} = 0 \tag{1.17}$$

and

$$\frac{\partial p}{\partial s} + \frac{1}{2} \frac{\partial^2 p}{\partial x^2} = 0. \tag{1.18}$$

Remark 1.4.2 *The unique transition probability density function satisfying (1.17) and (1.18) together with the appropriate initial condition at $t = 0$ is the Gauss kernel*

$$\phi(t,x,y) = \frac{1}{\sqrt{2\pi t}} \exp\left\{ -\frac{(y-x)^2}{2t} \right\}$$

for $t > 0$ and $-\infty < x, y < \infty$.

We note that the Wiener process $W = \{W_t\}_{t \geq 0}$ is a martingale. Since

$$E(W_t - W_s | W_s) = 0,$$

and

$$E(W_s|W_s) = W_s,$$

with probability 1, we have

$$E(W_t|W_s) = W_s$$

with probability 1.

Although the sample paths of Wiener processes are almost surely nowhere differentiable. The continuity question of Brownian motion can be answered by using another famous Theorem of Kolmogorov.

Theorem 1.4.4 ***(Kolmogorov's continuity Theorem.)*** *Suppose that the process $X = \{X_t\}_{t\geq 0}$ satisfies the following condition: For all $T > 0$ there exist positive constants a, b, c such that*

$$E[|X_t - X_s|^a] \leq c|t-s|^{1+b},$$

for $0 \leq s, t \leq T$, then there exists a continuous version of X.

Proof. See [Øksendal (1998].

For Wiener process W we have that

$$E[|W_t - W_s|^4 | W_s = x] \leq 3|t-s|^2.$$

So Wiener process satisfies Kolmogorov's condition of Theorem 1.4.4 with $a = 4, c = 3$ and $b = 1$, and therefore it has a continuous version.

Definition 1.4.9 *A Wiener process with drift μ and variance σ^2 is a process $Z = \{Z\}_{t\geq 0}$ taking the form*

$$Z_t = \mu t + \sigma W_t$$

for $t \geq 0$. Note that

$$Z_t \sim N(\mu t, \sigma^2 t).$$

Wiener measure

Definition 1.4.10 *The* ***space of Wiener****, $W(\mathbb{R})$, is the set of all continuous paths $w : [0,\infty] \to \mathbb{R}$ which satisfy $w(0) = 0$.*

Definition 1.4.11 *Let $E = \{W_t\}_{t\geq 0}$ be a Wiener process, the distribution measure $\mathbb{P}$ induced by $W(\mathbb{R})$ is the* ***Wiener measure****.*

Ornstein-Uhlenbeck Process.

This diffusion process is defined on $I = (-\infty, \infty)$ and that has $\mu(x,t) = -\alpha x$, $\sigma^2(x,t) = \sigma^2$ as parameters, where α and σ^2 are positive constants. The drift parameter reflects a restoring force direted towards the origin and of a magnitude proportional to the distance.

The Ornstein-Uhlenbeck process is an alternative model of the velocity of a particle as a function of time. There are two factors that affect the velocity, the frictional resistance of the surrounding medium and the random collisions with neighboring particles. These factors be specifications by $\mu(x,t) = -\alpha x$, $\sigma^2(x,t) = \sigma^2$ respectively.

If $\{X_t\}_{t\geq 0}$ is an Ornstein-Uhlenbeck process with parameters $\mu(x,t) = -\alpha x$, $\sigma^2(x,t) = \sigma^2$, then conditioned on $X_t = x$, the distribution of X_{t+s} is Normal with mean

$$E(X_{t+s}|X_t = x) = xe^{-\alpha s}$$

and variance

$$var(X_{t+s}|X_t = x) = \frac{(1-e^{-2\alpha s})\sigma^2}{2\alpha}.$$

The associated backward equation corresponding to the parameters $\mu(x,t) = -\alpha x$ and $\sigma^2(x,t) = \sigma^2$ is

$$\frac{\partial p}{\partial t} = \frac{1}{2}\sigma^2\frac{\partial^2 p}{\partial x^2} - \alpha x\frac{\partial p}{\partial x}, \tag{1.19}$$

for $t > 0$ and $-\infty < x < \infty$.

Remark 1.4.3 *The unique transition probability density function satisfying (1.19) is*

$$p_{x,y}(t) = \phi\left(\frac{\sigma^2(1-e^{-2\alpha t})}{2\alpha}, xe^{-\alpha t}, y)\right)$$

for $t > 0$ and $-\infty < x, y < \infty$.

Identify the parameters. We abbreviate $\tau = \frac{\sigma^2(e^{2\alpha t}-1)}{2\alpha}$ and since the Ornstein-Uhlenbeck process $\{X_t\}_{t\geq 0}$ can be realized from standard Wiener process $\{B_t\}$ through the representation

$$X_t = e^{-\alpha t}B_\tau$$

with the appropriate change of the time clock and rescaling of the state variable. Then we have

$$E(X_{t+h} - X_t|X_t = x)$$

$$= e^{-\alpha t}\left\{E\left[e^{-\alpha h}B\left(\frac{\sigma^2(e^{2\alpha t}e^{2\alpha h}-1)}{2\alpha}\right) - xe^{\alpha t}|B_\tau = xe^{\alpha t}\right]\right\}$$

$$= e^{-\alpha t}(e^{-\alpha h - 1}xe^{\alpha t} = -\alpha xh + o(h).$$

Moreover we have that the Ornstein-Uhlenbeck process $\{X_t\}_{t\geq 0}$ is a Gaussian process, i.e., the finite-dimensional distribution is a multivariate Normal with mean zero and covariance kernel

$$E(X_tX_s) = \sigma^2 e^{-\alpha(t-s)}\left\{\frac{e^{2\alpha s}-1}{2\alpha}\right\}$$

for $s < t$.

Proposition 1.4.1 *For Ornstein-Uhlenbeck process $\{X_t\}_{t\geq 0}$ with parameters $\mu(x) = -\alpha x$ and $\sigma^2(x) = \sigma^2$*

$$\lim_{t\to\infty} P(X_t \leq y) = P(X_\infty \leq y),$$

where X_∞ is normally distributed with mean zero and variance $\sigma^2/2\alpha$.

Geometric Brownian Motion.

Let $\{X_t\}_{t\geq 0}$ be a Brownian motion process with drift μ and diffusion σ^2, then the process defined by $Y_t = e^{X_t}$ is called geometric Brownian motion. This process has state space $(0, \infty)$.

We use the Theorem 1.4.3 and hence the parameters for geometric Brownian motion are

$$\mu_Y(y) = (\mu + \frac{1}{2}\sigma^2)y$$

and

$$\sigma_Y^2(y) = \sigma^2 y^2.$$

Geometric Brownian motion is used to model prices of assets, say, shares of stock, that are traded in a perfect market.

The Bessel Process.

Let

$$Z_t = (X_t^{(1)})^2 + \cdots + (X_t^{(n)})^2$$

where $\{X_t^{(i)}\}$ are independent standard Brownian motion processes. The Z_t,$t \geq 0$ process is a diffusion with parameters $\mu(z,t) = n$ and $\sigma^2(z,t) = 4z$.

The Bessel process is $Y_t = \sqrt{Z_t}$, then apply Theorem 1.4.3 to obtain the parameters

$$\mu_Y(y) = \frac{n-1}{2y}$$

and

$$\sigma_Y^2(y) = 1.$$

1.4.3 Stationary Distribution

A stationary density $\psi(x)$ satisfies

$$\psi(y) = \int \psi(x) p_{x,y}(t) dx$$

for all $t > 0$.

Mimicking the derivation of Kolmogorov equations we have that $\psi(y)$ satisfies

$$0 = \frac{1}{2}\frac{\partial^2}{\partial y^2}[\sigma^2(y,t)\psi(y)] - \frac{\partial}{\partial y}[\mu(y,t)\psi(y)]. \tag{1.20}$$

Moreover by analogy with the fundamental limit Theorem of Markov chains we have that the stationary density is approached to the extent

$$\lim_{t\to\infty} p_{x,y}(t) = \psi(y)$$

the existence of this limit with $\psi(y)$ representing a bona fide probability density on the state space (r, s) implies that the process is positive ergodic.

Calculating the Stationary Distribution

We take (1.20) and integrating gives

$$\frac{d}{dy}\left[\frac{\sigma^2(y,t)}{2}\psi(y)\right] - \mu(y,t)\psi(y) = C_1$$

where C_1 is a constant. Next, taking

$$k(y) = exp\left\{-\int^y \left[\frac{2\mu(z,t)}{\sigma^2(z,t)}\right] dz\right\}$$

then

$$\frac{d}{dy}\left[k(y)\sigma^2(y,t)\psi(y)\right] = C_1 k(y).$$

integrating this gives

$$\psi(x) = m(x)[C_1 K(x) + C_2] \tag{1.21}$$

Here the constants are determined to guarantee the constraints

$$\psi(x) \geq 0$$

on (r, s) and

$$\int_r^s \psi(y)dy = 1,$$

the stationary density exists only if this is possible.

Example. For the Ornstein-Uhlenbeck process (1.21) gives

$$\psi(x) = C_1\left(\int_0^x e^{\alpha z^2/\sigma^2} dz\right) e^{-\alpha x^2/\sigma^2} + C_2 e^{-\alpha x^2/\sigma^2}.$$

To ensure that $\psi(x)$ is positive entails $C_1 = 0$, then the unique stationary measure based in (1.21) is the normal density

$$\psi(x) = ce^{-\alpha x^2/\sigma^2}.$$

1.4.4 Semigroup of Diffusion processes

Let $\{X_t\}_{t\geq 0}$ be a regular time homogeneous diffusion process on the open interval $I = (l, r)$. We denote by

$$P_{x,y} = P(X_t \leq y | X_0 = x)$$

the transition distribution function of X_t subject to initial distribution distribution

$$P_{x,y}(0) = \left\{ \begin{array}{ll} 1 & \text{if } x \leq y \\ 0 & \quad x > y \end{array} \right.$$

We assume that $P_{x,y}(t)$ derives from a continuous density on (r, s), namely,

$$\frac{P_{x,y}(t)}{dy} = p_{x,y}(t)$$

for $t > 0$.

Consider the family of operators $\{U_t\}_{t \geq 0}$ that transform each bounded continuous function f on $I = (l, r)$ into the function $U_t f$ by the formula

$$(U_t f)(x) = E_x(f(X_t)) = E(f(X_t)|X_0 = x)$$

We have that this function is jointly continuous with respect to $t > 0$ and x in the interval I provided f is piecewise continuous and bounded on I.

This operator satisfies the Fokker-Plank equation or Chapman-Kolmogorov equation (semigroup property)

$$U_{t+s} f = U_t(U_s f) = U_s(U_t f)$$

for all $t, s > 0$.

We now define the associated resolvent operators R_λ of the process by

$$(R_\lambda f)(x) = \int_0^\infty e^{-\lambda t}(U_t f)(x) dt \tag{1.22}$$

with $\lambda > 0$ and f is a bounded piecewise continuous function on (l, r). We can write the resolvent as a kernel operator

$$(R_\lambda f)(x) = \int_l^r G_\lambda(x, y) f(y) dy$$

where

$$G_\lambda(x, y) = \int_0^\infty e^{-\lambda t} p_t(x, y) dt.$$

We observe that $U_t \to I$, the identity operator as $t \downarrow 0$. We define A as

$$\lim_{h \downarrow 0} \left[\frac{U_h - I}{h} \right] = A.$$

The problem of characterization of the domain of A no easy.

Example. Standard Brownian Motion. Let $f(x)$ be a bounded and continuous on $(-\infty, \infty)$, we have

$$(U_t f)(x) = \frac{1}{\sqrt{2\pi t}} \int_{-\infty}^\infty e^{-(x-y)^2/2t} f(y) dy.$$

The resolvent kernel is

$$G_\lambda(x,y) = \frac{1}{\sqrt{2\lambda}} e^{-\sqrt{2\lambda}|x-y|}$$

for $\lambda > 0$.

The domain for infinitesimal operator A is the set of all bounded continuous functions with second derivatives which are themselves bounded and continuous and vanishing as $|x| \to \infty$. We have that for those functions

$$(Af)(x) = \frac{1}{2} f''(x).$$

for $-\infty < x < \infty$.

Dynkin Formula.

Let $u = R_\alpha f$ and τ be a Markov time. Then

$$u(x) = E_x \left\{ \int_0^\tau e^{-\alpha t} f(X_t) dt \right\} + E_x(e^{-\alpha\tau} u(X_\tau))$$

Theorem. Assume τ is a Markov time with finite expectation and $u(x) \in D(A)$ (domain of A). Then

$$E_x \left\{ \int_0^\tau Au(X_t) dt \right\} = E_x(u(X_\tau) - u(x)).$$

1.4.5 Differential Equations

In this section we assume that $\{X_t\}_{t\geq 0}$ is a time homogeneous diffusion process satisfying the following conditions:

1. The state of space is an intervale I of the form $[r,s], [r,s), (r,s]$ or (r,s) where $-\infty \leq r < s \leq \infty$.
2. The process is regular in the interior of I.
3. The process has infinitesimal parameters $\mu(x)$ and $\sigma^2(x)$, for $r < x < s$.
4. The infinitesimal parameters $\mu(x)$ and $\sigma^2(x)$ are contours functions of x and $\sigma^2(x) > 0$ for $r < x < s$.

Let a and b be fixed with $r < a < b < s$ and let T_y be the hitting time of y, then we denote

$$T^* = T_{a,b} = min\{T_a, T_b\}$$

as the first time that the process reaches either a or b.

Now we concentrate on three problems:

1. Find
$$u(x) = P(T_b < T_b | X_0 = x)$$
with $a < x < b$, the probability that the process reaches b before a.

2. Find
$$v(x) = E(T^*|X_0 = x)$$
with $a < x < b$, the mean time to reach a or b.

3. Find
$$w(x) = E(\int_0^{T^*} g(X_s)ds|X_0 = x)$$
with $a < x < b$ and g a bounded and continuous function.

Considering the first problem we choose a time duration h sufficiently small that the probability of reaching a or b before time h is negligible.

We now conditioner the process at time h, then by law of total probabilities we have
$$u(x) = E(u(X_h)|X_0 = x) + o(h).$$
Now write $\Delta X = X_h - x$ and assume we can expand in a Taylor series we have
$$u(x) = u(x) + \mu(x)hu'(x) + \frac{1}{2}\sigma^2(x)hu''(x) + o(h),$$
the we have that $u(x)$ satisfy the equation
$$0 = \mu(x)u'(x) + \frac{1}{2}\sigma^2(x)u''(x) \tag{1.23}$$
with $u(a) = 0$ and $u(b) = 1$.

Again, we choose a short time duration h as before and at time h we condition on $X_h = z$ and then we have
$$E(\int_h^{T^*} g(X(t)dt|X_h = z) = E(\int_0^{T^*} g(X(t)dt|X_0 = z) = w(z)$$
the first equality by the Markov property and stationary. Then we have
$$w(x) = E(\int_0^h g(X(t)dt|X_0 = x) + E(w(X_h)|X_0 = x) \tag{1.24}$$
where
$$E(\int_0^h g(X(t)dt|X_0 = x) = g(x)h + o(h)$$
because the process has continuous sample paths and g are continuous.

We now take a expand in a Taylor serie, so that equation (1.24) becomes
$$w(x) = g(x)h + w(x) + \mu(x)w'(x)h + \frac{1}{2}\sigma^2(x)w''(x)h + o(h)$$
the we have that $w(x)$ satisfy the equation
$$g(x) = \mu(x)w'(x) + \frac{1}{2}\sigma^2(x)w''(x), \tag{1.25}$$

with $w(a) = w(b) = 0$.

Following the idea we have

$$-1 = \mu(x)v'(x) + \frac{1}{2}\sigma^2(x)v''(x), \tag{1.26}$$

with $v(a) = v(b) = 0$.

Now let

$$k(x) = exp\left(-\int^x \frac{2\mu(t)}{\sigma^2(t)}dt\right) \tag{1.27}$$

for $r < x < s$.

We now introduce the **scale function** of the process

$$K(x) = \int^x k(\tau)d\tau = \int^x exp\left(-\int^x \frac{2\mu(t)}{\sigma^2(t)}dt\right)d\tau \tag{1.28}$$

and **speed density**

$$m(x) = \frac{1}{\sigma^2(x)s(x)} \tag{1.29}$$

for $r < x < s$.

We have that the equations (1.23), (1.25) and (1.26) each involve the differential operator L defined by

$$Lf(x) = \mu(x)f'(x) + \frac{1}{2}\sigma^2(x)f''(x),$$

with $f(x)$ twice continuously differentiable function on (a, b).

If we use $l(x) = 1/s(x)$ as an integrating factor and we separate variables, we have

$$Lf(x) = \frac{1}{2}\frac{d}{dM}\left(\frac{df(x)}{dK}\right) \tag{1.30}$$

where $k(x)$ and $m(x)$ are written in a differential given by

$$dK = k(x)dx$$

and

$$dM = m(x)dx.$$

Then the solution of these equations follows from two successive integrations.

The canonical representation of the Problem 1 is given by

$$\frac{1}{2}\frac{d}{dM}\left(\frac{df(x)}{dK}\right) = 0$$

subject to the boundary conditions

$$u(a) = 0$$

and

$$u(b) = 1,$$

then the solution is

$$u(x) = \frac{K(x) - K(a)}{K(b) - K(a)} \tag{1.31}$$

for $a \leq x \leq b$.

Remark 1.4.4 *Since K is strictly monotone and twice continuous differentiable, if we define $Y(t) = K(X(t))$ on the interval $(K(r), K(s))$, then the Theorem 1.4.3 establishes that the infinitesimal parameters of the $\{Y_t\}$ process are*

$$\mu_Y(y) = \frac{1}{2}\sigma(x)K''(x) + \mu(x)K'(x) = 0$$

and

$$\sigma_Y(y) = \sigma(x)(K'(x))^2 = \sigma(x)k^2(x).$$

A process Y whose scale function is linear is said to be in natural o canonical scale because the probabilities

$$P(T_b < T_b | X_0 = x) = \frac{b - y}{b - a}$$

are manifestly proportional to actual distances for $a < y < b$.

We now proceed to Problem 3. In the canonical representation, the differential equation is written

$$\frac{1}{2}\frac{d}{dM}\left(\frac{df(x)}{dK}\right) = -g(x)$$

for $a < x < b$ and subject to the boundary conditions

$$w(a) = w(b) = 0,$$

we get that the solution is

$$w(x) = 2\left(u(x)\int_x^b [K(b) - K(t)]m(t)g(t)dt + [1 - u(x)]\int_a^x [K(t) - K(a)]m(t)g(t)dt\right) \tag{1.32}$$

For the problem 3 we have the case $g(x) = 1$ and we have that the solution is

$$v(x) = 2\left(u(x)\int_x^b [K(b) - K(t)]m(t)dt + [1 - u(x)]\int_a^x [K(t) - K(a)]m(t)dt\right) \tag{1.33}$$

Examples of the Functional Calculations

Standard Brownian Motion

Let $\{X_t\}_{t\geq 0}$ be standard Brownian motion, then we have

$$k(x) = 1$$

and the scale function is

$$K(x) = 1$$

then

$$u(x) = \frac{x-a}{b-a} \tag{1.34}$$

for $x \in [a, b]$. The speed density is

$$m(\tau) = 1$$

and the Green function for the interval $[a, b]$ is

$$G(x,\tau) = \begin{cases} \frac{2(x-a)(b-\tau)}{(b-a)} & a \leq x \leq \tau \leq b \\ \\ \frac{2[(b-x)(\tau-a)}{(b-a)} & a \leq \tau \leq x \leq b \end{cases}$$

using equation (1.33) we have

$$v(x) = (x-a)(b-x) \tag{1.35}$$

for $x \in [a, b]$.

Brownian Motion with Drift

Let $\{X_t\}_{t\geq 0}$ be Brownian motion with nonzero drift $\mu(x) = \mu$ and diffusion σ^2, then

$$k(x) = exp\left(\frac{-2\mu x}{\sigma^2}\right),$$

$$K(x) = Aexp\left(\frac{-2\mu x}{\sigma^2}\right) + B$$

and

$$u(x) = \frac{e^{-2\mu x/\sigma^2} - e^{-2\mu a/\sigma^2}}{e^{-2\mu b/\sigma^2} - e^{-2\mu a/\sigma^2}}$$

for A and B constants.

Conditioned Diffusion Processes

Let $\{X_t\}_{t\geq 0}$ be a regular diffusion process with state space $[0, 1]$ and infinitesimal parameters $\mu(x, t)$ and $\sigma^2(x, t)$, we assume that 0 and 1 are the exit boundaries. We concentrate only in realizations of the process that lead to absorption at 1. Let $\{X_t^*\}_{t\geq 0}$ be the process confined to the sampler paths involving ultimate absorption at 1 which is a diffusion process.

Assume that the boundaries are attainable in finite expected time, i.e.,

$$E_x(T_{0,1}) < \infty.$$

Moreover we have that the functions $K(x)$ and $k(x)$ are finite. Then we have that infinitesimal parameters of the process $\{X_t^*\}_{t\geq 0}$ are

$$\mu^*(x) = \mu(x) + \frac{k(x)}{K(x)}\sigma(x) \tag{1.36}$$

and

$$\sigma^*(x) = \sigma(x). \tag{1.37}$$

1.4.6 Stochastic Differential Equations

The ordinary differential equation $dx/dt = \mu(x,t)$ can be viewed as a degenerate form a stochastic differential equation as no randomness is involved. It can be written in symbolic differential form

$$dx = \mu(x,t)dt \tag{1.38}$$

or as an integral equation

$$x(t) = x_0 + \int_{t_0}^{t} \mu(x_s, s)ds \tag{1.39}$$

where the solution satisfying the initial condition $x_0 = x_{t_0}$. For some regularity conditions on μ, this solution is unique, which means that the future is completely defined by the present given the initial condition.

If we consider that the motion of a particle in a fluid is influenced by two forces. One, a nonrandom motion can be engendered by the nature of the underling fluid flow and another, the general interaction relationships with other particles cause random movements which over short time duration are well described by Brownian motion.

We consider a small duration time t to $t + \Delta t$, then the displacement of the particle is approximated by

$$X_{t+\Delta t} - X_t \approx \mu(x,t)\Delta t + \sigma(x,t)\Delta W_t, \tag{1.40}$$

where $X_t = x$ is the location of particle at time t,$\mu(x,t)$ is the instantaneous velocity of the fluid at time t and the position x, ΔW_t is the incremental change associated with the standard Wiener process W_t and $\sigma^2(x,t) > 0$ is the instantaneous variance associated with the coalitions of the X_t process.

We can infer that X_t is a continuous Markov process and that $\mu(x,t)$ and $\sigma(x,t)$ are appropriately continuous deterministic functions, then from equation (1.40) we infer that X_t constitutes a diffusion process with drift coefficient $\mu(x,t)$ and diffusion coefficient $\sigma^2(x,t)$.

To assign meaning to the limit of the relation equation (1.40) we make

$$\frac{dX_t}{dt} = \mu(x,t) + \sigma(x,t)\frac{dW}{dt}, \tag{1.41}$$

it is necessary to developed an extended version of stochastic differentials equations. The equation (1.41) is preferably written in the differential notation Stochastic Differential Equation (SDE)

$$dX_t = \mu(X_t,t)dt + \sigma(X_t,t)dW_t \tag{1.42}$$

or as an integral equation

$$X_t(\omega) = X_{t_0}(\omega) + \int_{t_0}^{t} \mu(X_s(\omega), s)ds + + \int_{t_0}^{t} \sigma(X_s(\omega), s)dW_t \tag{1.43}$$

and we usually model systems by stochastic differential equations of the form of equation (1.42).

The problem with this formulation is that the Wiener process W_t is (almost surely) nowhere differentiable. As a result, the second integral in equation (1.43) cannot be understood as an ordinary (Riemann or Lebesgue) integral. Worse, it is not a Riemann-Stieltjes integral since the continuous sample paths of a Wiener process are not of bounded variation for each sample path. Hence, it is at this point that *Itô*'s stochastic integral comes into play!

1.4.7 *Itô* Stochastic Integral

In this section, we consider a probability space $(\Omega, \mathscr{F}, P)$, a Wiener process $W = \{W_t\}_{t\geq 0}$ and an increasing family $\{\mathscr{F}_t\}_{t\geq 0}$ of sub-σ-algebras of $\mathscr{F}$ such that W_t is $\mathscr{F}_t$-measurable for each $t \geq 0$ and with

$$E(W_t|\mathscr{F}_0) = 0$$

and

$$E(W_t - W_s|\mathscr{F}_s) = 0,$$

with probability one, for $0 \leq s \leq t$.

We have that for constant $\sigma(x,t) = \sigma$ the second integral in (1.43) is expected to be equal to $\bar{\sigma}[W_t(\omega) - W_{t_0}(\omega)]$.

We consider the integral of the random function $f : \Omega \times T \to \mathbb{R}$ on the unit time interval:

$$I[f](\omega) = \int_0^1 f(\omega, s) dW_s(\omega). \tag{1.44}$$

We have the following remarks about equation (1.44)

1. If the function f is a nonrandom step function, that is $f(\omega,t) = f_j(\omega)$ on $t_j \leq t < t_{j+1}$ for $j = 1, 2, \ldots, n-1$ with $0 = t_1 < t_2 < \ldots t < t_n = 1$, then we have

   $$I[f](\omega) = \sum_{j=1}^{n-1} f_j(\omega)(W_{t_{j+1}}(\omega) - W_{t_j}(\omega))$$

   with probability one. Note that this integral is a random variable with zero mean as it is a sum of random variables with zero mean. Furthermore, we have the following result

   $$E(I[f](\omega)) = \sum_{j=1}^{n-1} f_j^2(t_{j+1} - t_j).$$

2. If the function f is a random step function, that is $f(\omega,t) = f_j(\omega)$ on $t_j \leq t < t_{j+1}$ for $j = 1, 2, \ldots, n-1$ with $t_1 < t_2 < \ldots < t_n$ and the function is $\mathscr{F}_{t_j}$-measurable and mean square integrable over Ω, that is

   $$E(f_j) < \infty$$

for $j = 1, 2, \ldots, n$. The stochastic integral $I[f](\omega)$ is defined as follows:

$$I[f](\omega) = \sum_{j=1}^{n-1} f_j(\omega)(W_{t_{j+1}}(\omega) - W_{t_j}(\omega)) \tag{1.45}$$

with probability one.

Lemma 1.4.2 *For any $a, b \in \mathbb{R}$ and any random step functions f, g such that f_j, g_j on $t_j \le t < t_{j+1}$ for $j = 1, 2, \ldots, n-1$ with $0 = t_1 < t_2 < \ldots t < t_n = 1$, is $\mathscr{F}_{t_j}$-measurable and mean square integrable, the stochastic integral equation (1.45) satisfies the following properties:*

(a) $I[f]$ is $\mathscr{F}_1$-measurable,

(b) $E(I[f]) = 0$,

(c) $E(I^2[f]) = \sum_j E(f_j^2)(t_{j+1} - t_j)$,

(d) $I[af + bg] = aI[f] + bI[g]$, with probability one.

Proof. Since f_j is $\mathscr{F}_{t_j}$-measurable and $W_{t_{j+1}} - W_{t_j}$ is $\mathscr{F}_{t_{j+1}}$-measurable, each term $f_j\{W_{t_{j+1}} - W_{t_j}\}$ is $\mathscr{F}_{t_{j+1}}$-measurable and thus $\mathscr{F}_1$-measurable. Hence, $I[f]$ is $\mathscr{F}_1$-measurable.

From the Cauchy-Schwarz inequality and the fact that each term in equation (1.45) is mean square integrable, it follows that $I[f]$ is integrable. Hence, $I[f](\omega)$ is again a zero mean random variable:

$$E(I[f]) = \sum_{j=1}^{n-1} E(f_j\{W_{t_{j+1}} - W_{t_j}\}) = \sum_{j=1}^{n-1} E(f_j E\{W_{t_{j+1}} - W_{t_j}\}|\mathscr{F}_{t_j}) = 0$$

Furthermore, $I[f]$ is mean square integrable:

$$E(I^2[f]) = \sum_{j=1}^{n-1} E(f_j^2)E(\{W_{t_{j+1}} - W_{t_j}\}^2|\mathscr{F}_{t_j}) = \sum_{j=1}^{n-1} E(f_j^2)(t_{j+1} - t_j).$$

And, $af + bg$ is a step random step function for any $a, b \in \mathbb{R}$. Therefore, we obtain (d), with probability 1. ■

3. If the (continuous) function f is a general integrand such that $f_t(*)$ is $\mathscr{F}_t$-measurable and mean square integrable, then we define the stochastic $I[f]$ integral as the limit of integrals $I[f^{(n)}]$ of random step functions $f^{(n)}$ converging to f. The problem is thus to characterize the limit of the following finite sums:

$$I[f^{(n)}](\omega) = \sum_{j=1}^{n-1} f_{t_j}^{(n)}(\omega)(W_{t_{j+1}}(\omega) - W_{t_j}(\omega))$$

with probability one, on $t_j \le t < t_{j+1}$ for $j = 1, 2, \ldots, n-1$ with $t_1 < t_2 < \ldots < t_n$. from (c). we get

$$E(I^2[f^{(n)}]) = \sum_{j=1}^{n-1} E((f_{t_j}^{(n)}(*))^2)(t_{j+1} - t_j).$$

This converges to the Riemann integral $\int_0^1 E(f^2(s,*))ds$ for $n \to \infty$. This result, along with the well-behaved mean square property of the Wiener process, i.e., $E\{(W_t - W_s)^2\} = t - s$, suggests defining the stochastic integral in terms of mean square convergence.

Theorem 1.4.5 *The Itô stochastic integral $I[f]$ of a function $f : \Omega \times T \to \mathbb{R}$ is the unique mean square limit of sequences $I[f^{(n)}]$ for any sequence of random step functions $f^{(n)}$ converging to f:*

$$I[f](\omega) = \lim_{n\to\infty} \sum_{j=1}^{n-1} f_{t_j}^{(n)}(\omega)(W_{t_j+1}(\omega) - W_{t_j}(\omega)),$$

with probability one.

The properties (a),(b) and (c) from Lemma 1.4.2 still apply, but we write

$$E(I^2[f]) = \int_0^1 E(f^2(*,t))dt$$

for (d) and call it the *Itô* isometry on the unit time interval. Now, we present the *Itô* integral.

Definition 1.4.12 *The time-dependent Itô integral is a random variable defined on any interval $[t_0, t]$:*

$$X_t(\omega) = \int_{t_0}^t f(\omega, s)dWs(\omega) \tag{1.46}$$

which is $\mathscr{F}_t$-measurable and mean square integrable.

From the independence of non-overlapping increments of a Wiener process, we have

$$E(X_t - X_s|\mathscr{F}_s) = 0,$$

with probability 1, for any $t_0 \leq s \leq t$. Hence, the process X_t is a martingale.

As the Riemann and the Riemann-Stieltjes integrals, equation(1.46) satisfies conventional properties such as the linearity property and the additivity property.

However, it has also the unusual property that

$$\int_0^t W_s(\omega)dW_s(\omega) = \frac{1}{2}W_t^2(\omega) - \frac{1}{2}t, \tag{1.47}$$

with probability 1, where $W_0 = 0$, with probability 1.

Note that this expression follows from the fact that

$$\sum_j W_{t_j}(W_{t_{j+1}} - W_{t_j}) = \frac{1}{2}W_t^2(\omega) - \frac{1}{2}\sum_j (W_{t_{j+1}} - W_{t_j})^2,$$

where the second term converges to t in mean square sense.

The *Itô* Formula

A consequence is that stochastic differentials, which are interpreted as stochastic integrals, do not follow the chain rule of classical calculus.

Roughly speaking, an additional term is appearing due to the fact that dW_t^2 is equal to dt in the mean square sense.

We consider the stochastic process $Y = \{Y_t = U(X_t, t)\}_{t\geq 0}$ with $U(x,t)$ having continuous second order partial derivatives.

If X_t were continuously differentiable, the chain rule of classical calculus would give the following expression:

$$dY_t = \frac{\partial U}{\partial t}dt + \frac{\partial U}{\partial x}dXt. \tag{1.48}$$

This follows from a Taylor expansion of U in ΔY_t and discarding the second and higher order terms in Δt.

When X_t is a process of the form equation (1.46), we get

$$dY_t = (\frac{\partial U}{\partial t} + \frac{1}{2}f^2\frac{\partial^2 U}{\partial x})dt + \frac{\partial U}{\partial x}dXt. \tag{1.49}$$

where $dX_t = fdW_t$ is the symbolic differential form of equation (1.46).

The additional term is due to the fact that $E(dX_t^2) = E(f^2)dt$ gives rise to an additional term of the first order in Δt of the Taylor expansion for U :

$$\Delta Y_t = (\frac{\partial U}{\partial t}\Delta t + \frac{\partial U}{\partial x}\Delta x) + \frac{1}{2}(\frac{\partial^2 U}{\partial t^2}\Delta t^2 + \frac{\partial^2 U}{\partial t \partial x}\Delta x \Delta t + \frac{\partial^2 U}{\partial x^2}\Delta x^2) + \dots$$

Theorem 1.4.6 *Consider the following general stochastic differential:*

$$dX_t(\omega) = g(\omega, t)dt + f(\omega, t)dW_t(\omega)$$

*Let $Y_t = U(X_t, t)$ with U having second continuous partial derivatives, then $Y = \{Y_t\}$ is again a diffusion process and the Itô**'s formula** is the following stochastic chain rule:*

$$dY_t = \frac{\partial U}{\partial t}dt + \frac{\partial U}{\partial x}dX_t + \frac{1}{2}\frac{\partial^2 U}{\partial x^2}(dX_t)^2 \tag{1.50}$$

with probability one and where $(dX_t)^2 = (dX_t)(dX_t)$ is computed according to the rules

$$dtdt = dtdW_t = dW_tdt = 0,$$

*and $dWtdW_t = dt$, then the Itô **formula** is*

$$dY_t = \{\frac{\partial U}{\partial t} + \frac{1}{2}f^2(\omega, t)\frac{\partial^2 U}{\partial x^2}\}dt + \frac{\partial U}{\partial x}dX_t \tag{1.51}$$

Finally, from the *Itô* formula, one can recover equation (1.47). For $X_t = W_t$ and $u(x) = x^n$, we have

$$d(W_t^n) = nW_t^{(n-1)}dW_t + \frac{n(n-1)}{2}W_t^{(n-2)}dt.$$

If we consider the case $n = 2$ we have

$$d(W_t^2) = 2W_t dW_t + dt.$$

then, we have

$$W_t^2 - W_s^2 = 2\int_s^t W_t dWt + (t - s).$$

Taking $s = 0$, we recover equation (1.47).

The Lamperti transform.

Here, we present a particular application of the *Itô* formula that is of interest in many of the simulation and estimation methods of stochastic differential equations. We consider a diffusion process which is the solution of the equation

$$dX_t = \mu(X_t; t)dt + \sigma(X_t)dW_t,$$

where the diffusion coefficient depends only on the state variable.

We can always transform the stochastic differential equation into one with a unitary diffusion coefficient by applying the **Lamperti transform**, given by

$$Y_t = U(X_t, t) = \int_x^{X_t} \frac{1}{\sigma(u)}du, \tag{1.52}$$

with x any arbitrary value in the state space of X. Using the *Itô* formula we have

$$\frac{\partial U(x,t)}{\partial t} = 0,$$

$$\frac{\partial U(x,t)}{\partial x} = \frac{1}{\sigma(x)}$$

and

$$\frac{\partial^2 U(x,t)}{\partial x^2} = -\frac{\sigma'(x)}{\sigma^2(x)},$$

then the process Y_t solve the stochastic differential equation

$$dY_t = -\frac{1}{2}\sigma'(X_t)dt + \frac{1}{\sigma(X_t)}\mu(X_t; t)dt + dW_t$$

therefore

$$Y_t = \alpha(Y_t, t)dt + dW_t \tag{1.53}$$

where

$$\alpha(Y_t, t) = \frac{\mu(X_t, t)}{\sigma(X_t)} - \frac{1}{2}\sigma'(X_t).$$

1.4.8 The Girsanov Theorem.

Basically the Girsanov Theorem says that if we change the drift coefficient of a given diffusion process, then the law of the process will not change dramatically. In fact, the law of the new process will be absolutely continuous with respect to the law of the original process and we can compute explicitly the Radon-Nikodym derivative.

Theorem 1.4.7 ***The Girsanov Theorem.*** *Let $X = \{X_t\}_{t \geq 0}$ be an diffusion process on a probability space $(\Omega, \mathscr{F}, \mathbb{P})$ with values in $\mathbb{R}$ of the form*

$$dX_t = \mu(x,t)dt + dW_t$$

with $X_0 = 0$, and $t \leq T \leq \infty$ and W_t is one-dimentional Wiener process. Put

$$M_t = \exp\{-\int_0^t \mu(x,s)ds - \frac{1}{2}\int_0^t \mu^2(x,s)ds\} \tag{1.54}$$

and suppose that $\mu(x,t)$ satisfies the ***Novikov's condition***

$$E[\exp(\frac{1}{2}\int_0^T \mu^2(x,s)ds)] < \infty \tag{1.55}$$

where $E = E_{\mathbb{P}}$ is the expectation with respect to $\mathbb{P}$. Define the measure $\mathbb{Q}$ on $(\Omega, \mathscr{F}_T)$ by

$$d\mathbb{Q}(\omega) = M_T(\omega)d\mathbb{P}(\omega) \tag{1.56}$$

Then X is an one-dimensional Wiener process with respect to the probability law $\mathbb{Q}$, for $t \leq T$.

Remark 1.4.5 *The transformation $\mathbb{P} \to \mathbb{Q}$ given by equation (*1.56*) is called the* ***Girsanov transformation*** *of measures and since M_t is a martingale we actually have that*

$$M_T d\mathbb{P} = M_t d\mathbb{P} \tag{1.57}$$

on $\mathscr{F}_t, t \leq T$.

Proof. For a proof see [Øksendal (1998].

1.4.9 Numerical Methods for Stochastic Differential Equations.

In this section we present the principal methods for the simulation of solutions of stochastic differential equations. Simulation methods are usually based on discrete approximations of the continuous solution to a stochastic differential equation. The methods of approximation are classified according to their different properties. Mainly two criteria of optimality are used in the literature: the strong and the weak convergence.

Strong convergence

A time-discretized approximation X^δ of a continuous-time process X, with δ the maximum time increment of the discretization, is said to be of general strong order of convergence γ to X if

$$E[X_T^\delta - X_T] \leq C\delta^\gamma$$

is fixed for any time horizon T, for all $\delta < \delta_0$ with $\delta_0 > 0$ and C a constant not depending on δ.

Weak convergence

We again consider the time-discretized approximation X^δ, then X^δ is said to converge weakly of order β to X if

$$|Eg(X_T) - E(X_T^\delta)| \leq C\delta^\beta$$

is fixed for any time horizon T, any continuous $2(\beta+1)$ differentiable function g of polynomial growth, with $\delta_0 > 0$ and C a constant not depending on δ.

Euler approximation.

Consider a the diffusion process $X = \{X_t\}_{t\geq 0}$ solution of the stochastic differential equation

$$dX_t = \mu(X_t, t)dt + \sigma(X_t, t)dW_t \tag{1.58}$$

with initial deterministic value $X_{t_0} = x_0$ and a time discretization

$$0 = t_0 < t_1 < \ldots < t_n = T$$

of the interval $[0, T]$. The Euler approximation of X is a continuous stochastic process Y satisfying the iterative scheme

$$Y_{t_{i+1}} = Y_{t_i} + \mu(Y_{t_i}, t_i)\Delta_i + \sigma(Y_{t_i}, t_i)\Delta W_i \tag{1.59}$$

for $i = 0, 1, \ldots, n-1$, with $Y_0 = x_0$ and

$$\Delta_i = t_{i+1} - t_i$$

$$\Delta W_i = Wt_{i+1} - W_{t_i},$$

then $\Delta W_i \sim N(0, \Delta_i)$.

In between any two time points t_i and t_{i+1}, the process can be defined differently. One natural approach is to consider linear interpolation so that Y_t is defined as

$$Y_t = Y_{t_i} + \frac{t - t_i}{t_{i+1}}(Y_{t_{i+1}} - Y_{t_i})$$

for all $t \in [t_i, t_{i+1})$. The Euler scheme has order $\gamma = 1/2$ of strong convergence.

Euler scheme Example.

Here, we present a simulation of a diffusion process using Euler scheme, we consider the Ornstein-Uhlenbeck process, which is a solution of the stochastic differential equation

$$dX_t = -\alpha X_t dt + \sigma dW_t, \tag{1.60}$$

where $\alpha > 0$ and $\sigma > 0$ are unknown parameters to be estimated, and W is a standard Wiener process. We suppose that $X_0 = 0$ and considering an the interval of estimation $[0, 1]$ with a discretization

$$0 = t_0 < t_1 < \ldots < t_n = 1,$$

where $n = 100$ and $\Delta = \Delta_i = 0.01$, for all $i = 1, 2, \ldots, n-1$, then **Euler scheme** for this process is

$$Y_{t_{i+1}} = Y_{t_i} - \alpha Y_{t_i}\Delta + \sigma\Delta W$$

In the Figure 1.1, we present a realization of the process when $\alpha = 0.5$ and $\sigma = 2.5$ on the interval $[0, 1]$.

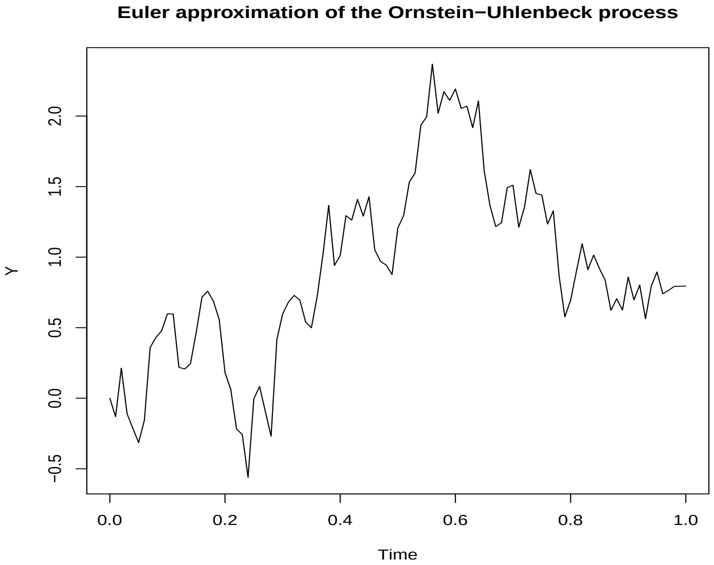


Figure 1.1: A simulated path of the Ornstein-Uhlenbeck process using a discretization of 100 subintervals of [0,10]

Milstein approximation.

The Milstein scheme makes use of $It\hat{o}$'s Lemma to increase the accuracy of the approximation by adding the second-order term, we can approximate X_t by the **Milstein scheme**

$$Y_{t_{i+1}} = Y_{t_i} + \mu(Y_{t_i}, t_i)\Delta_i + \sigma(Y_{t_i}, t_i)\Delta W_i + \frac{1}{2}\sigma(Y_{t_i}, t_i)\sigma'(Y_{t_i}, t_i)t_i[(\Delta W_i)^2 - \Delta_i] \tag{1.61}$$

This scheme has strong and weak orders of convergence equal to one.

Milstein scheme Example.

We present an Example of simulation of a diffusion process using Milstein scheme, we consider the geometric Brownian motion, which is a solution of the stochastic differential equation

$$dX_t = \theta_1 X_t d_t + \theta_2 X_t dW_t. \tag{1.62}$$

For this process, $\mu(x,t) = \theta_1 x, \sigma(x,t) = \theta_2 x$, and $\sigma'(x,t) = \theta_2$ and the Milstein scheme reads

$$\begin{aligned} Y_{t_{i+1}} &= Y_{t_i} + \theta_1 Y_{t_i}\Delta_i + \theta_2 \Delta Y_{t_i} W_i + \frac{1}{2}\theta_2^2 Y_{t_i}[(\Delta W_i)^2 - \Delta_i] \qquad (1.63) \\ &= Y_{t_i}(1 + \Delta_i(\theta_1 - \frac{\theta_2^2}{2})) + \theta_2 \Delta W_i + \frac{\theta_2^2}{2} Y_{t_i}(\Delta W_i)^2 \end{aligned}$$

We again suppose that $X_0 = 0$ and considering an the interval of estimation $[0,1]$ with a discretization

$$0 = t_0 < t_1 < \ldots < t_n = 1,$$

where $n = 100$ and $\Delta = \Delta_i = 0.1$, for all $i = 1, 2, \ldots, n-1$. In the Figure 1.2, we present a realization of the process when $\theta = (1.0, 0.5)$ on the interval $[0,1]$.

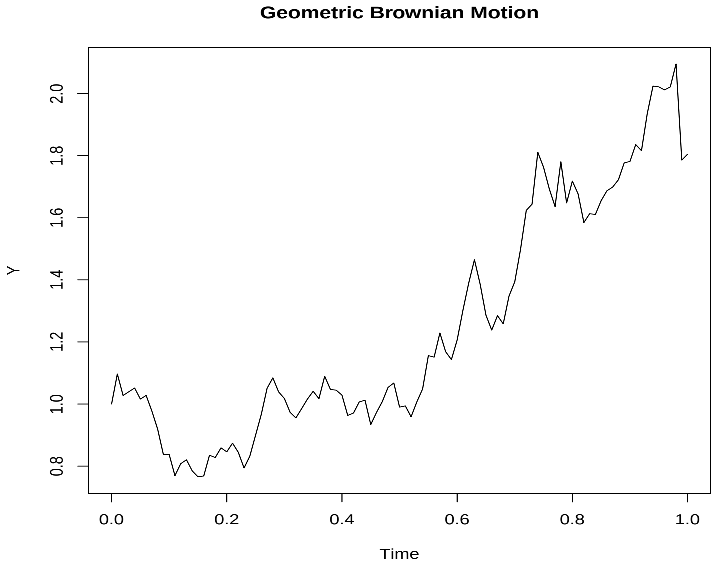


Figure 1.2: A simulated path of the Geometric Brownian motion using a discretization of 100 subintervals of [0,1]

1.5 Markovian Bridges.

In this section we consider Markov processes subject to constraints on their initial and ending points, i.e., Markov bridges.

Markov bridges are prototypical of a variety of conditioned diffusions arising in applications and hence it is of some interest to understand the behaviour of MCMC methods in this context. Furthermore, the corresponding target bridges are representative of the family of distributions defined as a change of measure from Gaussian laws on arbitrary separable Hilbert spaces.

Markov bridges plays an important role in some very useful approaches to likelihood inference (including Bayesian inference) for discretely sampled diffusion processes and other diffusion-type processes like stochastic volatility models.

1.5.1 Construction of Bridges

If we consider a Markov process $X = \{X_t\}_{t\geq 0}$ with state space E, then a Markovian bridge is a process obtained by conditioning a Markov process X to start in some state x at time t_0 and arrive at some state y at time $t > t_0$ and we call this process the (t_0, x, t, y)-bridge derived from X.

Let $X = \{X_t\}_{t\geq 0}$ be a Markov process with state space E, strong Markov property and transition semigroup $P(t)$. We assume that E is Lusinian [1], that $P(t)$ maps Borel functions to Borel functions, and that the paths of X are cadlag. This allows us to realize X as the coordinate process on the sample space Ω of all cadlag paths from $[0, \infty)$ to E . The law of X when started at x is $\mathbb{P}_x$. We write $\{\mathscr{F}_t\}_{t\geq 0}$ for the natural filtration of X and $\{\theta_t\}_{t\geq 0}$ for the usual shift operators i.e., $X_t \circ \theta_t = X_{s+t}$. The transition densities of X is given by

$$P_{x,dy}(t) = p_{xy}(t)\lambda(dy) \tag{1.64}$$

where λ is a σ-finite measure on E. We suppose that there is a second process $\hat{X}$ with strong Markov property in duality with X relative to the measure λ, i.e., the semigroup $\hat{P}(t)$ of $\hat{X}$ is related to $P(t)$ by

$$\int_E f(x)P(t)g(x)\lambda(dx) = \int_E \hat{P}(t)f(x)g(x)\lambda(dx) \tag{1.65}$$

for all $t > 0$ and all positive Borel functions f and g. Then, we have that equations (1.64) and (1.65) imply that there is a version of the density $p_{xy}(t)$ that is jointly measurable in (t, x, y) and such that the Chapman-Kolmogorov identity

$$p_{xy}(t+s) = \int_E p_{xz}(t)p_y(s)\lambda(dz) \tag{1.66}$$

holds for all $s, t > 0$, and $x, y \in E$. By the hypothesis of duality we have that

$$\hat{P}_{x,dy}(t) = p_{xy}(t)\lambda(dy). \tag{1.67}$$

Here, we present an outline of Doob's method of h-transforms to construct bridge laws $\mathbb{P}^s_{xy}$, for details see [P. Fitzsimmons & Yor (1992].

[1] homeomorphic to a Borel subspace of some compact metric space

Doob's Method

The idea is construct bridge laws $\mathbb{P}^s_{xy}$, which for each x and s will serve as a family of regular $\mathbb{P}_x$ conditional laws for $\{X_t\}_{0\leq t<s}$ given $X_{s^-}=y$, which will serve equally well as conditional laws given $X_s=y$.

Now, if we fix $x,y\in E$ and $s>0$ such that $0<p_{xy}(s)<\infty$ by Chapman-Kolmogorov we have that

$$H_t = p_{X_t y}(s-t)$$

is a (positive) martingale under $\mathbb{P}_x$ with $0\leq t<0$. Then we have that

$$Q(A)=\int_A H_t(\omega)\mathbb{P}_x(d\omega) \tag{1.68}$$

with $A\in\mathscr{F}$ defines a finitely additive set function $Q=Q^s_{xy}$ on the algebra $\mathscr{G}=\bigcup_{0\leq t<s}\mathscr{F}_t$ such that each restriction $Q|_{\mathscr{F}_t}$ is σ-additive and if we normalize by $p_{xy}(s)$, Q extends to a measure on $\mathscr{F}_{s^-}$ given by $\mathbb{P}^s_{xy}$. Then we have the next result

Proposition 1.5.1 *If it satisfies equations 1.64,1.65, 1.66 and* $0<p_{xy}<\infty$ *then there is a unique probability measure* $\mathbb{P}^s_{xy}$ *on* $(\Omega,\mathscr{F}_{s^-})$ *such that*

$$\mathbb{P}^s_{xy}(F)p_{xy}(s)=\mathbb{P}(Fp_{X_t y}(s-t)) \tag{1.69}$$

for all positive $\mathscr{F}_t$*-measurable functions* F *on* Ω*, for all* $0\leq t<s$ *. Under* $\mathbb{P}_{xy}$*, the coordinate process* $\{X_t\}_{0\leq t<s}$ *is a non-homogeneous strong Markov process with transition densities*

$$p^{(y,s)}(x,r;z,t)=\frac{p_{xz}(t-r)p_{zy}(s-t)}{p_{xy}(s-r)} \qquad 0<r<t<s. \tag{1.70}$$

Moreover, if $F\geq 0$ *is* $\mathscr{F}_t$*-measurable and* $g\geq 0$ *is a Borel function on* E*, then*

$$\mathbb{P}(F(g(X_{s^-})))=\int_E \mathbb{P}(F)g(y)p_{xy}(s)\lambda(dy). \tag{1.71}$$

Remark 1.5.1 *We have that* $\{\mathbb{P}^s_{xy}\}_{y\in E}$ *is a regular version of the family of conditional probability distributions* $\mathbb{P}_x(*|X_{s^-}=y)$*.*

For a proof see [P. Fitzsimmons & Yor (1992]. Now, we present some results of Proposition 1.5.1, which used in the simulation of bridges in the following section.

Corollary 1.5.1 *Suppose* $0<p_{xy}(s)<\infty$*, then*

1. *The law* $\mathbb{P}^s_{xy}$ *of the time-reversed process* $\{X_{(s-t)^-}\}_{0\leq t<s}$ *is* $\hat{\mathbb{P}}^s_{yx}$*, the law of a* $(0,y,s,x)$*-bridge for the dual process* $\hat{X}$*.*
2. *For each stopping time* τ *on* $\{\mathscr{F}_t\}$*, a* $\mathbb{P}^s_{xy}$ *regular conditional distribution for* $\{X_{\tau+r}\}_{0\leq r<s-\tau}$ *given* $\mathscr{F}_t$ *on* $\{\tau<l\}$ *is provided by* $\mathbb{P}^{s-\tau}_{X_\tau y}$

Finally, we will mention an important remark about of this results.

Remark 1.5.2 *Applied to the dual process after time reversal and conditioning on* X*, Proposition 1.5.1 implies the following decomposition of the original Markov process* X *at random times* τ *that correspond to stopping times on the reversed time scale.*

1.5.2 Simulation of Diffusion Bridges.

Algorithms for simulation of diffusion bridges were proposed by [Roberts & Stramer (2001]. [Beskos et al. (2006] developed algorithms for exact simulation of diffusion bridges when the drift and diffusion coefficients satisfy certain boundness conditions. Under strong boundness conditions the algorithm is relatively simple, whereas it is more involving under weaker condition.

Here, we will briefly outline of the method for simulation of diffusion bridges proposed by [Bladt & Sørensen (2009] which will be used later. Let $X = \{X_t\}_{t\geq 0}$ be a one-dimensional diffusion given by equation 1.42 where the coefficients are sufficiently regular to ensure that the equation has a unique weak solution that is a strong Markov process.

We consider two x and y points in the state space of X and suppose that we are interested in the realization of the process in the interval $[t_0, t_1]$ given that $X_{t_0} = x$ and $X_{t_1} = y$. We will call to a solution of equation 1.42 in the interval $[t_0, t_1]$ such that $X_{t_0} = x$ and $X_{t_1} = y$, a $(t_0, x, t_1, y)-$bridge.

Suppose that the state space is given by (l, r) and let W^1 and W^2 be two independent standard Wiener processes, and define X^1 and X^2 as the solutions to

$$dX_t^i = \mu(X_t^i, \psi)dt + \sigma(X_t^i, \psi)dW_t^i$$

for $i = 1, 2$ and $X_{t_0}^1 = x$ and $X_{t_0}^2 = y$, then the idea is to realize an approximation to $(t_0, x, t_1, y)-$bridge by simulating the process X^1 from a forward in time and X^2 from b backward in time starting at time one. If the samples paths of the two processes intersect, they can be combined into a realization of a process that approximates a $(t_0, x, t_1, y)-$bridge.

Let $Y_{\delta_i}^1$ and $Y_{\delta_i}^2$, $i = 0, 1, \ldots, N$ and be independent simulations of X^1 and X^2 in $[0, \Delta]$ with step size $\delta = \Delta/N$. Then a simulation of an approximation to a $(0, x, \Delta, y)-$bridge is obtained by the following rejection sampling scheme. Keep simulating Y^1 and Y^2 until the sample paths cross. Once a trajectory crossing has been obtained, define

$$B_{\delta i} = \begin{cases} Y_{\delta i}^1 & \text{for } \ i = 1, \ldots, \tau \\ Y_{\delta i}^2 & \text{for } \ i = \tau, \ldots, N \end{cases}$$

where $\tau = \min\{i \in 1, \ldots, N | Y_{\delta i}^1 \leq Y_{\delta(N-i)}^2\}$ if $Y_0^1 \geq Y_\Delta^2$, and $\tau = \min\{i \in \{1, \ldots, N\} | Y_{\delta i}^1 \geq Y_{\delta(N-i)}^2\}$ if $Y_0^1 \leq Y_\Delta^2$ and then B approximates a $(0, x, \Delta, y)-$bridge.

Consider the diffusion bridge in the interval $[0, 1]$ we have that the distribution of the process that is simulated by the algorithm presented above and the sense in which it is an approximation of a diffusion bridge is seen in the following Theorem.

Theorem 1.5.1 *Let $\tau = inf\{0 \leq t \leq 1 | X_t^1 = x_{1-t}^2\}$. and*

$$Z_t = \begin{cases} X_{\delta i}^1 & \textit{if } \ 0 \leq t \leq \tau \\ X_{\delta i}^2 & \textit{for } \ \tau < t \leq 1 \end{cases}$$

Then the distribution of $Z = \{Z_t\}_{0 \le t \le 1}$ conditional on the event $\{\tau \le 1\}$ equals the distributions of a $(0, x, 1, y)-$bridge conditional on the event that the bridge is hit by an independent diffusion with stochastic differential equation 1.42 and initial distribution with density $p_{y,*}(1)$.

For a proof see [Bladt & Sørensen (2009].

Remark 1.5.3 *We obviously have that the quality of the approximation depends on the probability π that a $(0, x, 1, y)$-bridge is hit by an independent diffusion with initial distribution $p_1(y, *)$. When π is close to one, the simulated process is essentially a $(0, x, 1, y)$-bridge.*

Now, since that the speed measure $s(x)$ for any one-dimensional diffusion satisfies the balance equation

$$p_{xy}(t)s(x) = p_{yx}(t)s(y) \tag{1.72}$$

see [Ito & McKean (1965] we have followin result.

Lemma 1.5.1 *The time-reversed process $\{\hat{X}_t\}_{t \ge 0}$ given by $\hat{X}_t = X^2_{1-t}$. The process $\{\hat{X}_t\}$ and the conditional process $\{X_t\}_{t \ge 0}$ given that $X_1 = y$ have the same transition densities and the distribution of $\{\hat{X}_t\}_{t \ge 0}$ is equal to the conditional distribution of the process $\{X_t\}_{t \ge 0}$ with $X_0 \sim \nu$ given that $X_1 = y$.*

This result follows of Proposition 1.5.1, see [Bladt & Sørensen (2009].

The generalization of this results to an interval $[0, \Delta]$, the prove that for ergodic diffusions the probability π_Δ that a $(0, x, \Delta, y)$-bridge is hit by an independent diffusion with initial distribution $p_\Delta(y, *)$ is close to one and that the rejection probability $P(\tau > \Delta)$, where τ is the stopping time, converges to zero obviously are presented in [Bladt & Sørensen (2009].

1.6 Integrated Diffusion Process

We consider the one-dimensional diffusion

$$dX_t = \mu(X_t; \psi)dt + \sigma(X_t; \psi)dW_t$$

where ψ is an unknown p-dimensional parameter belonging to the parameter space $\Psi \subseteq \mathbb{R}^p$ and W is a one-dimensional standard Wiener process. We assume that X_0 is independent of W, that the stochastic differential equation has a unique weak solution, and that X is an ergodic, stationary diffusion with invariant measure ν_ψ.

Suppose that the interval of observation of the paths of the diffusion is $[0, T]$ but, a running integral of the process with respect to some weight function is available. Specifically, suppose the interval of observation $[0, T]$ is subdivided into n smaller intervals of length $\Delta = T/n$ and let v be a probability measure on the interval $[0, \Delta]$. We shall consider observations of the form

$$Y_i = \int_0^\Delta X_{(i-1)\Delta+s} dv(s)$$

for $i = 1, \ldots, n$.

Since that, typically, v will have a density π with respect to the Lebesgue measure on $[0, \Delta]$, we can write, in this case

$$Y_i = \int_{(i-1)\Delta}^{i\Delta} X_s \pi(s - (i-1)\Delta) ds \tag{1.73}$$

for $i = 1, \ldots, n$.

Suppose that the observations are obtained by integrating uniformly over the time axis, then v is simply the uniform distribution on $[0, \Delta]$ with $\pi = 1/\Delta$, then we can write equation 1.73

$$Y_i = \frac{1}{\Delta} \int_{(i-1)\Delta}^{i\Delta} X_s ds$$

Remark 1.6.1 *Since the process X is invariant under time translations by stationarity, the $Y = \{Y_i\}$ is a stationary process.*

Here we consider the estimation of the parameter ψ when we observe a discrete-sampling of the integrated process. Integrated diffusion processes play an important role in finance as models for realized volatility, see e.g. [Andersen & Bollerslev (1998], [T. Andersen & Labys (2001], [Bollerslev & Zhou (2002], and [Barndorff-Nielsen & Shephard (2002].

These processes are also used for modelling purposes in fields of engineering and the sciences. An Example is provided by the records of the concentration of oxygen isotopes in ice-core data from Greenland and Antarctica, see e.g. [Ditlevsen, Ditlevsen & Andersen (2002]. Such data are used to investigate the paleo-climate.

Estimation of parameters in diffusion models is investigated when the observations are integrals over intervals of the process with respect to some weight function. This type of observations can, for Example, be obtained when the process is observed after passage through an electronic filter.

Parametric inference for integrated diffusion process has been considered by [Gloter (2006], and [Bollerslev & Zhou (2002]. Nonparametric inference has been considered in [Comte, Genon-Catalot & Rozenholc (2009].

Here, we will briefly outline of the optimal prediction-based inference for integrated diffusions.

Prediction Based Estimating Functions

We consider the diffusion define by 1.73 and again we suppose that we don't observe X itself but we observe a discrete sampling of integrate process $Y = (Y_1, \ldots, Y_n)$ is available, but now we consider observations of the form [Ditlevsen & Sørensen (2004]

$$Y_i = \int_{t_{i-1}}^{t_i} X_s ds,$$

for $i = 1, \ldots, n$.

Remark 1.6.2 *Note that by stationarity, the law of the process X is invariant under time translations, which easily implies that Y is a stationary process.*

In this section we present the problem of estimating the parameter $\theta = (\alpha, \sigma)$ of the process X by applying the method of prediction-based estimating functions introduced in [Sørensen (2000].

An estimate is a function $G_n(\theta)$ that depends on the parameter as well as on the observations. We can obtain an estimator by solving the equation

$$G_n(\theta) = 0.$$

Define the one dimensional functions $f_j, j = 1, \ldots, m$ on the state space of Y, such that $E_\theta(f_j(Y_i)^2) < \infty$ with $i = 1, \ldots, n$ and we denote the expectation when θ is the true parameter value by $E_\theta(*)$. Define $\mathcal{F}_i := \sigma_{k \leq i}(Y_k)$ and $\mathcal{H}_i^\theta = L^2(\mathcal{F}_i)$ and let $P_{i,j}^\theta, j = 1, \ldots, m$ be closed linear subspaces of $\mathcal{H}_i^\theta$.

The subspace $P_{i,j}^\theta$ can be interpreted as a set predictors of $f_j(Y_{i+1})$ given $Y_1, \ldots, Y_i$. We will study the estimating function

$$G_n(\theta) = \sum_{i=1}^{n} \sum_{j=1}^{n} \Pi_j^{i-1}(\theta)(f_j(Y_i) - \hat{\pi}_j^{i-1}(\theta)), \tag{1.74}$$

with $\Pi_j^{i-1}(\theta) = (\pi_{1,j}^{i-1}(\theta), \ldots, \pi_{p,j}^{i-1}(\theta))^T$ a p-dimensional stochastic vector, the coordinates of which belong to $P_{i-1,j}^\theta$ and where $\hat{\pi}_j^{i-1}(\theta)$ is the minimun mean square error predictor of $f_j(Y_i)$ in $P_{i-1,j}^\theta$, then $\hat{\pi}_j^{i-1}(\theta)$ is the orthogonal projection of $f_j(Y_i)$ on $P_{i-1,j}^\theta$ with respect to the inner product in $\mathcal{H}_i^\theta$.

The projection exist and is uniquely determined by the normal equations

$$E_\theta(\pi(f_j(Y_i) - \hat{\pi}_j^{i-1}(\theta))) = 0$$

with $\pi \in P_{i-1,j}^\theta$.

We shall be particularly interested in prediction-based estimating functions where each of the sets $P_{i-1,j}^\theta$ is finite dimensional. We assume that $P_{i-1,j}^\theta$ is spanned by $Z_{j0}^{i-1}, Z_{j1}^{i-1}, \ldots, Z_{jq_{ij}}^{i-1}$ of the form $Z_{jk}^{i-1} = h_{jk}^i(Y_1, \ldots, Y_{i-1}), k = 1, \ldots, q_{ij}$, which are linearly independent in $\mathcal{H}_{i-1}^\theta$. Assume that Z_{j0}^{i-1} is constant equal to 1 and thus we can write the elements of $P_{i-1,j}^\theta$ in the form $a^T Z_j^{i-1}$ with $a^T = (a_0, \ldots, a_{q_j})$ and $Z_j^{i-1} = (Z_{j0}^{i-1}, Z_{j1}^{i-1}, \ldots, Z_{jq_{ij}}^{i-1})^T$. When θ is the true parameter value, we define $C_j(\theta)$ as the covariance matrix of Z_j^{i-1} and $b_j(\theta) = (Cov_\theta(Z_{j1}^r, f_j(Y_{r+1})), \ldots, Cov_\theta(Z_{jq_j}^r, f_j(Y_{r+1})))^T$ and by the normal equations, the minimum mean square error predictor of $f_j(Y_i)$ in $P_{i-1,j}^\theta$ is given by

$$\hat{\pi}_j^{i-1}(\theta) = \hat{a}_{j0}(\theta)^T + \hat{a}_j(\theta)^T Z_j^{i-1}$$

where

$$\hat{a}_j(\theta)^T = C_{i-1,j}(\theta)^{-1} b_j^{i-1}(\theta) \tag{1.75}$$

and

$$\hat{a}_{j0}(\theta)^T = E_\theta(f_j(Y_i)) - \hat{a}_j(\theta)^T E_\theta(Z_j^{i-1}). \tag{1.76}$$

The optimal prediction-based inference for integrated diffusions

We use the results an notation in [Sørensen (2000] and [Ditlevsen & Sørensen (2004] we have that the optimal estimating function of the type equation (1.74) is given by

$$G_n^* = A_n^* \sum_{i=r+1}^{n} H^{(i)}(\theta),$$

with

$$H^{(i)}(\theta) = Z^{i-1)}(F(Y_i) - \hat{\pi}^{(i-1)}(\theta)),$$

where

$$F(x) = (f_1(x), \ldots, f_N(x))^T,$$

$$\hat{\pi}^{(i-1)}(\theta) = (\hat{\pi}_1^{(i-1)}(\theta), \ldots, \hat{\pi}_N^{(i-1)}(\theta))^T,$$

$$Z^{(i-1)} = \begin{pmatrix} Z_1^{(i-1)} & a_{q_1} & \ldots & a_{q_1} \\ a_{q_2} & Z_1^{(i-1)} & \ldots & a_{q_2} \\ \ldots & \ldots & & \ldots \\ a_{q_N} & a_{q_N} & \ldots & Z_N^{(i-1)} \end{pmatrix}$$

and

$$A_n^*(\theta) = \partial_\theta \hat{a}(\theta)^T \bar{C}(\theta) \bar{M}_n(\theta)^{-1},$$

where

$$\bar{M}_n(\theta) = E_\theta(H^{r+1}(\theta) H^{r+1}(\theta)^T)$$

$$+ \sum_{k=1}^{n-r-1} \frac{n-r-k}{n-r} \times [E_\theta(H^{r+1}(\theta) H^{r+1+k}(\theta)^T) + E_\theta(H^{r+k+1}(\theta) H^{r+1}(\theta)^T)],$$

$$\bar{C}(\theta) = E_\theta(Z^{(i-1)}(Z^{(i-1)})^T)$$

and

$$\hat{a}(\theta)^T = (\hat{a}_1(\theta)^T, \ldots, \hat{a}_N(\theta)^T)$$

where $\hat{a}_j(\theta)$ is given by equation (1.76).

Simulated Data Example

In this section we apply the method developed above to the Ornstein-Uhlenbeck process, which is a solution of the stochastic differential equation

$$dX_t = -\alpha X_t d_t + \sigma dW_t, \tag{1.77}$$

where α and σ are unknown parameters to be estimates, and W is the standard Wiener process.

Assume that $\sigma > 0$ for all x in the state interval.

Now, we will find the optimal prediction-based estimating function with $N = 1$,$f_1(y) = y$,$f_2(y) = y^2$,$Z_{1,0}^{(i-1)} = Z_{2,0}^{(i-1)} = 1$ and $Z_{1,1}^{(i-1)} = Y_{i-1}$.

Then following the results in [Ditlevsen & Sørensen (2004] we have that the optimal-based estimate is given by:

$$\frac{(1-e^{-\alpha\Delta})^2}{2(\alpha\Delta - 1 + e^{-\alpha\Delta})} = \frac{\sum_{i=2}^{n} Y_{i-1}Y_i}{\sum_{i=2}^{n} Y_{i-1}^2} \tag{1.78}$$

and

$$\sigma^2 = \frac{\alpha^3\Delta^2\sum_{i=2}^{n} Y_i^2}{(n-1)(\alpha\Delta - 1 + e^{-\alpha\Delta})} \tag{1.79}$$

Remark 1.6.3 $\alpha\Delta - 1 + e^{-\alpha\Delta} > 0$ *when* $\alpha > 0$*, then is no solution if* $\sum_{i=2}^{n} Y_{i-1}Y_i < 0$.

Output Analysis

In this section the Milstein scheme was used to simulate a Ornstein-Uhlenbeck process with discretization level ($n = 1000$) with $\alpha = 0.1$ and $\sigma = 0.5$ In this the the value initial X_0 has density given by (see Kutoyants Y.(2004)).

$$X_0 \sim N(0, \sigma^2/2\alpha).$$

Summaries of the implementation output for the simulated data set are reported in Table III.

α	σ
0.098165542	0.5093917

Table III.

Summaries of the implementation output for the simulated data set with discretization level ($n = 5000$) are reported in Table IV.

α	σ
0.098505601	0.4707101

Table IV.

The method was implemented in Fortran 95.

Chapter 2

Statistical inference for Markov Jump Processes

In this chapter we present an algorithm for the estimation of transition rates by a Markov Chain Monte Carlo (MCMC) approach to observations from several Markov jump processes which conditional on the underlying Markov jump process are independent with the same transition rates and apply our results to analysis of credit rating transition.

2.1 Introduction

Markov jump processes with finite state space have many applications. The likelihood estimation theory based on a continuous time observation of a Markov jump process (MJP) is well-known; see, for Example,
[Billingsley (1961], [Jacobsen (1982] and [Kuchler & Sørensen (1999]). The case when the MJP is observed only at discrete time points has been studied in the last decade, for Example, study of the problem for discretely observed birth process and birth-and-death process(see [Keiding (1974] and [Keiding (1975]), [Dehay & Yao (2006], proposed an approach based on an explicit formula for the transition matrix of the sampled chain assuming that the intensity function is bounded above and away from zero and [Bladt & Sørensen (2005] demonstrate that the maximum likelihood estimator can found either by the EM algorithm or by a Markov chain Monte Carlo procedure.

An important application of MJP is in credit risk modelling; see
[R. Jarrow & Turnbull. (1997], a method of estimating the jump intensities from discrete observations can found in [R. Israel & Wei (2001] and [Bladt & Sørensen (2006] demonstrated that continuous-time Markov model can also be used to analyse discrete-time observations, where ratings have only been observed at discrete points in time.

In this chapter we present the explicit likelihood function for a Markov jump processes when its has been observed discretely and continuously and the principal objective is discuss the problems that are related to maximum likelihood estimation of the intensity matrix based on a discretely sampled Markov jump processes generalizing the results of Bladt and Sørensen in [Bladt & Sørensen (2005] on estimation of transition rates by a MCMC approach to observations from several Markov jump processes which conditional on underlying Markov jump process are independent with the same transition rates and apply our results to analysis of credit rating transition .

2.2 The Likelihood Function

2.2.1 Continuous Likelihood Function

Suppose that the Markov jump process $X = \{X_t\}_{t \geq 0}$ with finite state space $E = \{1, 2, \ldots, m\}$ and infinitesimal generator Q has been observed continuously in certain time interval $[0, T]$. The process is assumed to be a separable continuous-time homogeneous and irreducible. With probability one, all sample paths are right-continuous step functions with a finite number of jumps in each finite interval.

A sample path in a given interval $[0, T]$ is characterized by the number of jumps, the sequence of visited states, and the time spent in each state within that interval.

Using the random variables of the Definition 4.2.5, a sample path of $X = \{X_t\}_{0 \leq t \leq T}$ can be represented as an ordered sequence:

$$\{X_t\}_{0 \leq t \leq T} = \{(x_1, \Delta t_1), (x_2, \Delta t_2), \ldots, (x_n, \Delta t_n), x_{n+1}\}.$$

Note that n jumps, in all, have been made.

Then the sample path starts at x_1 at time zero, remains in x_1 for Δt_1 units of time, makes a jump to x_2, remains in x_2 for Δt_2 units of time,..., jumps to x_n, remains there for Δt_n units of time and then makes the final jump to t_{n+1} and remains there at least until time T.

Now we can write the probability distribution:

Theorem 2.2.1 *Let*

$$r_{ij} = \begin{cases} 0 & \textit{if} \quad i = j \\ q_{i,j} & \textit{if} \quad i \neq j \end{cases}$$

then

$$P(N(T) = n, X_{t_0} = x_1, \Delta t_1 \leq s_1, \ldots, X_{t_{n-1}} = x_n, \Delta t_n \leq s_n, X_{t_n} = x_{n+1}) \quad = \\ P(X_{t_0} = x_1) e^{-q_{x_{n+1}}} \textstyle\int_{\tau_n} \prod_{i=1}^{n} d\Delta t_i r_{x_i x_{i+1}} e^{-(q_{x_i} - q_{x_{n+1}})\Delta t_i}$$

where

$$\tau_n = \{(\Delta t_1, \Delta t_2, \ldots, \Delta t_n) : \sum_{i=1}^{n} \Delta t_i < T, 0 \leq \Delta t_i \leq s_i\},$$

if $n > 0$ *and* $P(N(T) = 0, X_{t_0} = x_1) = P(X_{t_0} = x_1) e^{-q_{x_1} T}$.

Proof. The assertions follows from Theorem 1.3.4.

Now, if we define

$$\Omega_n = (\prod_{i=1}^{n}(E \otimes \mathbb{R})) \otimes E$$

every sample path of MJP which makes n jumps can be represented as a point in Ω.

Let λ be the Lebesgue measure on $\mathbb{R}$, $\mathcal{C}$ be the counting measure on E and let μ_n be the product measure on Ω_n, defined by the relation:

$$\mu_n = (\prod_{i=1}^{n}(\mathcal{C} \times \lambda)) \times \mathcal{C}.$$

Then every sample path of process $X = \{X_t\}_{0 \leq t \leq T}$ can be represented as a point in

$$\Omega = \cup_{n=0}^{\infty} \Omega_n.$$

For each set $W \subseteq \Omega$ for which $W \cap \Omega_n$ is μ_n measurable define

$$\mu^*(W) = \sum_{n=0}^{\infty} \mu_n(W \cap \Omega_n).$$

Let μ be a measure on the space of all sample functions, defined for all subsets B whose intersection with Ω is in $\mathfrak{B}$ (the Borel σ-field):

$$\mu(B) = \mu^*(B \cap \Omega).$$

The density of the Markov jump process can be written

Theorem 2.2.2 *If B is a subset of the space of all sample functions over* $[0, T]$ *which is measurable with respect to μ, then*

$$P(B) = \int_B f_Q(x^*) d\mu(x^*),$$

where x^* denote a realization of $X = \{X_t\}_{0 \leq t \leq T}$ and

$$f_Q(x^*) = \begin{cases} P(X_{t_0} = x_1)e^{-q_{x_1}T} \qquad \textit{if} \qquad x^* = x_1 \\ P(X_{t_0} = x_1)e^{-q_{x_{n+1}}T} \prod_{i=1}^{n} r_{x_i x_{i+1}} e^{-(q_{x_i} - q_{x_{n+1}})\Delta t_i} \\ \qquad \textit{if} \qquad x^* = \{(x_1, \Delta t_1), (x_2, \Delta t_2), \ldots, (x_n, \Delta t_n), x_{n+1}\} \\ 0 \qquad\qquad \textit{otherwise.} \end{cases}$$

In second case with $n > 0, x_i \in E, \Delta t_i \geq 0, i = 1, 2, \ldots, n$ and $\sum_{i=1}^{n} \Delta t_i < t$.

Proof. See [Doob (1953].

Now, we suppose that k independent realizations $x_1^*, x_2^*, \ldots, x_k^*$ of $X = \{X_t\}_{0 \leq t \leq T}$ are continuously observed. The likelihood function, $L_T^{(c)}(Q)$, has been traditionally defined by the equation

$$L_{T,k}^{(c)}(Q) = \prod_{i=1}^{k} f_Q(x_i^*).$$

Let the random variable $R_i^k(T)$ be the time spent in state i before time T during the k trials

$$R_i^k(T) = \sum_{j=1}^{k} \int_0^T I_i^j(X_s)ds$$

and denote by $N_{ij}^k(T)$ the number of transition from i to state j in the time interval $[0, T]$ observed during the k trials. The continuous time likelihood function $L^{(c)}$ of the observed trajectories $x_1^*, x_2^*, \ldots, x_k^*$ of $\{X_t : 0 \leq t \leq T\}$ is given by

$$L_{T,k}^{(c)}(Q) = C_k \prod_{i=1}^{m} \prod_{i \neq j} q_{ij}^{N_{ij}^k(T)} e^{-q_{ij} R_i^k(T)}. \tag{2.1}$$

Considering the log-function we have

$$\log L_{T,k}^{(c)}(Q) = C_k + \sum_{i=1}^{m} \sum_{i \neq j} [N_{ij}(T)^k \log(q_{ij}) - q_{ij} R_i^k(T)]. \tag{2.2}$$

The Halmos-Savage factorization Theorem can be applied to the last expression, and we see that the set

$$\{N_{ij}(T)^k, R_i^k(T)\}_{i \neq j}$$

is a sufficient statistic for Q. Then a easy calculation shows

$$\frac{\partial \log L_{T,k}^{(c)}(Q)}{\partial q_{ij}} = 0$$

if and only if

$$q_{ij} = \frac{N_{ij}^k}{R_i^k(T)}$$

and

$$\frac{\partial \log L_{T,k}^{(c)}(Q)}{\partial q_{ij} \partial q_{r,l}} = \frac{-N_{ij}^k(T)}{q_{ij}^2} I_r(i) I_l(j) \leq 0$$

then the maximun likelihood estimator of Q is

$$\hat{q}_{ij}(T) = \frac{N_{ij}}{R_i(T)} \tag{2.3}$$

provided that $R_i^k(T) > 0$. If the process has not been in state i, there is no information about q_{ij} in the data, and the maximum likelihood estimator of q_{ij} does not exist.

2.2.2 Discrete Likelihood Function

Consider the case where the process has only been observed at discrete time points $0 = t_0, t_1, \ldots, t_n = T$, then the process $Y = \{Y_i\}_{i=0}^n = \{X_{t_i}\}_{i=0}^n$ is discrete time Markov chain, in general time-inhomogeneous and the discrete likelihood function for the discrete time data is given in terms of the transition matrix $P^{\Delta_i}(Q) = \exp(\Delta_i Q)$,

$$L_n^{(d)}(Q) = \prod_{i=1}^{n-1} P^{\Delta_i}_{x_i,x_{i+1}}(Q) \tag{2.4}$$

where $x_0, x_1, \dots, x_n$ denote the observed values of X, $\Delta_i = t_{i+1} - t_i$ and $Q \in \mathscr{Q}$.

In the case where the Markov chain Y is time-homogeneous (when $\Delta_i = \Delta$) with transition matrix $P^{\Delta}(Q)$, so the likelihood function is given by

$$L_n^{(d)}(Q) = \prod_{i=1}^{m}\prod_{j=1}^{m} (P^{\Delta}_{ij}(Q))^{c_{ij}(n)} \tag{2.5}$$

where $c_{ij}(n)$ is the number of transition from i to j in the discrete time Markov chain Y. In this case the discrete log-likelihood function

$$\log L_n^{(d)}(Q) = \sum_{i=1}^{m}\sum_{j=1}^{m} (\log(P^{\Delta}_{ij}(Q)^{c_{ij}(n)}) \tag{2.6}$$

has the following derivative with respect to the entries of Q

$$\frac{\partial}{\partial Q} \log L_n^{(d)}(Q) = \sum_{k=1}^{\infty}\sum_{l=1}^{k} \frac{\Delta^k}{k!}(Q^T)^{l-1} Z (Q^T)^{k-l} \tag{2.7}$$

where $Z = (Z_{ij})_{ij \in E}$ and $z_{ij} = c_{ij}(n)/exp(\Delta Q)_{ij}$, which has a complicated form that can not be found analytically. Hence no analytical expression for the maximun likelihood estimator with respect to Q is available.

On the other hand the derivative of log-likelihood with respect to the transition matrix P can analytically be obtained and the maximum is given by

$$\hat{P}_{ij} = \frac{c_{ij}(n)}{\sum_{j=1}^{m} c_{ij}(n))} \tag{2.8}$$

The Embedding Problem.

As pointed out in the previous section the discrete likelihood function $L_n^{(d)}$ does not permit an analytical maximum likelihood estimator. On the other hand, the maximun likelihood estimator for a continuous time observation can be obtained analytically, but for an incomplete observation the information between two consecutive observations is hidden and, hence, the observables $R_i(T)$ and $N_{ij}(T)$ are unknown.

In this section we consider the problem of how to determine the generator if only observations at discrete time points $0 = t_0, t_1, \dots, t_n = T$ are available. There are, however, several problems here. First, from a finite number of samples it is impossible to tell if the underlying process is actually Markovian. Second, it is not clear if the observed data originate indeed from discrete samples of a continuous-time Markov chain with some generator Q, or rather from a discrete-time Markov chain which cannot be embedded into a time-continuous counterpart. In the latter case, a generator does not exist because the transition matrix of the discrete chain does not belong to the set

$$P_0 = \{exp(Q)|Q \in \mathscr{Q}\},$$

and the problem of identifying the set P_0 is very complicated when $m > 2$. It is usually referred to as *the embedding problem*, then if we calculate $\hat{P}$ by equation (2.8) based on our discrete time observations of a continuous time Markov jump process and $\hat{P} \in P_0$, there is a $\hat{Q} \in \mathscr{Q}$ such that $P^{\Delta}(\hat{Q}) = \hat{P}$, and the likelihood function given by equation (2.5) attains its maximal value at $\hat{Q}$, which is thus the maximum likelihood estimator. A third difficulty is the fact that the matrix exponential function is not injective if the eigenvalues of the generator are complex, so $\hat{Q}$ need not be unique. When $\hat{P} \notin P_0$, the situation is not clear owing to the complicated structure of P_0, but it seems not to be uncommon that the maximum likelihood estimator does not exist, in particular when the time Δ between observations is large. General results on the existence and uniqueness of the maximum likelihood estimator are summarized in the next Theorem enunciated by Bladt and Sørensen in [Bladt & Sørensen (2005]

Theorem 2.2.3 *If $\hat{P}$ given by equation (2.8) belongs to P_0, then the maximum likelihood estimator of the intensity matrix $\hat{Q}$ exists and is the solution to $\hat{P} = exp(\Delta\hat{Q})$. If $\hat{P} \notin P_0$, then either the maximum likelihood estimator exists and satisfies the condition that $exp(\Delta\hat{Q}) \in \delta P_0$ or the likelihood function given by equation (2.5) has no maximum in $\mathscr{Q}$. If the true transition matrix Q_0 satisfies the condition that $exp(\Delta Q_0) \in int(P_0)$ and if the Markov process is ergodic, then the probability that the maximum likelihood estimator exists goes to 1 as $n \to \infty$, and $exp(\Delta\hat{Q}) \to exp(\Delta Q_0)$ almost surely. Moreover, if Q_0 satisfies the condition that $exp(\Delta Q_0) \in int(P_{00})$, then the probability that the maximum likelihood estimator is unique goes to 1 and $\hat{Q} \to Q_0$ almost surely as $n \to \infty$. The condition $exp(\Delta Q_0) \in int(P_{00})$ is satisfied when Δ is sufficiently small.*

In here δP_0 denote the boundary of P_0 relative to P_+, then

$$\delta P_0 = (\cup_{i \neq j} E_{ij}) \cup E$$

where E_{ij} is a non-empty subset of the set of exponential of intensity matrices with $q_{ij} = 0$, and E is a non-empty subset of the $m \times m$ transition matrices with fewer than m distinct eigenvalues,

$$P_+ = \{P \in \mathscr{P}|det(P) > 0\},$$

and P_{00} denote the subset of P_0 of transition matrices $P \in P_0$, for which Q is uniquely determined by $P = exp(Q)$, for details of Theorem see [Bladt & Sørensen (2005].

2.3 Markov jump processes in a random environment

In this section we generalize the results of Bladt and Sørensen in [Bladt & Sørensen (2005] on estimation of transition rates by an MCMC approach to observations from several Markov jump processes which conditional on underlying Markov jump process are independent with the same transition rates.

2.3.1 Model and Data

Let $U = \{U_t\}_{t\geq 0}$ be a Markov jump process on a finite state space $\{1, 2, \ldots, d\}$ and with intensity matrix $\Lambda_0 = \{\lambda_{0ij}\}_{ij=1,\ldots,d}$. Let $X_i = \{X_t^i\}_{t\geq 0}, i = 1, \ldots, n$ be Markov jump processes with finite state space $E = \{1, \ldots, m\}$ which conditional on U are independet and identically distributed and such that X_i has intensity matrix Λ_{U_t} where $\Lambda_k = \{\lambda_{kij}\}_{ij,=1,\ldots,m}$. We may think of U_t as an underlying environment and define $\Lambda = (\Lambda_0, \Lambda_1, \ldots, \Lambda_d)$ the full parameter. Our aim is to estimate the parameter Λ.
If the X_i's have been observed continuously in the time interval $[0, \tau]$. The conditional likelihood function given U is

$$L_\tau(X_1, \ldots, X_n | U; \Lambda) = \prod_{v=1}^{d} \prod_{i=1}^{m} \prod_{j\neq i} \lambda_{vij}^{N_{vij}(\tau)} e^{-\lambda_{vij} R_{vi}(\tau)}, \tag{2.9}$$

where $N_{vij}(\tau)$ is the number of transitions from state i to state j of all the processes X_i when the underlying environment is in state v in the time interval $[0, \tau]$, and

$$R_{vi}(\tau) = \sum_{k=1}^{n} \int_0^\tau I_{i,v}\{X_s^k, U_s\} ds, \tag{2.10}$$

is the time spent in state i of all processes X_i while the underlying environment is in state v before time τ. The likelihood function of U is given by

$$L_\tau(U; \Lambda) = \prod_{i=1}^{d} \prod_{j\neq i} \lambda_{0ij}^{N_{0ij}(\tau)} e^{-\lambda_{0ij} R_{0i}(\tau)}, \tag{2.11}$$

where $N_{0ij}(\tau)$ is the number of transitions from state i to state j of the process U in the time interval $[0, \tau]$ and $R_{0i}(\tau)$ is the total time spent in state i of the process U before time τ.

For the case when the processes have been discretely observed, we consider three cases:

1. The case when the underlying process has been observed continuously in the time interval $[0, \tau]$.
2. The case when the underlying process has been observed only at discrete time points.
3. We assume that the underlying process is a diffusion process and we will study the rates of jumps in the Markov jump processes.

2.3.2 Observed Discretely Markov Jump Processes.

We consider the situation where the processes X_i's have been observed discretely and suppose two cases: one is that the underlying process U has been observed continuously and the other is that this process has been observed only at discrete time points.

In the first case we suppose that $0 \leq t_1 < \ldots < t_k \leq \tau$ be k time points at which the processes X_i's have been observed and that the Markov jump process U has been observed continuously in the time interval $[0, \tau]$. The observed data point of the processes X_i's are denoted by $x = \{x^1, \ldots, x^n\}$ where $x^i = \{X^i_{t_1} = x^i_1, \ldots, X^i_{t_k} = x^i_k\}$ and $U^c = \{U_t | 0 \leq t \leq \tau\}$ is the continuous time sample paths of the process U. We assume that time points at which the processes X_i's have been observed and number of observations is the same for all processes.

Markov chain Monte Carlo estimation when the underlying process has been observed continuously.

In the MCMC approach we choose a prior $\phi(\Lambda)$ and find the conditional distribution of Λ given the data (x, U^c). In fact, this method provides draws from the conditional distribution of (Λ, X) given (x, U^c), where $X = \{X^i_t | 0 \leq t \leq \tau, i = 1, \ldots, n\}$ is the collection of continuous time sample paths of the processes X_i for $i = 1, \ldots, n$. For this we employ the Gibbs sampler with two sites λ and X and sample by alternately drawing X given (Λ, x, U) and Λ given (X, x, U^c) (x is of course of no importance when conditioning on X). Iteration of the Gibbs sampler results in a sequence of variables $(\lambda^{(n)}, X^{(n)})$. Under suitable conditions the Gibbs sampler will eventually produce a stationary and ergodic sequence, i.e., after discarding a certain burn-in period, say the first $K - 1$ iterations, the sequence $(\lambda^{(n)}, X^{(n)})_{n \geq K}$ may be considered stationary, and the stationary distribution is exactly the conditional distribution of (Λ, X) given (x, U^c). By ergodicity, the empirical average

$$\frac{1}{M} \sum_{i=K}^{K+M} \Lambda^{(i)}$$

converges to the true mean of Λ conditionally on (x, U^c).

In situations where Λ is not uniquely determined by the distribution of the discrete time sample, the mean of the posterior distribution may not be a meaningful quantity. However, functional of Λ that are invariant under different representations of the distribution of the data can be estimated using a method similar to that just described. Specifically, let F be some functional which depends on the distribution of the data and is invariant under changes of the representation by Λ (i.e., if Λ_1 and Λ_2 result in the same distribution of the data, then $F(\Lambda_1) = F(\Lambda_2)$).

Then we can estimate $F(\Lambda)$

$$\frac{1}{M} \sum_{i=K}^{K+M} F(\Lambda^{(i)}).$$

For instance, the transition matrix $P_t(\Lambda) = \exp(t\Lambda)$ can be estimated in this way if “t“ is the time between two observations in the data (or a multiple of such a time).

Now, we choose the prior

$$\phi(\Lambda) \propto \prod_{i=1}^{d}\prod_{j\neq i}\lambda_{0ij}^{\alpha_{0ij}-1}e^{-\lambda_{0ij}\beta_{0i}}\prod_{v=1}^{d}\prod_{i=1}^{p}\prod_{j\neq i}\lambda_{vij}^{\alpha_{vij}-1}e^{-\lambda_{vij}\beta_{vi}}. \tag{2.12}$$

Where α_{vij} and β_{vi} are constants to be chosen conveniently, $\lambda_{vij} \sim Gamma(1/\beta_{vi}, \alpha_{vij})$ and λ_{vij}'s are independent. In this way parameters near the critical boundary, where $det(\exp(\Lambda)) = 0$, are effectively penalized because there at least one of the λ_{vij}'s must go to infinity.

Thus problems of non-existence of the estimator are avoided. This family of priors is conjugate for the model for continuous observation in the time interval $[0, \tau]$, which is an exponential family of processes, see [Kuchler & Sørensen (1999].Then samples of Λ given (X, U^c) are drawn from the posterior distribution:

$$\begin{aligned} p^*(\Lambda) &\propto \phi(\Lambda)p(X|\Lambda, U)p(U|\Lambda) \\ &\propto \prod_{i=1}^{p}\prod_{j\neq i}\lambda_{0ij}^{N_{0ij}+\alpha_{0ij}-1}e^{-\lambda_{0ij}(R_{0i}+\beta_{0i})}\prod_{v=1}^{d}\prod_{i=1}^{p}\prod_{j\neq i}\lambda_{vij}^{N_{vij}+\alpha_{vij}-1}e^{-\lambda_{vij}(R_{vi}+\beta_{vi})}. \end{aligned}$$

The Gibbs sampler now works as follows

1. Draw initial Λ from the prior distribution.
2. Simulate a sample of $X|U^c, \Lambda, x$.
3. Calculate the statistics N_{vij} and R_{vi}.
4. Draw a new Λ from the posterior distribution.
5. Go to 2.

Step two requires a simulation of the Markov jump processes step-by-step through the intervals $[t_j, t_{j+1}]$ starting from the initial condition $X^i_{t_j} = x^i_j$ such that the process will be $X^i_{t_{j+1}} = x^i_{j+1}$ for all $i = 1, \dots, n$ and $j = 1, \dots, k-1$. This can be done by simple rejection if the criterion it not me and acceptance otherwise.

A simulation study. In this section we present a simulation study of estimation of full parameter Λ when X's have been observed discretely and U has been observed continuously. We suppose that we have a sample path of the Markov jump process U with three states in the time interval $[0, 500]$ and we know that this process has intensity matrix

$$\Lambda_0 = \begin{pmatrix} -0.10 & 0.05 & 0.05 \\ 0.10 & -0.20 & 0.10 \\ 0.15 & 0.50 & -.30 \end{pmatrix}.$$

Then the one-step transition probabilities matrix is

$$P_U = \begin{pmatrix} 0.5975 & 0.2358 & 0.16659 \\ 0.4716 & 0.3336 & 0.1947 \\ 0.4997 & 0.2920 & 0.2081 \end{pmatrix}$$

Met./Par.	λ_{112}	λ_{113}	λ_{121}	λ_{123}	λ_{131}	λ_{132}
TRUE	0.06	0.04	0.08	0.07	0.1	0.15
MCMC	0.0593	0.0613	0.0946	0.0516	0.0915	0.1975
0%	0.0090	0.0164	0.0350	0.0079	0.0093	0.0729
25%	0.0467	0.0489	0.0792	0.0390	0.0678	0.1648
50%	0.0576	0.0598	0.0933	0.0499	0.0879	0.1941
75%	0.0699	0.0719	0.1084	0.0623	0.1120	0.2264
100%	0.1499	0.1455	0.1920	0.1398	0.2519	0.5213

Table 2.1: TRUE, the parameter values that were used in the simulation of the data; MCMC, estimates when applying the MCMC method to discrete time data and the remaining rows are the quantiles of the data from the MCMC method when the underlying process is in the state 1.

On the other hand we have simulated a sample path of the Markov jump processes X_i with $i = 1, 2$ conditioned on U with three states in the time interval $[0, 500]$ and with known intensity matrices:

$$\Lambda_1 = \begin{pmatrix} -0.1 & 0.06 & 0.04 \\ 0.08 & -0.15 & 0.07 \\ 0.1 & 0.15 & -0.25 \end{pmatrix},$$

while the underlying process is in state 1,

$$\Lambda_2 = \begin{pmatrix} -0.05 & 0.025 & 0.025 \\ 0.07 & -0.1 & 0.03 \\ 0.12 & 0.08 & -0.2 \end{pmatrix}.$$

while the underlying process is in state 2 and

$$\Lambda_3 = \begin{pmatrix} -0.15 & 0.08 & 0.07 \\ 0.06 & -0.2 & 0.14 \\ 0.05 & 0.05 & -0.1 \end{pmatrix}.$$

while the underlying process is in state 3.

The data are the states of the X_i's processes at 500 time points, equidistantly displaced by 1.0 and the complete path of process U.

MCMC
We present the result of a small simulation study, in which we simulated 10,000 intensity matrices including an initial burn-in of 1,000 iterations. The values of the parameters in the prior are $\alpha_{lij} = \beta_{li} = 1$. The average of the 9,000 intensity matrices is presented in Table 2.1, 2.2 and 2.3.

Markov chain Monte Carlo estimation when the underlying process has been observed discretely.

We consider the situation where the processes X_i's and U have been discretely observed. We assume that time points at which the processes X_i's have been

Met./Par.	λ_{212}	λ_{213}	λ_{221}	λ_{223}	λ_{231}	λ_{232}
TRUE	0.025	0.025	0.07	0.03	0.12	0.08
MCMC	0.0540	0.0242	0.0785	0.1130	0.1380	0.1018
0%	0.0059	0.0008	0.0037	0.0094	0.0134	0.0021
25%	0.0400	0.0142	0.0510	0.0803	0.0964	0.0647
50%	0.0519	0.0218	0.0725	0.1080	0.1301	0.0938
75%	0.0660	0.0318	0.0998	0.1400	0.1710	0.1302
100%	0.1656	0.1093	0.3356	0.3646	0.5631	0.3697

Table 2.2: The corresponding results when the underlying process is in the state 2.

Met./Par.	λ_{312}	λ_{313}	λ_{321}	λ_{323}	λ_{331}	λ_{332}
TRUE	0.08	0.07	0.06	0.14	0.05	0.05
MCMC	0.0740	0.0388	0.1505	0.1002	0.0667	0.1011
0%	0.0044	0.0003	0.0085	0.0007	0.0002	0.0018
25%	0.0498	0.0219	0.1034	0.0613	0.0324	0.0615
50%	0.0696	0.0344	0.1418	0.0916	0.0573	0.0922
75%	0.0937	0.0513	0.1881	0.1303	0.0908	0.1321
100%	0.2343	0.2285	0.5222	0.4382	0.3813	0.5102

Table 2.3: The corresponding results when the underlying process is in the state 3.

observed and number of observations is the same for all processes. Observation points of the underlying process are not necessarily the same as the processes X_i's.

Then $0 \leq t_1 < \ldots < t_k \leq \tau$ be k time points at which the processes X_i's have been observed and $0 \leq t_{u_1} < \ldots < t_{u_r} \leq \tau$ be time points at which the process U has been observed. The observed data point are denoted by $x = \{x_1, \ldots, x_n\}$ where $x_i = \{X^i_{t_1}, \ldots, X^i_{t_k}\}$ and $u = (U_{t_{u_1}}, \ldots, U_{t_{u_r}})$ for the processes X_i's and U respectively.

We present an approach to estimating the full parameter *Lambda* using the Markov Chain Monte Carlo methods. In the MCMC approach we choose a prior $\phi(\Lambda)$ and find the conditional distribution of Λ given the data (x, u). We shall study the slightly more general problem of finding the conditional distribution of (Λ, X, U) given (x, u) where $X = \{X^i_t | 0 \leq t \leq \tau, i = 1, \ldots, n\}$ is the continuous time sample paths of the processes X_i.

Let ϑ denote the space of intensity matrices. Let E_X be the space of possible values of X and let E_U denote the space of possible sample paths of U. A Markov chain taking values in $\vartheta^{d+1} \times E_X \times E_U$ with stationary distribution equal to the conditional distribution of (Λ, X, U) given (x, u), and is given by the following Gibbs sampler with three sites, Λ, X and U.

Now, we sample by alternately drawing X given (Λ, U, x), U given (Λ, X, u)

and λ given (U, X). Here, there is the question of determination of the component $(\Lambda_v, v = 1, \ldots, d)$ to which each of the $X^i_{t_j}, i = 1, \ldots, n, j = 1, \ldots, k.$ belongs.

This is facilitated in mixture models, this classification expose a hidden structure in the model which may be viewed as missing data: each observation is associated with an unobserved indicator of componet from which it originated.

Then the algorithm is a data augmentation, which consider the complete model, namely the distribution of $(z^i_j, X^i_{t_j})$, where z^i_j is the component indicator such that $P^t_i | z^i_j = \exp(t\Lambda_{z^i_j})$ and we complete the missing data $Z = \{Z^1, \ldots, Z^n\}$ where $Z^i = \{z^i_1, \ldots, z^i_k\}$, $i = 1, \ldots, n$ and $j = 1, \ldots, k$.

Here, the Gibbs sampler works as:

1. Draw initial Λ from the prior distribution.
2. Update $Z|\Lambda, U, X$.
3. Simulate a sample of $X|Z, U, \Lambda, x$.
4. Simulate a sample of $U|Z, u, \Lambda, X$.
5. Calculate the statistics N_{vij} and R_{vi}.
6. Draw a new Λ from the posterior distribution.
7. Go to 2.

Samples of Λ.

In order to obtain samples of Λ given (U, X) we first define a prior

$$\phi(\Lambda) \propto \prod_{i=1}^{d} \prod_{j \neq i} \lambda_{0ij}^{\alpha_{0ij}-1} e^{-\lambda_{0ij}\beta_{0i}} \prod_{v=1}^{d} \prod_{i=1}^{p} \prod_{j \neq i} \lambda_{vij}^{\alpha_{vij}-1} e^{-\lambda_{vij}\beta_{vi}},$$

where α_{vij} and β_{vi} are constants to be chosen conveniently, $\lambda_{vij} \sim Gamma(1/\beta_{vi}, \alpha_{vij})$ and λ_{vij}'s are independent.

Then samples of Λ given (X, U) are drawn from the posterior distribution:

$$\begin{aligned} p^* &\propto \phi(\Lambda) p(X|\Lambda, U) p(U|\Lambda) \\ &\propto \prod_{i=1}^{p} \prod_{j \neq i} \lambda_{0ij}^{N_{0ij}+\alpha_{0ij}-1} e^{-\lambda_{0ij}(R_{0i}+\beta_{0i})} \prod_{v=1}^{d} \prod_{i=1}^{p} \prod_{j \neq i} \lambda_{vij}^{N_{vij}+\alpha_{vij}-1} e^{-\lambda_{vij}(R_{vi}+\beta_{vi})} \end{aligned}$$

Samples of Z.

In the step 2, we sample from conditional distribution of Z. Let $0 \leq t^i_1 < \ldots < t^i_{n_i} \leq \tau$ be n_i time points of jumps of the processes X_i, n_i the number of jumps of the processes X_i and $0 \leq t^u_1 < \ldots < t^u_{n_u} \leq \tau$ be n_u time points of jumps of the process U and n_u the number of jumps of the process U.

We define $t_j^{io} = max\{t_{r-1}^i | t_r^i \leq t_j, r = 1, \ldots, n_i\}$ and $t_j^{iu} = max\{t_{r-1}^u | t_r^u \leq t_j^{io}, r = 1, \ldots, n_u\}$ for $j = 1, \ldots, k$ and $i = 1, \ldots, n$.

Now, we consider the auxiliary variables $Y = (Y^1, \ldots, Y^n)$ where $Y^i = (y_1^i, \ldots, y_k^i)$, $X_{t_j}^{io} = y_j^i$ and $W = (W^1, \ldots, W^n)$ where $W^i = (w_1^i, \ldots, w_k^i)$, $U_{t_j}^{iu} = w_j^i$ for $j = 1, \ldots, k$ and $i = 1, \ldots, n$.. Then the conditional distribution of Z given the observations is

$$P(z_j^i = v | \Lambda, U, X) = \frac{\lambda_{vy_j^i x_j^i}}{\lambda_{vy_j^i}} \frac{\lambda_{0w_j^i v}}{\lambda_{0w_j^i}} \tag{2.13}$$

where $\lambda_{vi} = \sum_{j=1}^n \lambda_{vij}$ for $v = 1, \ldots, d$ and $\lambda_{0i} = \sum_{j=1}^d \lambda_{0ij}$.

Samples of X.

Samples of X given (Λ, U, x, Z) are obtain as follow: Simulate Markov jump processes X, with intensity matrix $(\Lambda_1, \ldots, \Lambda_d)$ up to time τ such that: $X_{t_j}^i = x_j^i$ and $U_{t_j^{iu}} = z_j^i$. This can be done by simple rejection if the criterion it not me and acceptance otherwise.

Samples of U.

To sampling of U given (X, Λ, u) we construct a Markov process with state-space E_U for which the stationary distribution is equal to $\pi(U) = p(U|X, \Lambda, u, Z)$, then the acceptance ratio is given by

$$\begin{aligned}
\alpha(U_0, U_1) &= min(1, \frac{\pi(U_1)q(U_0)}{\pi(U_0)q(U_1)}) \\
&= min(1, \frac{p(X|U_1, \Lambda)p(U_1|\Lambda, u)p(U_0|\Lambda, u)}{p(X|U_0, \Lambda)p(U_0|\Lambda, u)p(U_1|\Lambda, u)}) \\
&= min(1, \frac{p(X|U_1, \Lambda)}{p(X|U_0, \Lambda)})
\end{aligned}$$

where $q(U) = p(U|\Lambda, u)$.

Then samples of U given (X, λ, u) are obtain by Metropolis-Hasting algorithm:

1. Draw initial U_0 from q such that $U_{t_{u_i}} = u_i$ for $i = 1, \ldots, r$.
2. Draw initial U_1 from q such that $U_{t_{u_i}} = u_i$ for $i = 1, \ldots, r$.
3. Draw V from a uniform distribution on $[0, 1]$.
4. If $V \leq \alpha(U_0, U_1)$ then $U_0 := U_1$.
5. Go to 2.

The algorithm is uniformly ergodic because $\alpha(U_0, U_1)$ is bounded. Thus after a certain number of iterations (burn-in), any further draw from the Metropolis-Hasting algorithm may be consider as a draw from the target distribution.

A simulation study. In this section we present a simulation study of estimation of full parameter Λ when X and U have been discretely observed. We have simulated a sample path of the Markov jump process U with three states in the time interval $[0, 500]$ and with intensity matrix Λ_0.

On the other hand we have simulated a sample path of the Markov jump processes X_i with $i = 1, 2$ conditioned on U with three states in the time interval $[0, 500]$ and,with intensity matrices $\Lambda_1, \Lambda_2, \Lambda_3$ while the underlying environment is in state 1, state 2 or state 3 respectively.

The data are the states of the processes X_i' and U at 1000 time points, equidistantly displaced by 0.5 and at the same points.

MCMC

We present the result of a small simulation study, in which we simulated 10,000 intensity matrices including an initial burn-in of 1,000 iterations. The Metropolis-Hasting algorithm was run with an burn-in of 1,000 iterations. The values of the parameters in the prior were simply $\alpha_{lij} = \beta_{li} = 1$. The average of the 9,000 intensity matrices is

$$\Lambda_{0MCMC} = \begin{pmatrix} -0.074071147 & 0.036046077 & 0.038025122 \\ 0.078120135 & -0.2078160 & 0.1296959 \\ 0.1221993 & 0.2452670 & -0.3674664 \end{pmatrix}$$

$$\Lambda_{1MCMC} = \begin{pmatrix} -0.1218436 & 0.061492514 & 0.060351007 \\ 0.095069662 & -0.1422418 & 0.047172237 \\ 0.087769620 & 0.1996901 & -0.2874598 \end{pmatrix},$$

if the underlying environment is in state 1,

$$\Lambda_{2MCMC} = \begin{pmatrix} -0.079956256 & 0.054756112 & 0.025200104 \\ 0.067239515 & -0.1844493 & 0.1172098 \\ 0.1248880 & 0.1063413 & -0.2312294 \end{pmatrix}$$

if the underlying environment is in state 2 and

$$\Lambda_{3MCMC} = \begin{pmatrix} -0.1088241 & 0.067903571 & 0.040920567 \\ 0.1683657 & -0.2844607 & 0.1160949 \\ 0.089159317 & 0.096346438 & -0.1855058 \end{pmatrix}.$$

if the underlying environment is in state 3.

2.4 Analysis of credit risk data.

In this section we implement the proposed method for analyse credit rating data.

2.4.1 MCMC of macroeconomics variables.

Here, we analyze macroeconomics variables data. The data is a monthly record of the ratings of United States of America in the period of December 31, 1990 to December 31, 1999 of interest rate, the time serie of this data is shows in Figure 2.1

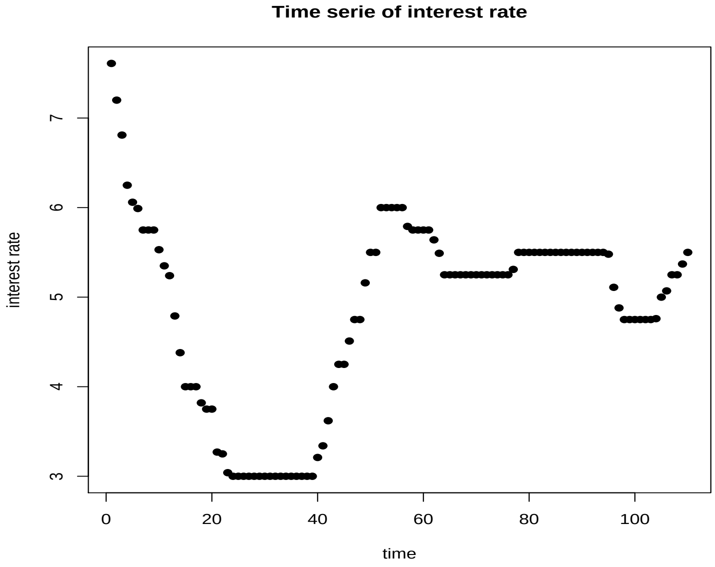


Figure 2.1: The data is a monthly record of the interest rate of United States of America.

Based on the observed data of the interest rate, we define a Markov jump process with three possible states, with respect to change in the interest rate on the market to study the impact of a change in this variable. The states are defined as follows: state 1 if the interest rate decreases, state 2 if it does not change and state 3 if the rate increases. Then, we have a jump Markov process observed in discrete time with three states and matrix of unknown intensity. The figure shows the behavior of the interest rate according to defined process and the data are the states of the process at 109 time points corresponding to the monthly observations. The Figure 2.2 shows the evolution of this process.

We present the result of a small simulation study, in which we simulated 10,000 intensity matrices including an initial burn-in of 1,000 iterations. The values of the parameters in the prior were simply $\alpha_{lij} = \beta_{li} = 1$. The average of the 9,000 intensity matrices is

INTEREST RATE

$$\Lambda_{INT} = \begin{pmatrix} -0.4170606 & 0.3531195 & 0.06394077 \\ 0.1354001 & -0.3158898 & 0.18048990 \\ 0.0930055 & 0.5632873 & -0.65629440 \end{pmatrix}.$$

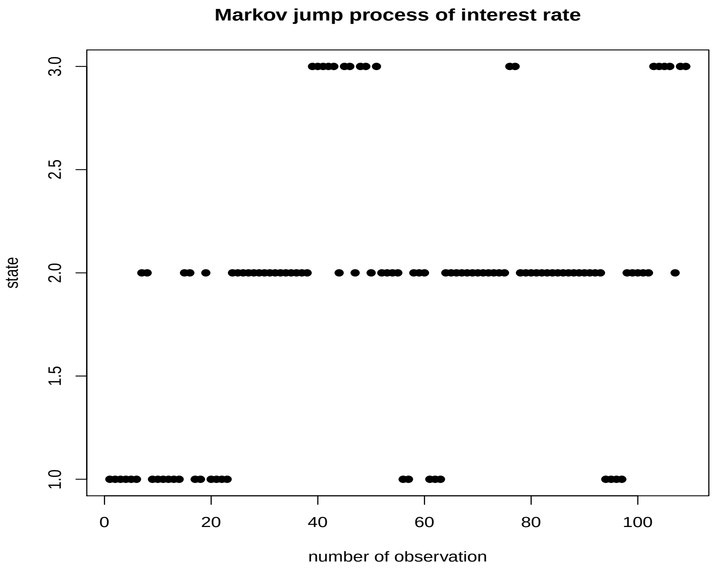


Figure 2.2: Discrete observations of Markov jump process for the interest rate.

The one-step transition probabilities of the discrete time Markov chain with step length 1 corresponding to Λ_{INT}, $P^{1.0} = exp^{1.0*\Lambda}$ is

$$P^1_{INT} = \begin{pmatrix} 0.67848292 & 0.2626306 & 0.05888615 \\ 0.10165440 & 0.7810266 & 0.11731905 \\ 0.08039944 & 0.3679999 & 0.55159950 \end{pmatrix}$$

and the stationary distribution is given by

$$\pi_{INT} = (0.2309447, 0.5853302, 0.1834741).$$

2.4.2 Credit rating data conditional on interest rate

We analyse credit rating data drawn from Moody's Corporate Bond Default Database. The data is a continuous record of the ratings of 696 issuers/firms in the period of January 1, 1990 to December 31, 1999. Data for the interest rate are the same as the view in the previous subsection

We assume that we have only annual data observations and we used only observations at discrete time points. Assuming that rating grades migrate according to 8-state Markov jump process in the period 1990-1999, we calculated the maximum likelihood estimators of the transition rates by MCMC method.

We present the result of a small simulation study, in which we simulated 10,000 intensity matrices including an initial burn-in of 1,000 iterations. The Metropolis-Hasting algorithm was run with an burn-in of 1,000 iterations. The values of the parameters in the prior were simply $\alpha_{lij} = \beta_{li} = 1$. The average of the 9,000 intensity matrices is

INTEREST RATE

$$\Lambda_{INT} = \begin{pmatrix} -3.3068023 & 2.890275 & 0.4165266 \\ 1.2488606 & -2.664666 & 1.4158056 \\ 0.5452789 & 3.451879 & -3.9971581 \end{pmatrix}.$$

The intensity matrix of Markov jump processes:

$$\Lambda_1 = \begin{pmatrix} -0.0663 & 0.0160 & 0.0020 & 0.0019 & 0.0051 & 0.0372 & 0.0021 & 0.0018 \\ 0.0703 & -0.1131 & 0.0102 & 0.0064 & 0.0065 & 0.0065 & 0.0065 & 0.0063 \\ 0.0178 & 0.0439 & -0.1515 & 0.0183 & 0.0180 & 0.0177 & 0.0182 & 0.0173 \\ 0.0057 & 0.0057 & 0.0056 & -0.1204 & 0.0494 & 0.0090 & 0.0338 & 0.0110 \\ 0.0093 & 0.0069 & 0.0071 & 0.0372 & -0.1505 & 0.0734 & 0.0077 & 0.0087 \\ 0.0336 & 0.0025 & 0.0025 & 0.0029 & 0.0231 & -0.0700 & 0.0027 & 0.0024 \\ 0.0589 & 0.0569 & 0.0586 & 0.1245 & 0.0652 & 0.0598 & -0.4993 & 0.0750 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \end{pmatrix},$$

if the underlying environment is in state 1 and

$$\Lambda_2 = \begin{pmatrix} -0.0457 & 0.0105 & 0.0008 & 0.0007 & 0.0041 & 0.0278 & 0.0008 & 0.0007 \\ 0.0493 & -0.0692 & 0.0064 & 0.0026 & 0.0027 & 0.0027 & 0.0026 & 0.0026 \\ 0.0071 & 0.0369 & -0.0805 & 0.0073 & 0.0072 & 0.0072 & 0.0073 & 0.0071 \\ 0.0028 & 0.0024 & 0.0024 & -0.1099 & 0.0582 & 0.0056 & 0.0300 & 0.0083 \\ 0.0057 & 0.0027 & 0.0027 & 0.0325 & -0.1180 & 0.0670 & 0.0030 & 0.0041 \\ 0.0235 & 0.0010 & 0.0011 & 0.0015 & 0.0213 & -0.0507 & 0.0010 & 0.0009 \\ 0.0191 & 0.0192 & 0.0184 & 0.0623 & 0.0244 & 0.0223 & -0.2137 & 0.0476 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \end{pmatrix},$$

if the underlying environment is in state 2.

$$\Lambda_3 = \begin{pmatrix} -0.0457 & 0.0105 & 0.0008 & 0.0007 & 0.0041 & 0.0278 & 0.0008 & 0.0007 \\ 0.0493 & -0.0692 & 0.0064 & 0.0026 & 0.0027 & 0.0027 & 0.0026 & 0.0026 \\ 0.0071 & 0.0369 & -0.0805 & 0.0073 & 0.0072 & 0.0072 & 0.0073 & 0.0071 \\ 0.0028 & 0.0024 & 0.0024 & -0.1099 & 0.0582 & 0.0056 & 0.0300 & 0.0083 \\ 0.0057 & 0.0027 & 0.0027 & 0.0325 & -0.1180 & 0.0670 & 0.0030 & 0.0041 \\ 0.0235 & 0.0010 & 0.0011 & 0.0015 & 0.0213 & -0.0507 & 0.0010 & 0.0009 \\ 0.0191 & 0.0192 & 0.0184 & 0.0623 & 0.0244 & 0.0223 & -0.2137 & 0.0476 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \end{pmatrix},$$

if the underlying environment is in state 3.

Interest matrix transition

$$P_{INT} = \begin{pmatrix} 0.2520553 & 0.5382590 & 0.2096855 \\ 0.2385578 & 0.5438604 & 0.2175818 \\ 0.2329981 & 0.5426968 & 0.2243050 \end{pmatrix}.$$

and the stationary distribution is given by

$$\pi_{INT} = (0.2405742, 0.5422064, 0.2171204).$$

We calculate the transition matrix for credit rating based on the stationary distribution of interest rate.

	Aaa	Aa	A	Baa	Ba	B	C	D
Aaa	0.9507	0.0102	0.0015	0.0016	0.0048	0.0278	0.0013	0.0016
Aa	0.0574	0.9085	0.0076	0.0052	0.0053	0.0061	0.0042	0.0053
A	0.0149	0.0405	0.8779	0.0139	0.0135	0.0140	0.0109	0.0139
Baa	0.0065	0.0051	0.0049	0.8863	0.0524	0.0094	0.0248	0.0101
Ba	0.0085	0.0050	0.0050	0.0313	0.8768	0.0616	0.0048	0.0066
B	0.0286	0.0021	0.0023	0.0027	0.0201	0.9399	0.0017	0.0021
C	0.0343	0.0305	0.0292	0.0838	0.0369	0.0345	0.6902	0.0601
D	0	0	0	0	0	0	0	1

Table 2.4: Annual transition probabilities based on the maximum likelihood estimate for the period 1990-1999.

2.5 Probability Default Backtesting.

A backtesting procedure typically evaluates the following characteristics of a rating system: calibration, discrimination and stability. The dataset used in this paper originates from a real life dataset covering 10 years (1990 to 1999). The last 2 years are used to run the backtest on, the first 8 years are used as a reference test.

2.5.1 Probability Default Calibration.

Correct calibration of a Probability Default rating system means that the calibrated PD estimates are accurate and conform to the observed default rates. Hence, when backtesting PD calibration, one will typically start with a matrix contrasting the estimated PD with the observed default rates for each rating and time period considered.

Binomial Test.

The binomial test contrasts the forecast default rate of a rating, $\hat{PD}$ versus the observed default rate, DR using following hypothesis test : $H_0 : PD = \hat{PD}$ vs $H_1 : PD > \hat{PD}$.

We assume that defaults occur independently and H_0 is true. In this case as the number of observations is small we evaluated Binomial distribution directly. Then, given the PD and n observations, the default rate DR follows a binomial distribution and we construct a confidence interval for DR. It can be shown that an exact confidence interval for the parameter is given by the following values:

$$p_1 = \frac{k}{(n-k+1)F_{\alpha/2,2(n-k+1),2k} + k}$$

and

$$p_2 = \frac{(k+1)F_{\alpha/2,2(k+1),2(n-k)}}{(n-k) + (k+1)F_{\alpha/2,2(k+1),2(n-k)}}$$

where $F_{\alpha/2,a,b}$ is the value of a distribution of Fisher-Snedecor F with a and b degrees of freedom left to right a probability of $\alpha/2$ for a confidence interval of $(1-\alpha)100\%$.

	$\hat{PD}$	p_1	p_2	H_0
Aaa	0.0016	0.0000	0.0020	accepted
Aa	0.0053	0.0000	0.0506	accepted
A	0.0139	0.0000	0.1322	accepted
Baa	0.0101	0.0090	0.1218	accepted
Ba	0.0066	0.0003	0.0729	accepted
B	0.0021	0.0000	0.0177	accepted
C	0.0601	0.0548	0.5717	accepted

Table 2.5: Confidence interval 95% for each credit rating.

Chapter 3

On Likelihood Inference for Diffusion-type Models

The aim of this chapter is to propose a method for obtaining maximum likelihood estimates of parameters in diffusion models when the data is a discrete time sample of the integral of the process, while no direct observations of the process itself are available.

3.1 Parametric Estimation of Diffusion Processes

In this section we consider parametric estimation problem for diffusion processes sampled at discrete times.

Diffusion processes are extensively used for modelling continuous time phenomena in many scientific areas. An incomplete list with some indicative references includes economics, biology, chemistry, physics and engineering. Their appeal lies in the fact that the model is built by specifying the instantaneous mean and variance of the process through a stochastic differential equation (SDE).

We consider the general one-dimensional diffusion process $X = \{X_t\}_{t \geq 0}$ given by the stochastic differential equation

$$dX_t = \mu(X_t; \psi)dt + \sigma(X_t; \psi)dW_t \tag{3.1}$$

where $W = \{W_t\}_{t \geq 0}$ is a standard Wiener process, and where the drift and diffusion coefficients depend on an unknown p-dimensional parameter ψ belonging to the parameter set $\Psi \subseteq \mathbb{R}^p$. The state space of the process is denoted by $I = (l, r)$, and $-\infty \leq l < r \leq \infty$ is an open set.

We assume that the solution X is an ergodic, stationary diffusion with invariant measure with density function $\nu_\psi(x)$ ($X_0 \sim \nu_\psi$ is independent of W). We also assume that the stochastic differential equation has a unique weak solution, i.e. a solution exists and all solutions have identical finite-dimensional distributions. It is well-known that sufficient conditions for these assumptions can be expressed in terms of the so-called scale function and speed measure by

$$\nu_\psi(x) = \frac{1}{M(\psi)\sigma^2(x,\psi)K(x,\psi)}$$

for some $x \in I$ and $M(\psi)$ the normalizing constant. Here, $K(x,\psi)$ is the scale measure and $m(x) = \nu_\psi(x)M(\psi)$ is the speed measure.

The exact dynamics of the diffusion process is governed by its transition density

$$p_{x,y}(t;\psi) = P(X_t \in dy|X_0 = x;\psi)/dy \tag{3.2}$$

3.1.1 Continuous observation.

We consider the infinitesimal generator of the diffusion X in the multidimensional parametric case. The operator L_ψ defined as

$$L_\psi(x,\psi) = \mu(x,\psi)\frac{\partial f(x,\psi)}{\partial x} + \frac{1}{2}\sigma^2(x,\psi)\frac{\partial^2 f(x,\psi)}{\partial x^2} \tag{3.3}$$

is called the infinitesimal generator of the diffusion, where f is a twice-continuous differentiable function $f : \mathbb{R} \times \psi \to \mathbb{R}$.

If the diffusion process has been continuously observeted as a function of x is quite straightforward to estimate the parameters efficiently. The part of parameter ψ in the diffusion coefficient can be calculated rather than estimated from the quadratic variation of the process since, for all $t \geq 0$,

$$\langle X, X\rangle_t = \lim_{n\to\infty}\sum_{k=1}^{2^n}(X_{t\wedge k/2^n} - X_{t\wedge(k-1)/2^n})^2 = \int_0^t \sigma^2(X_s,\psi)ds.$$

The rest of the parameters present only in the drift coefficient can be estimated using the maximum likelihood approach. Indeed, once the diffusion coefficient is independent of the parameter, which we just say is always true in principle (i.e., $\sigma(x,\psi) = \sigma(x)$) the likelihood function of X is given by

$$L_T(\psi) = \exp\Big(\int_0^T \frac{\mu(X_s,\psi)}{\sigma^2(X_s)}dX_s - \frac{1}{2}\int_0^T \frac{\mu^2(X_s,\psi)}{\sigma^2(X_s)}ds\Big). \tag{3.4}$$

Then ψ can be estimated by maximizing of $L_T(\psi)$.

3.1.2 Statistical Inference for discretely observed diffusion process.

Now, we suppose that we only data available from a realization of the diffusion process at times $t_1 < \cdots < t_n$, where $X_{t_i} = x_i$, $i = 1\ldots,n$. The time increments between consecutive observations will be denoted $\Delta_i = t_i - t_{i-1}$ for $1 \leq i \leq n$.

The aim is to estimate ψ given $\hat{X} = \{X_{t_1},\ldots,X_{t_n}\}$. In the likelihood context, estimation of ψ is based on the likelihood function of the data set X given by

$$L_n(\psi|\hat{X}) = \prod_{i=1}^{n} p_{X_{t_{i-1}},X_{t_i}}(\Delta_i;\psi)p(X_0;\psi), \tag{3.5}$$

where $t_0 = 0$ and $p_{X_t, X_{t+\Delta}}(\Delta; \psi)$ is the probability density function of the conditional distribution of $X_{t+\Delta}$ given that $X_t = x$.

We have that the log-likelihood function is

$$l_n(\psi|\hat{X}) = \sum_{i=1}^{n} \log(p_{X_{t_{i-1}}, X_{t_i}}(\Delta_i; \psi)) + \log(p(X_0; \psi)).$$

If the number of observations increases with time, we can assume that the relative weight of $p(X_0; \psi)$ in the whole likelihood $L_n(\psi|\hat{X})$ decreases, so we will assume that $p(X_0; \psi) = 1$ from now on without mentioning it any further.

Since the corresponding maximum likelihood estimator $\hat{\psi}$ for ψ is known in many cases to have the usual good properties, see [Billingsley (1961], classical likelihood inference about ψ based on $l_n(\psi|\hat{X})$ can for instance be performed for Gaussian diffusion processes, see [Pedersen (1995], but in general this is impossible since that transition densities of X are usually unknown, in this case the first approach was to perform the inference about ψ by the discretization of the likelihood function for ψ based on continuous observation of X.

If the transition density is differentiable, we define the score function

$$s_n(\psi|\hat{X}) = \begin{pmatrix} \sum_{i=1}^{n} \frac{\partial l_i(\psi)}{\partial \psi_1} \\ \vdots \\ \sum_{i=1}^{n} \frac{\partial l_i(\psi)}{\partial \psi_p} \end{pmatrix}$$

where $l_i(\psi) = p_{X_{t_{i-1}}, X_{t_i}}(\Delta_i; \psi)$ for $i = 1, 2 \ldots, n$ and the Fisher information matrix for ψ

$$MF_n(\psi) == \sum_{i=1}^{n} E_\psi(s_n(\psi|\hat{X})(s_n(\psi|\hat{X}))^t). \tag{3.6}$$

Exact likelihood inference.

In [Dacunha-Castelle & Florens-Zmirou (1986] sufficient conditions for the consistency and asymptotic normality of these maximum likelihood estimators are given. In particular consistency and asymptotic normality are proved, irrespective of the size of Δ, which seems a good property in applications. We present a smaller set of hypotheses that are the basic set used by many methods.

1. **Linear growth assumption.** There exists a constant K independent of ψ such that, for all x,

$$|\mu(x, \psi)| + |\sigma(x, \psi)| \leq K(1 + |x|).$$

2. **Global Lipschitz assumption.** There exists a constant K independent of ψ such that

$$|\mu(x, \psi) - \mu(y, \psi)| + |\sigma(x, \psi) - \sigma(y, \psi)| \leq K|x - y|.$$

3. **Positiveness of diffusion coefficient.**

$$inf_x \sigma^2(x, \psi) > 0$$

.

4. **Bounded moments.** For all $k > 0$, all moments of order k of the diffusion process exist and are such that

$$sup_t E(|X_t|)^k < \infty.$$

5. **Smoothness of the coefficients** The two coefficients μ and σ and their derivatives in ψ are smooth in x and of polynomial growth order in x uniformly on ψ.

This set of assumptions is also completed by technical conditions to ensure the proper rate of convergence and the existence of Fisher information of the experiment.

Inference for discretely observed diffusions has been pursued in many directions. In the rest of section we consider general methods for the statistical analysis of discretely observed diffusion processes. The principal methods for the estimation of unknown parameters in stochastic differential equation are presented.

Pseudo-likelihood methods.

We can use some approximation scheme for estimation of parameters. In this case we can not approximate the transition density directly but the path of the process in such a way that the discretized version of the process has a likelihood that is usable. We present the principal method of this type.

Euler method. We consider the general one-dimensional diffusion process $X = \{X_t\}_{t\geq 0}$ given by the stochastic differential equation

$$dX_t = \mu(X_t; \psi)dt + \sigma(X_t; \psi)dW_t.$$

In the case when the coefficients of the stochastic differential equation above are constant over small intervals $[t, t + \Delta_t)$, then the Euler scheme produces the discretization

$$X_{t+\Delta t} - X_t = \mu(X_t, \psi)\Delta t + \sigma(X_t, \psi)(W_{t+\Delta t} - W_t),$$

and the increments $X_{t+\Delta t}$ are then independent Gaussian random variables with mean $\mu(X_t, \psi)$ and variance $\sigma^2(X_t, \psi)\Delta t$. Therefore the transition density of the process can be written as

$$p_{x,y}(t|\psi) = \frac{1}{\sqrt{2\pi t \sigma^2(x, \psi)}} \exp(-\frac{(x - y - \mu(x, \psi)t)^2}{2t\sigma^2(x, \psi)}). \tag{3.7}$$

Then, under certain conditions, is possible obtain the maximum likelihood estimator of the parameters see [Yoshida (1992]. This approximation is good if Δt is very small.

Elerian method. This approximation was proposed in [Elerian (1998], the basic idea is use the transition density from the Milstein scheme, this approximation is good when the process has constan diffusion coefficient, for details see [Elerian (1998].

Local linearization methods. Another way to approximate the solution of stochastic differential equation is to use a local linearization method. There are two principals methods in this type of approximation, the Ozaki method for homogeneous stochastic differential equations and the Shoji-Ozaki method for the non-homogeneous case. Then, if the Ozaki method is for a homogeneous diffusion process with constant diffusion coefficient and the Shoji-Ozaki method return the value of transition density.

Approximated likelihood methods.

The are methods that differ from the previous in that they do not try to approximate the paths of a diffusion but instead provide direct approximation of the likelihood. The principal methods of this type are

1. **Kessel method.** Here, Kessel proposed to use of a higher-order *Itô*-Taylor expansion to approximate the mean and variance of the conditional density, see [Kessler (1997].

2. **Simulated likelihood method.** The idea in this method is as follows; Let $p_{x,y}(t|\psi)$ be the true transition density of $X_t + \Delta$ at point y given $X_t = x$. When the time step Δ is too large, we have seen that the Euler approximation usually gives a poor estimate of $p_{x,y}(\Delta|\psi)$. The idea is then to consider a smaller $\delta << \Delta$ and then use the Chapman-Kolmogorov equation as follows:

   $$p_{x,y}(\Delta|\psi) = \int p_{z,y}(\delta|\psi)p_{x,z}(\Delta - \delta|\psi) = E(_{z,y}(\delta|\psi)|\Delta - \delta).$$

   which means that $p_{x,y}(\Delta|\psi)$ is seen as the expected value over all possible transitions of the process from time $t + (\Delta - \delta)$ to $t + \Delta$, taking into account that the process was in x at time t. This method was proposed in [Pedersen (1995].

Solve Fokker-Planck equation.

Since that the Markovian property of equation (3.1) ensures that the transitional probability density function $p_{X_t,X_{t+\Delta}}(\Delta;\psi)$ satisfies the Fokker-Planck equation

$$\frac{\partial p}{\partial t} = \frac{\partial}{\partial x}(\frac{1}{2}\frac{\partial(\sigma^2(x;\psi)p)}{\partial x} - \mu(x;\psi)p) \tag{3.8}$$

with suitable initial and boundary conditions, then the maximum-likelihood estimation relies crucially on the ability to compute the value of the transitional probability density function through numerical solution of the Fokker-Planck equation, see details in [Jensen & Poulsen. (2002].

Hermite polynomial expansion approaches.

[Aït-Sahalia & Mykland. (2004] develops two estimation procedures in which the unknown transitional probability density function $p_{X_t, X_{t+\Delta}}(\Delta;\psi)$ is approximated by means of an expansion based on modified Hermite polynomials, for datails see [Aït-Sahalia & Mykland. (2004].

Bayesian estimation.

The basic idea a Bayesian estimator of a parameter ψ is obtained as the expected value of the posterior probability distribution of ψ,

$$p(\psi|x) = \frac{L_n(\psi)p(\psi)}{\int L_n(\psi)p(\psi)d\psi}.$$

where $p(\psi)$ is a prior distribution for ψ and x denotes the discrete-time observations from the diffusion process. Recently a new stream of results based on Markov chain Monte Carlo (MCMC) algorithms have been proposed. Approximations p were proposed by [Roberts & Stramer (2001], [Eraker (2001], [Beskos et al. (2006] and [Elerian, Chib & Shephard (2001] who used Markov chain Monte Carlo methods.

Approximation by Markov Chain Monte Carlo

Consider a the diffusion process given by equation (3.1) and a time discretization

$$0 = t_0 < t_1 < \ldots < t_n$$

Put

$$\Delta_i = t_{i+1} - t_i$$

$$\Delta W_i = Wt_{i+1} - W_{t_i}$$

Then $\Delta W_i \sim N(0, \Delta_i)$ and we can approximate X_t by the **Euler scheme**.

Under this approximation, the transition density $p_{X_{t_{i-1}}, X_{t_i}}(\Delta_i;\theta)$ is

$$f(X_{t_i}|X_{t_{i-1}},\theta) = \phi(X_{t_i}; X_{t_{i-1}} + \mu(X_{t_{i-1}};\psi)\Delta_i, \sigma^2(X_{t_{i-1}};\psi)\Delta_i). \tag{3.9}$$

where $\phi(u; a_1, a_2)$ denotes the density of the Normal distribution with mean a_1 and variance a_2 evaluated at u.

Now we assume for notational simplicity that the time gap $\Delta_i = \Delta^*$ is independent of t and consider between any two consecutive points (t_i, t_{i+1}), imputing M auxiliary points $X^*_{t_i} = (X^*_{t_{i,1}}, ..., X^*_{t_{i,M}})$ at times $t_{i,1} < ... < t_{i,M}$ with $\{t_{i,1}, \ldots, t_{i,M}\}$ between (t_i, t_{i+1}) and $X^*_{t_{i,j}} = x^*_{i,j}$ for $j = 1, \ldots, M$, denote the latent observation. We assume that the points are evenly spaced, with time gap

$$\Delta = t_{i,j+1} - t_{i,j} = \frac{\Delta^*}{M+1}$$

for all i, j.

Then, an approximation of the true transition density is given by

$$f^M(X_{t_i}|X_{t_{i-1}},\psi) = \int f(X_{t_i}|X^*_{t_{i-1,M}},\psi)[\prod_{j=2}^{M} f(X^*_{t_{i-1,j}}|X^*_{t_{i-1,j-1}},\psi)]$$

$$\times f(X^*_{t_{i-1,1}}|X_{t_{i-1}},\psi)dX^*_{t_{i-1,M}},...dX^*_{t_{i-1,1}}$$

$$= \int f(X_{t_i}|X^*_{t_{i-1}},\psi)f(X^*_{t_{i-1}}|X_{t_{i-1}},\psi)dX^*_{t_{i-1}} \quad (3.10)$$

where

$$f(X^*_{t_{i-1,j}}|X^*_{t_{i-1,j-1}},\theta) = \phi(X^*_{t_{i-1,j}};X^*_{t_{i-1,j-1}}+\mu(X^*_{t_{i-1,j-1}};\psi)\Delta,\sigma^2(X^*_{t_{i-1,j-1}};\psi)\Delta).$$

is the transition density using the Euler approximation. As $M \to \infty$, this transition density converges to the true transition density (see [Pedersen (1995]. One can, therefore, developed an estimation scheme that utilizes these auxiliary variables to minimize the error of the Euler discretization.

We have the problem that in general, the density $f^M(X_{t_i}|X_{t_{i-1}},\psi)$, cannot be computed exactly, but an effective way of dealing with this difficulty is to consider the joint posterior distribution of the parameters and the augmented data $X^* = (X^*_{t_1},\ldots,X^*_{t_n})$. To analyze the posterior density of $\psi, X^*|\hat{X}$ we can utilize Markov chain Monte Carlo this density is achieved by sampling in turn the full conditional distributions $X^*|\hat{X},\psi$, and $\psi|X^*,\hat{X}$.

Simulation of the auxiliary variables

Consider the question of sampling X^* from

$$f(X^*|X,\psi) = \prod_{i=1}^{T-1} f(X^*_{t_i}|X_{t_i},X_{t_{i+1}},\psi),$$

where we have used the fact that the $X^*_{t_i}$ are conditionally independent, given X_{t_i} and $X_{t_{i+1}}$. It therefore suffices to consider the simulation of $X^*_{t_i}$ from

$$f(X^*_{t_i}|X_{t_i},X_{t_{i+1}},\psi)$$

.

The target density of interest is

$$f(X^*_{t_i}|X_{t_i},X_{t_{i+1}},\psi) = \prod_{k=0}^{M} f(X^*_{t_i,k+1}|X^*_{t_i,k},\psi)$$

where $X^*_{t_{i,0}} = X_{t_i}$ and $X^*_{t_{i,M+1}} = X_{t_{i+1}}$. Each conditional density in this expression can be derived from the discrete-time Euler approximation.

A computationally effective approach for sampling $X^*_{t_i}$ from this density can be developed by working in sequence with contiguous subsets of $X^*_{t_i}$. Let $X^*_{t_i,(k,m)}$ denote a block of length m that starts at $X^*_{t_{i,k}}$ and ends at $X^*_{t_{i,k+m-1}}$, i.e.,

$$X^*_{t_i(k,m)} = (X^*_{t_{i,k}},X^*_{t_{i,k+1}},...,X^*_{t_{i,k+m-1}})$$

with density conditioned on $(X^*_{t_{i,k-1}}, X^*_{t_{i,k+m}}, \psi)$ given by

$$f(X^*_{t_i}|X_{t_i}, X_{t_{i+1}}, \psi) = \prod_{j=k-1}^{k+m} f(X^*_{t_{i,j+1}}|X^*_{t_{i,j}}, \psi).$$

The idea now is to sampler each of the m dimensional vectors $X^*_{t_i(k,m)}$ in sequence by the Metropolis-Hastings algorithm (see [Chib & Greenberg (1995]).

Then, let $q(X^*_{t_i(k,m)}|X^*_{t_i,k-1}, X^*_{t_i,k+m}, \psi)$ denote the proposal density conditioned on $(X^*_{t_i,k-1}, X^*_{t_i,k+m}, \psi)$ and suppose that the current value of $X^*_{t_i(k,m)}$ at the end of the nth iteration of Markov chain is $X^{*(n)}_{t_i(k,m)}$, then the Metropolis-Hasting step for $X^*_{t_i(k,m)}$ is implemented as follows

1. Drawing a candidate value $w \sim q(X^*_{t_i(k,m)}|X^*_{t_i,k-1}, X^*_{t_i,k+m}, \psi)$.

2. Accept the proposed value with probability

$$min(1, \frac{f(w|X^*_{t_i,k-1}, X^*_{t_i,k+m}, \psi)q(X^{*(n)}_{t_i(k,m)}|X^*_{t_i,k-1}, X^*_{t_i,k+m}, \psi)}{f(X^{*(n)}_{t_i(k,m)}|X^*_{t_i,k-1}, X^*_{t_i,k+m}, \psi)q(w|X^*_{t_i,k-1}, X^*_{t_i,k+m}, \psi)}).$$

Otherwise $X^{*(n+1)}_{t_i(k,m)} = X^{*(n)}_{t_i(k,m)}$.

, Since that the probability of moving is based only on ratios of densities, one does not need the normalizing constant of the target density.

Remark 3.1.1 *The Metropolis Hasting algorithm proposed here is referred to as an independence Metropolis Hasting sampler (see [Chib & Greenberg (1995]) because the proposal density does not depend on the current value of w. However, because the mean of the proposal depends on the immediate neighbours of w and because the underlying process for the observations is continuos, the mean of the proposal is typically close to w. Then the algorithm proposed will normally have high acceptance probability and is unlikely to get stuck in the tails of the target distribution.*

The proposal density A method of specifying such a proposal density is to approximate the target density at the mode a multivariate-normal or multivariate distribution with location given by the mode of $f(w|X^*_{t,k-1}, X^*_{t,k+m}, \psi)$, obtained by a few Newton-Raphson steps, and dispersion given by the negative of the inverse Hessian evaluated at the model.

To develop the proposal density let

$$X^*_{t(k,m)} = (X^*_{t,k}, X^*_{t,k+1}, ..., X^*_{t,k+m-1}) = (w_1, ..., w_m) = w$$

denote the block of latent values with neighbors $w_0 = X^*_{t,k-1}$ and $w_{m+1} = X^*_{t,k+m}$.

The approximate log density for w given its neighbours is then

$$l(w) = \log f(w|w_0, w_{m+1}) = c + \sum_{j=0}^{m} \log f(w_{j+1}|w_j) = c + \sum_{j=0}^{m} l_j$$

where l_j is given by the Euler approximation

$$l_j = \log f(w_{j+1}|w_j) = -(1/2)\log(\sigma^2\Delta) - \frac{(w_{j+1} - w_j(1 + \alpha\Delta))^2)}{2\sigma^2\Delta}$$

for $j = 0, ..., m$.

To deal with the non-Gaussian element of the likelihood, we propose to perform a second order Taylor expansion of $l(w)$ around a fixed value of w, denoted by $\hat{w} = (\hat{w_1}, ..., \hat{w_m})$ where the $(j+1)st$ element of $\hat{w}$ us given by

$$w_{\hat{j+1}} = w_0 + \frac{(j+1)(w_{m+1} - w_0)}{m+1}.$$

Then we have

$$l(w) \approx c + x^{'}u - \frac{1}{2}x^{'}Vx = c^{'} - \frac{1}{2}(x - V^{-1}u)^{'}V(x - V^{-1}u)$$

where $x = (w - \hat{w})$ and

$$u = \frac{\partial l(w)}{\partial w} = \{u_j\}$$

$$V = -\frac{\partial^2 l(w)}{\partial w \partial w'} = \{V_{ij}\}$$

for $i, j = 1, ..., m$.

denote the gradient and negative Hessian matrix respectively, of the log target density. Then $l(w)$ is approximately the log of a multivariate Gaussian density with mean $\tilde{\alpha} = \hat{w} + V^{-1}u$ and variance matrix $\tilde{\Sigma} = V^{-1}$. To obtain expressions for these parameters, define

$a_j = b(w_{j+1})$

$g_j = \frac{1}{\sigma^2(w_{j+1})}$

$d_j = X^*_{t,k+j+1} - (X^*_{t,k+j} + a_j\Delta) = w_{j+2} - (w_{j+1} + a_j\Delta)$

$c_j = 1 + a'\Delta$

for $j = -1, 0, ..., m-1$.

Now, we consider a given element, w_{j+1} of the block, the only terms in $l(w)$ involving w_{j+1} are

$l_j = \frac{1}{2}\log(\frac{g_{j-1}}{\Delta} - \frac{g_{j-1}}{2\Delta}(w_{j+1} - (w_j + a_{j-1}\Delta)]^2$

$l_{j+1} = \frac{1}{2}\log(\frac{g_j}{\Delta} - \frac{g_j}{2\Delta}(w_{j+2} - (w_{j+1} + a_j\Delta)]^2$

which implies that Bayesian estimation

$$\frac{\partial l(w)}{\partial w_{j+1}} = \frac{\partial l_j}{\partial w_{j+1}} + \frac{\partial l_(j+1)}{\partial w_{j+1}} = -\frac{1}{\Delta}(g_{j-1}d_{j-1} - g_j d_j c_j + \frac{1}{2}g_j^{'} d_j^2 - \frac{\Delta g_j^{'}}{2g_j})$$

and for the second derivatives we have

$$\frac{\partial^2 l(w)}{\partial w_{j+1}^2} = -\frac{1}{\Delta}(g_{j-1} - g_j c_j^2 - g_j d_j a_j^{''}\Delta - 2g_j^{'} d_j c_j + \frac{1}{2}g_j^{''} d_j^2 - \frac{\Delta(g_j g_j^{''} - (g_j^{'})^2)}{2g_j^2}).$$

The only term involving w_{j+1} and w_{j+2} is l_{j+1}, implying that

$$\frac{\partial^2 l(w)}{\partial w_{j+1}\partial w_{j+2}} = -\frac{1}{\Delta}(g_j^{'} d_j - g_j c_j).$$

In summary, the proposal density $q(w|w_0, w_{m+1})$ obtained from the second-order Taylor expansion of $l(w)$ is Gaussian with mean

$$\tilde{\alpha} = \hat{w} + V^{-1}u$$

and variance matrix

$$\tilde{\Sigma} = V^{-1}$$

where u is an $m \times 1$ vector with elements

$$u_{j+1} = \frac{\partial l_{j+1}}{\partial w_{j+2}} + \frac{\partial l_{j+2}}{\partial w_{j+2}}$$

for $j = 0, ..., m-1$, and V is an $m \times m$ matrix with elements

$$V_{j+1,j+1} = -\frac{1}{\Delta}\left[g_{j-1} - g_j c_j^2 - g_j d_j a_j''\Delta - 2g_j' d_j c_j + \frac{g_j'' d_j^2}{2} + \frac{(g_j g_{j-1}'' - (g_j')^2)\Delta}{2g_j^2}\right],$$

$$V_{j+1,j+2} = V_{j+2,j+1} = -\frac{1}{\Delta}(g_j' d_j - g_j c_j)$$

and

$$V_{j+1,j+s} = 0$$

for all j and $s > 2$.

Monte Carlo Methods, where the diffusion is simulated a large number of times has been studied in [Pedersen (1995] and [Durham & Gallant (2002].

EM Algorithm.

The EM-algorithm has been used in [Beskos et al. (2006] and [Bladt & Sørensen (2009]. Here, consider the general one-dimensional diffusion process $X = \{X_t\}_{t\geq 0}$ given by the stochastic differential equation (SDE)

$$dX_t = b(X_t; \alpha)dt + \sigma(X_t; \beta)dW_t \qquad (3.11)$$

where α and β are unknown parameters to be estimate, W is the standard Wiener process and we assume that $\sigma(x;\beta) > 0$ for all x in the state interval.

Now, we suppose that the data available from a realization of the diffusion process at times $t_1 < \cdots < t_n$, $x_i = X_{t_i}$, $i = 1\ldots,n$.

It is essential that we first transform the diffusion process (3.11) into an SDE of unit diffusion coefficient by applying the 1-1 transformation

$$X_s \rightarrow h(X_s;\beta) = Y_s,$$

where

$$h(x;\beta) = \int_{x^*}^{x} \frac{1}{\sigma(y;\beta)} dy \tag{3.12}$$

is any antiderivate of $\sigma^{-1}(*,\beta)$ and x^* is some arbitrary point (but appropriately chosen point in the state interval). We assume that $\sigma(*;\beta)$ is continuously differentiable and apply *Itô*'s formula to find that $Y_t = h(x;\beta)$, solves

$$dY_t = \mu(Y_t;\alpha,\beta)dt + dW_t, \tag{3.13}$$

where

$$\mu(y;\alpha,\beta) = \frac{b\{h^{-1}(y;\beta);\alpha\}}{\sigma\{h^{-1}(y;\beta);\beta\}} - \frac{\sigma'\{h^{-1}(y;\beta);\beta\}}{2}$$

Now, we have that in (3.13) the diffusion coefficient does not depend on the parameters, so the probability measures are equivalent, then the likelihood function can be found, see [Roberts & Stramer (2001] and [Beskos et al. (2006].

Finding the likelihood We define the function

$$g(x;\alpha,\beta) = s(x;\alpha,\beta) - \frac{log(\sigma(x;\beta))}{2},$$

where

$$s(x;\alpha,\beta) = \int_{x^*}^{x} \frac{b(z:\alpha)}{\sigma^2(z;\beta)} dz.$$

Remark 3.1.2 *We have that $\int_{y^*}^{y} \mu(z;\alpha,\beta) = g(h^{-1}(y;\beta);\alpha,\beta) - g(h^{-1}(y^*;\beta);\alpha,\beta)$ and the functions $g(*;\alpha,\beta)$ and $s(*,\alpha,\beta)$ are closely related to density $\phi(*;\alpha,\beta)$ of the stationary distribution of the original diffusion model given by (3.11), see [Bladt & Sørensen (2009]. Bayesian estimation*

Then if the stationary density is know, we only need find $h(*;\beta)$ and its inverse, see [Forman & Sørensen (2008].

To keep the original discrete time data fixed when running the EM-algorithm we define

$$Y_t^*(\beta,\tilde{\beta}) = Z_t^{(i,\tilde{\alpha},\tilde{\beta})} + \frac{(t_i - t)(h(x_{i-1};\beta) - h(x_{i-1};\tilde{\beta}) + (t - t_{i-1})(h(x_i;\beta) - h(x_i;\tilde{\beta})}{t_i - t_{i-1}},$$

for $t_{i-1} \leq t \leq t_i, i = 2,\ldots,n$, where $Z_t^{(i,\tilde{\alpha},\tilde{\beta})}$ denotes the $(t_{i-1}, h(x_{i-1};\tilde{\beta}), t_i, h(x_i;\tilde{\beta}))$-bridge for the diffusion given by equation (3.13), with parameter value $\tilde{\alpha}, \tilde{\beta}$ and

$Z_t^{(i,\tilde{\alpha},\tilde{\beta})}$ are independent for $i = 2, \ldots, n$.

Now, let $\tilde{\alpha}, \tilde{\beta}$ be the initial values of the parameters, the E-M algorithm is:

EM Algorithm

1. (E-step) Calculate the function

$$q(\alpha,\beta) = g(x_n;\alpha,\beta) - g(x_1;\alpha,\beta) - \frac{1}{2}\sum_{i=2}^{n} \frac{[h(x_i;\beta) - h(x_{i-1};\beta)]^2}{(t_i - t_{i-1})}$$

$$-\sum_{i=2}^{n} log(\sigma(x_i;\beta)) - \frac{1}{2}\sum_{i=2}^{n} E_{Z^{(i,\tilde{\alpha},\tilde{\beta})}}(\int_{t_{i-1}}^{t_i} [\mu'(Y_t^*(\beta,\tilde{\beta};\alpha,\beta)) + \mu(Y_t^*(\beta,\tilde{\beta};\alpha,\beta))^2]dt).$$

2. (M-step) $\tilde{\alpha}, \tilde{\beta} = argmax_{\alpha,\beta} q(\alpha,\beta)$.

3. go to 1.

Here $E_{Z^{(i,\tilde{\alpha},\tilde{\beta})}}$ means that the data points are fixed so that only the diffusion bridge is random, and the expectation is with respect to the distribution of the diffusion bridge. The expectations are approximated by simulating independent diffusion bridges.

Following [Beskos et al. (2006], the conditional expectation can be calculated as

$$E_{Z^{(i,\tilde{\alpha},\tilde{\beta})},U}(\mu'_{\alpha,\beta}(Y_U^*(\beta,\tilde{\beta})) + \mu_{\alpha,\beta}(Y_U^*(\beta,\tilde{\beta}))^2),$$

where $U \sim U_{[t_{i-1},t_i]}$ independent of $Z^{(i,\tilde{\alpha},\tilde{\beta})}$ and the data. If the drift parameter of the diffusion given by equation (3.11) can be written as a linear combination of the vector of parameters α

$$b(x;\alpha) = \alpha_1 a_1(x) + \ldots \alpha_k a_k(x)$$

then the maximization is less complicated if also the diffusion parameter β is fixed, the model for continuous time observations of X as well as the transformed process Y is an exponential family of the stochastic process, see [Kuchler & Sørensen (1999]. In this case for the E-M algorithm we have

$$q(\alpha,\beta) = \sum_{i=1}^{k} \alpha_i H(*;i,\beta) - \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k} \alpha_i\alpha_j B(*;i,j,\beta) + G(*,\beta),$$

with

$$H(*;i,\beta) = s(x_n;i,\beta) - s(x_1;i,\beta)$$

$$+\sum_{j=2}^{n} E_Z(\int_{t_{j-1}}^{t_j} [a_i(h^{-1}(Y_t^*(\beta,\tilde{\beta});\beta))(log\sigma(*;\beta)'(h^{-1}(Y_t^*(\beta,\tilde{\beta});\beta))) - \frac{1}{2}a_i'(h^{-1}(Y_t^*(\beta,\tilde{\beta});\beta))]dt),$$

where

$$s(x;i,\beta) = \int_{x^*}^{x} \frac{a_i(y)}{\sigma^2(y;\beta)}dy,$$

$$B(*;i,j,\beta)=\sum_{j=2}^{n}E_{Z^{(i,\tilde{\alpha},\tilde{\beta})}}(\int_{t_{j-1}}^{t_j}[\frac{a_i(h^{-1}(Y_t^*(\beta,\tilde{\beta});\beta))a_j(h^{-1}(Y_t^*(\beta,\tilde{\beta});\beta)))}{\sigma^2(h^{-1}(Y_t^*(\beta,\tilde{\beta});\beta))}]dt),$$

and

$$G(*;\beta)=-\frac{1}{2}log(\sigma(x_n;\beta)/\sigma(x_1;\beta))-\frac{1}{2}\sum_{i=2}^{n}\frac{[h(x_i;\beta)-h(x_{i-1};\beta)]^2}{t_i-t_{i-1}}-\sum_{i=2}^{n}log(\sigma(x_i;\beta))$$

$$+\frac{1}{4}\sum_{j=2}^{n}E_{Z^{(i,\tilde{\alpha},\tilde{\beta})}}(\int_{t_{j-1}}^{t_j}[\sigma''(h^{-1}(Y_t^*(\beta,\tilde{\beta});\beta))\sigma(h^{-1}(Y_t^*(\beta,\tilde{\beta});\beta))-\frac{1}{2}(\sigma(h^{-1}(Y_t^*(\beta,\tilde{\beta});\beta)))^2]dt).$$

If β is fixed, then the function $\alpha\rightarrow q(\alpha,\beta)$ is maximal for

$$\hat{\alpha}(\beta)=B^{-1}(*;\beta)H(*;\beta),$$

with $\hat{\alpha}=(\hat{\alpha_1},\ldots,\hat{\alpha_k})^T, H(*;\beta)=(H_1(*;\beta),\ldots,H_k(*;\beta))^T$ and $B(*;\beta)=\{B(*;i,j,\beta)\}$. And $B(*;\beta)$ is invertible if the functions $a_i, i=1,\ldots,k$ are linearly independent, then $q(\alpha,\beta)$ attains its maximal value at $(\hat{\alpha}(\hat{\beta}),\hat{\beta})$ and $\hat{\beta}$ maximizes

$$\beta\rightarrow q(\hat{\alpha}(\beta),\beta)=\frac{1}{2}H^T(*;\beta)B^{-1}(*;\beta)H(*;\beta)+G(*;\beta).$$

Simulated Data Example In this section we apply the method presented above to the Ornstein-Uhlenbeck process, which is a solution of the stochastic differential equation

$$dX_t=-\alpha X_t d_t+\sigma dW_t,$$

where α and σ are unknown parameters to be estimates, and W is the standard Wiener process.

Assume that $\sigma>0$ for all x in the state interval.

Consider the transformation (3.12) we have

$$h(x;\sigma)=\int_{x^*}^{x}\frac{1}{\sigma(y)}dy=\frac{x}{\sigma},$$

with $x^*=0$. If we use the *Itô*'s formula, then $Y_t=\frac{x}{\sigma}$ solves the SDE

$$dY_t=-\alpha Y_t d_t+dW_t.$$

then $\mu(y;\alpha,\sigma)=-\alpha y$.

The likelihood

Doing some simple calculations we have

$$s(x;\alpha,\sigma)=\frac{-\alpha x^2}{2\sigma^2},$$

and hence

$$g_{\alpha,\sigma}(x) = \frac{-\alpha x^2}{2\sigma^2} - \frac{1}{2}\log(\sigma).$$

Let be $Z_{i,j} = Z_j^{(i,\tilde{\alpha},\sigma_0)}$ the jth point of the bridge $(t_{i-1}, \frac{X_{i-1}}{\sigma_0}, t_i, \frac{X_i}{\sigma_0})$, with $j = 1, ..., M$, $i = 2, ..., n$, M is the number of points of the bridge. In addition we define $Z_{i,0} = Y_{i-1}$ and $Z_{i,M+1} = Y_i$ for all $i = 2, ..., n$, then we can define

$$Y_j^*(\sigma, \sigma_0) = Z_{i,j} + \frac{1}{t_i - t_{i-1}} \left(\frac{\Delta X_{i,j}}{\sigma} - \frac{\Delta X_{i,j}}{\sigma_0} \right),$$

where $\Delta X_{i,j} = X_{i-1}(1 - j\Delta) + j\Delta X_i$ for $i = 2, ..., n$, $j = 0, 1, ..., M + 1$ and $\Delta = \frac{1}{M+1}$.

The EM algorithm works as follow.

E-STEP

We calculate the function

$$q(\alpha, \sigma) = \frac{\alpha(X_1^2 - X_n^2)}{2\sigma^2} - \frac{1}{2}\sum_{i=2}^{n} \frac{(X_i - X_{i-1})^2}{\sigma^2(t_i - t_{i-1})}$$

$$-\sum_{i=2}^{n} \log(\sigma) - \frac{1}{2}\sum_{i=2}^{n} E_{Z_t^{(i,\tilde{\alpha},\sigma_0)}} \left(\int_{t_{i-1}}^{t_i} -\alpha + \alpha^2 (Y_t^*(\sigma, \sigma_0))^2 dt \right)$$

. where

$$\begin{aligned} \int_{t_{i-1}}^{t_i} \alpha^2 (Y_t^*(\sigma, \sigma_0))^2 dt &= \alpha^2 \sum_{j=0}^{M} \int_{t_{i-1}+j\Delta}^{t_{i-1}+(j+1)\Delta} (Y_{j+1}^*)^2 dt \\ &= \alpha^2 \sum_{j=0}^{M} \int_{t_{i-1}+j\Delta}^{t_{i-1}+(j+1)\Delta} a_{1,i,j} + \frac{a_{2,i,j}}{\sigma^2} + \frac{a_{3,i,j}}{\sigma} dt \\ &= \alpha^2 \Delta (a_{1,i,.} + \frac{a_{2,i,.}}{\sigma^2} + \frac{a_{3,i,.}}{\sigma}) \end{aligned}$$

where

$$a_{1,i,j} = (Z_{i,j+1}^2 - \frac{\Delta X_{i,j}}{\sigma_0})^2,$$

$$a_{2,i,j} = (\Delta X_{i,j})^2$$

$$a_{3,i,j} = 2Z_{i,j+1}\Delta X_{i,j} - \frac{2(\Delta X_{i,j})^2}{\sigma_0},$$

and

$$a_{k,i,.} = \sum_{j=0}^{M} \int_{t_{i-1}+j\Delta}^{t_{i-1}+(j+1)\Delta} a_{k,i,j}$$

for $k = 1, 2, 3$ and $i = 2, ..., n$, and taking $t_i - t_{i-1} = 1$.

Then we can write

$$q(\alpha, \sigma) = \frac{a\alpha}{\sigma^2} - \frac{f}{\sigma^2} - b\log(\sigma) + \frac{b\alpha}{2} - \frac{\alpha^2}{2}(a_1 + \frac{a_2}{\sigma^2} + \frac{a_3}{\sigma})$$

where

$$a = \frac{X_1^2 - X_n^2}{2},$$
$$b = n - 1,$$
$$f = \frac{1}{2}\sum_{i=2}^{n} \frac{(X_i - X_{i-1})^2}{t_i - t_{i-1}}$$

and

$$a_k = \sum_{i=2}^{n} a_{k,i,\cdot}.$$

for $k = 1, 2, 3$ and $i = 2, ..., n$.

M-STEP

We get the following partial derivatives

$$\frac{\partial q(\alpha,\sigma)}{\partial \alpha} = \frac{a}{\sigma^2} + \frac{b}{2} - \alpha(a_1 + \frac{a_2}{\sigma^2} + \frac{a_3}{\sigma}), \tag{3.14}$$

and

$$\frac{\partial q(\alpha,\sigma)}{\partial \sigma} = -2\frac{\alpha a}{\sigma^3} + 2\frac{f}{\sigma^3} - \frac{b}{\sigma} + \frac{a_2\alpha^2}{\sigma^3} + \frac{a_3\alpha^2}{2\sigma^2}. \tag{3.15}$$

From equations (3.14) and (3.15) we get the follow equations system

$$a + \frac{b\sigma^2}{2} - \alpha(a_1\sigma^2 + a_2 + a_3\sigma) = 0 \tag{3.16}$$

$$4a\alpha - 4f + 2b\sigma^2 - \alpha^2(2a_2 + a_3\sigma) = 0 \tag{3.17}$$

Thus from equation (3.16), we get

$$\tilde{\alpha} = \frac{a + \frac{b\sigma^2}{2}}{a_1\sigma^2 + a_3\sigma + a_2)}, \tag{3.18}$$

Now replacing expression (3.18) in equation3.17 we get the polynomial

$$c_6\sigma^6 + c_5\sigma^5 + c_4\sigma^4 + c_3\sigma^3 + c_2\sigma^2 + c_1\sigma + c_0 = 0 \tag{3.19}$$

where
$c_6 = a_1^2$,
$c_5 = 2a_1a_3 - \frac{ba_3}{8}$,
$c_4 = aa_1 - \frac{ba_2}{4} - \frac{2a_1^2 f}{b} + a_3^2 + 2a_1a_2$,
$c_3 = \frac{aa_3}{2} - \frac{4fa_1a_3}{b} + 2a_2a_3$,
$c_2 = \frac{2a^2a_1}{b} - \frac{2fa_3}{b} - \frac{4f_1a_2}{b} + a_2^2$,
$c_1 = \frac{3a^2a_3}{2b} - \frac{4fa_3a_2}{b}$,
$c_0 = \frac{a^2a_2}{b} - \frac{2fa_2^2}{b}$.

the solution of this polynomial is $\tilde{\sigma}$.

EM Output Analysis.

Here, the Milstein scheme was used to simulate a Ornstein-Uhlenbeck process with discretization level (n=1500). The method developed by [Bladt & Sørensen (2009] was used to build the Ornstein-Uhlenbeck bridges.

Summaries of the EM output for the simulated data set are reported in Table OU-I. The EM algorithm is run for N=10,000 iterations. The real values of the parameters are $\alpha = 0.1$,$\sigma = 0.5$

M	α	σ
10	0.1027678	0.4996258
25	0.09914815	0.4988454
50	0.0967	0.4971465

Table OU-I.

Summaries of the EM output for the simulated data set are reported in Table OU-II with discretization level (n=3000).

M	α	σ
10	0.1166107	0.5127427
25	0.1119799	0.5116187
50	0.110897	0.5093764

Table OU-II.

Real Data Example We now consider the analysis of a diffusion model specified by a Ornstein-Uhlenbeck process and we used the climate data which consist of 3000 observations.

Summaries of the EM output for the simulated data set are reported in Table OU-III . The EM algorithm is run with M=30,50, for N=10,000 iterations.

Parameter	M=30	M=50
α	0.047059234	0.046296176
σ	0.9415721	0.9411101

Table OU-III.

Sample DNA matching procedures.

There another kind of methods which may be loosely labelled as sample DNA matching procedures. This encompass methods which differ greatly in their mode of implementation, but which all have in common the fact that they attempt to match some feature or characteristic of the data to a theoretical counterpart of the model by choice of parameters. A pertinent case in point is estimation based on the characteristic function.

The principal methods in this category are

1. **General Method of Moments**, see [Hansen & J.Scheinkman (1995].
2. **Indirect Estimation**, see [Gallant & Tauchen (1996].
3. **Characteristic Function**, see [Chacko & Viceira (2003].

4. **Match to Marginal Density**, see [Aït-Sahalia (1996].

5. **Estimating Functions**, martingale estimating functions for discretely observed diffusions are reviewed in [Sørensen (1997] and [Sørensen (2010].

Here, we present an outline of the method developed in [Sørensen (2000].

Estimating functions.

Prediction Based Estimating Functions

An estimating is a function $G_n(\theta)$ that depends on the parameter as well as on the observations. We can obtain an estimator by solving the equation

$$G_n(\theta) = 0.$$

Defined the one dimensional functions $f_j, j = 1, \ldots, m$ on the state space of $\hat{X}$, such that $E_\theta(f_j(X_i)^2) < \infty$ with $i = 1, \ldots, n$ and we denote the expectation when θ is the true parameter value by $E_\theta(*)$. Define $\mathcal{F}_i := \sigma_{k \leq i}(X_k)$ and $\mathcal{H}_i^\theta = L^2(\mathcal{F}_i)$ and let $P_{i,j}^\theta, j = 1, \ldots, m$ be closed linear subspaces of $\mathcal{H}_i^\theta$. The subspace $P_{i,j}^\theta$ can be interpreted as a set predictors of $f_j(X_{i+1})$ given $X_1, \ldots, X_i$. We will study the estimating function

$$G_n(\theta) = \sum_{i=1}^{n} \sum_{j=1}^{n} \Pi_j^{i-1}(\theta)(f_j(X_i) - \hat{\pi}_j^{i-1}(\theta)),$$

with $\Pi_j^{i-1}(\theta) = (\pi_{1,j}^{i-1}(\theta), \ldots, \pi_{p,j}^{i-1}(\theta))^T$ is a p-dimensional stochastic vector, the coordinates of which belong to $P_{i-1,j}^\theta$ and where $\hat{\pi}_j^{i-1}(\theta)$ is the minimun mean square error predictor of $f_j(X_i)$ in $P_{i-1,j}^\theta$, then $\hat{\pi}_j^{i-1}(\theta)$ is the orthogonal projection of $f_j(X_i)$ on $P_{i-1,j}^\theta$ with respect to the inner product in $\mathcal{H}_i^\theta$. The projection exist and is uniquely determined by the Normal equations

$$E_\theta(\pi(f_j(X_i) - \hat{\pi}_j^{i-1}(\theta))) = 0$$

with $\pi \in P_{i-1,j}^\theta$.

We shall be particularly interested in prediction-based estimating functions where each of the sets $P_{i-1,j}^\theta$ is finite dimensional. We assume that $P_{i-1,j}^\theta$ is spanned by $Z_{j0}^{i-1}, Z_{j1}^{i-1}, \ldots, Z_{jq_{ij}}^{i-1}$ of the form $Z_{jk}^{i-1} = h_{jk}^i(X_1, \ldots, X_{i-1}), k = 1, \ldots, q_{ij}$, which are linearly independent in $\mathcal{H}_{i-1}^\theta$. Assume that Z_{j0}^{i-1} is constantly equals to 1 and thus we can write the elements of $P_{i-1,j}^\theta$ in the form $a^T Z_j^{i-1}$ with $a^T = (a_0, \ldots, a_{q_j})$ and $Z_j^{i-1} = (Z_{j0}^{i-1}, Z_{j1}^{i-1}, \ldots, Z_{jq_{ij}}^{i-1})^T$. When θ is the true parameter value, we define $C_j(\theta)$ as the covariance matrix of Z_j^{i-1} and $b_j(\theta) = (Cov_\theta(Z_{j1}^r, f_j(X_{r+1})), \ldots, Cov_\theta(Z_{jq_j}^r, f_j(X_{r+1})))^T$ and by the Normal equations, the minimum mean square error predictor of $f_j(X_i)$ in $P_{i-1,j}^\theta$ is given by

$$\hat{pi}_j^{i-1}(\theta) = \hat{a}_{j0}(\theta)^T + \hat{a}_j(\theta)^T Z_j^{i-1}$$

where $\hat{a}_j(\theta)^T = C_{i-1,j}(\theta)^{-1} b_j^{i-1}(\theta)$

and $\hat{a}_{j0}(\theta)^T = E_\theta(f_j(X_i)) - \hat{a}_j(\theta)^T E_\theta(Z_j^{i-1})$

3.2 Maximum likelihood estimation for integrated diffusion processes.

In this section we propose a method for obtaining maximum likelihood estimates of parameters in diffusion models when the data is a discrete time sample of the integral of the process, while no direct observations of the process itself are available. The data are, moreover, assumed to be contaminated by measurement errors.

3.2.1 Introduction

We consider maximum likelihood estimation in the situation where we do not observe the process X itself directly, but instead observe integrals of the process over disjoint time-intervals. These observations are, moreover, assumed to be contaminated by measurement errors.

The likelihood function for a discretely sampled integrated diffusion with observation error is in almost all cases not explicitly available. Moreover, the integrated process is not a Markov process, so there is no easily calculated martingales. Therefore martingale estimating functions are not a feasible alternative, but prediction-based estimating function can be applied, see [Sørensen (2000]. We note instead that the data can be viewed as incomplete observations from a model with a tractable likelihood function. The full data set is a continuous time record of the diffusion process and the observation errors.

We can therefore find maximum likelihood estimates by applying the Expectation-Maximization (EM) algorithm, see [Dempster, Laird & Rubin (1977]. To do so we need to calculate the conditional expectation of the log-likelihood function for the full model given the observations. We do this by simulating sample paths of the diffusion process given the data using ideas from [Chib, Pitt & Shephard (2006]. An essential step in doing this is to simulate a part of a sample path given the rest, which corresponds to simulation a diffusion bridge. This is done by applying the method for approximate diffusion bridge simulation recently proposed by [Bladt & Sørensen (2009].

3.2.2 Model and data

We consider likelihood estimation the general one-dimensional diffusion process $X = \{X_t\}_{t\geq 0}$ given by the stochastic differential equation

$$dX_t = \mu(X_t;\psi)dt + \sigma(X_t;\psi)dW_t \tag{3.20}$$

where $W = \{W_t\}$ is a standard Wiener process, and where the drift and diffusion coefficients depend on an unknown p-dimensional parameter ψ belonging to the parameter set $\Psi \subseteq \mathbb{R}^p$. We assume that the solution X is an ergodic, stationary diffusion with invariant measure with density function $\nu_\psi(x)$ ($X_0 \sim \nu_\psi$ is independent of W). We also assume that the stochastic differential equation has a unique weak solution, i.e., a solution exists and all solutions have identical finite-dimensional distributions; see e.g. [Karatzas & Shreve (1991]. It is well-known that sufficient conditions for these assumptions can be expressed in terms of the so-called scale function and speed measure.

In this section we consider the situation where the process X has not been observed directly. Instead the data are integrals of X_t over intervals $[t_{i-1}, t_i]$ observed with measurement error, i.e.

$$Y_i = \int_{t_{i-1}}^{t_i} X_s ds + Z_i, \quad i = 1, \ldots, n, \tag{3.21}$$

where $Z_i \sim N(0, \tau^2)$, $i = 1, \ldots, n$ are mutually independent and independent of X.

We assume that $t_0 = 0$, so the total interval of observation is $[0, t_n]$. Note that the variance of the measurement error, τ^2, is an extra unknown parameter.

Thus we now need to estimate the $p + 1$-dimensional parameter $\theta = (\psi, \tau^2)$.

Conditionally on the sample path of X, the observations Y_i, $i = 1, \ldots, n$ are independent and Normal distributed:

$$Y_i \,|\, X_t : t \in [0, t_n] \sim N\left(\int_{t_{i-1}}^{t_i} X_s ds, \tau^2 \right), \tag{3.22}$$

We assume that the coefficients of the stochastic differential equation (3.20) satisfy the following conditions which we need in the following sections.

Condition 3.2.1 *The drift and diffusion coefficients of (3.20), $\mu(x; \psi)$ and $\sigma(x; \psi)$ satisfy that for all $\psi \in \Psi$*

$\mu(x; \psi)$ *is continuously differentiable with respect to x*

$\sigma(x; \psi)$ *is twice continuously differentiable with respect to x*

$\sigma(x; \psi) > 0$ *for all x in the state space of X*

3.2.3 The likelihood function and the EM-Algorithm

We can think of the data set $Y = (Y_1, \ldots, Y_n)$ as an incomplete observation of a full data set given by the sample path $X_t, t \in, [0, t_n]$ and the measurement errors $Z_1, \cdots, Z_n$, or equivalently X_t, $t \in [0, t_n]$ and $Y = (Y_1, \ldots, Y_n)$. Therefore likelihood based estimation can be done by means of the EM-algorithm or MCMC-methods. In this section we concentrate on the EM-algorithm.

We need to find the likelihood function for the full data set and the conditional expectation of this full log-likelihood function given the observations $Y = (Y_1, \ldots, Y_n)$.

Likelihood with full diffusion observation

The full observation of a diffusion sample path in the time interval $[0, t_n]$ is an element in the space $\mathscr{C}$ of continuous functions from $[0, t_n]$ to $\mathbb{R}$. We equip this space with the usual σ-algebra, $\mathcal{C}$, generated by the cylinder sets, and consider the probability measures induced on $(\mathscr{C}, \mathcal{C})$ by the solutions to (3.20). These measures are in general singular because the diffusion coefficient depends on the parameter ψ.

In order to obtain a likelihood function, we use the Lamperti transformation

$$h(x;\psi) = \int_{x^*}^{x} \frac{1}{\sigma(u;\psi)} du, \tag{3.23}$$

where x^* is some arbitrary element of the state space of X. By this parameter dependent transformation, we obtain a diffusion process with unit diffusion coefficient. Specifically, we obtain (by $It\hat{o}$'s formula) that

$$U_t = h(X_t;\psi)$$

satisfies the stochastic differential equation

$$dU_t = \alpha(U_t;\psi)dt + dW_t, \tag{3.24}$$

with

$$\alpha(u;\psi) = \frac{\mu\left(h^{-1}(u;\psi);\psi\right)}{\sigma\left(h^{-1}(u;\psi);\psi\right)} - \frac{\sigma'\left(h^{-1}(u;\psi);\psi\right)}{2},$$

where σ' denotes the derivative of σ with respect to x.

Note that in (3.24) the diffusion coefficient does not depend on the parameters, so the probability measures induced on $(\mathscr{C}, \mathcal{C})$ by the solution to (3.24) are equivalent and the likelihood function can be found.

We can express the observations Y_i in terms of the process U. By inserting $X_s = h^{-1}(U_s;\psi)$ in (3.21), we find that

$$Y_i = \int_{t_{i-1}}^{t_i} h^{-1}(U_s;\psi)ds + Z_i, \quad i = 1,\ldots,n.$$

Therefore we will think of the full dataset as $U_t,\ t \in [0,t_n]$ and $Y = (Y_1,\ldots,Y_n)$. Since conditionally on the sample path of U the observations $Y_i,\ i,\ldots,n$ are independent, we have that the likelihood of Y conditional on the sample path of U in $[0,t_n]$ is

$$L(Y_1,\ldots,Y_n \,|\, U_t, t \in [0,t_n]) = \prod_{i=1}^{n} \phi(Y_i; \int_{t_{i-1}}^{t_i} h^{-1}(U_s;\psi)ds, \tau^2) \tag{3.25}$$

Let P_ψ be the probability measure induced by $U = \{U_t\}_{t\in[0,t_n]}$ on $(C, \mathcal{C})$, i.e. the probability measure with respect to which the coordinate process has the same distribution as U, and let Q be the Wiener measure on $(C, \mathcal{C})$. We assume that the coefficient μ satisfies conditions ensuring that the Girsanov Theorem holds so that we have the Radon-Nykodym derivative

$$\frac{dP_\psi}{dQ}(B) = \exp\left\{\int_0^{t_n} \alpha(B_t;\psi)dB_t - \frac{1}{2}\int_0^{t_n} \alpha^2(B_t;\psi)dt\right\}. \tag{3.26}$$

The evaluation of $\frac{dP_\psi}{dQ}$ is difficult because of the *Itô* integral term. To simplify the likelihood function, we apply the transformation

$$a(x;\psi) = \int^x \alpha(u;\psi)du$$

(any antiderivative of μ), which under Condition 3.2.1 is twice continuously differentiable. By *Itô*'s formula

$$\int_0^{t_n} \alpha(B_t)dB_t = a(B_{t_n};\psi) - a(B_0;\psi) - \frac{1}{2}\int_0^{t_n} \alpha'(B_t;\psi)dt,$$

where α' denotes the derivative of $\alpha(u;\psi)$ w.r.t. u.

We can now write the likelihood function (3.26) as

$$\frac{dP_\psi}{dQ}(B) = \exp\left\{a(B_{t_n};\psi) - a(B_0;\psi) - \frac{1}{2}\int_0^{t_n}[\alpha(B_t;\psi)^2 + \alpha'(B_t;\psi)]dt\right\}.$$

By combining this expression and (3.25), we see that the log-likelihood function for θ based on the full data set $U_t,\ t \in [0, t_n]$ and $Y = (Y_1, \ldots, Y_n)$ is given by

$$\begin{aligned}\log L(\theta; Y_1, \ldots, Y_n, U_t, t \in [0,t_n]) \quad &= \quad \sum_{i=1}^n \log\phi(Y_i; \int_{t_{i-1}}^{t_i} h^{-1}(U_s;\psi)ds, \tau^2) \\ &+ a(U_{t_n};\psi) - a(U_0;\psi) - \frac{1}{2}\int_0^{t_n}\left(\alpha(U_t;\psi)^2 + \alpha'(U_t;\psi)\right)dt.\end{aligned}$$

EM Algorithm.

We can now apply the EM-algorithm to the full log-likelihood function to obtain the maximum likelihood estimate of the parameter θ.

As the initial value for the algorithm, let $\hat{\theta}$ be any value of the parameter vector $\theta = (\psi, \tau^2) \in \Psi \times (0, \infty)$. Then the EM-algorithm works as follow.

1. **E-STEP.**

   Generate M sample paths of the diffusion process X, $X^{(k)}, k = 1, \ldots, M$, conditional on the observations $Y_1, \ldots, Y_n$ using the parameter value $\hat{\theta} = (\hat{\psi}, \hat{\tau}^2)$, and calculate

   $$g(\theta) = \frac{1}{M - M_0}\sum_{k=M_0+1}^{M} \log L(\theta; Y_1, \ldots, Y_n, h(X_t^{(k)}; \hat{\psi}), t \in [0, t_n]),$$

   for a suitable burn-in period M_0 and M sufficiently large.

2. **M-STEP.**

   $\hat{\theta} = \text{argmax } g(\theta)$.

3. Go to 1.

To implement this algorithm, the main issue is how to generate sample paths of X conditionally on $Y_1, \ldots, Y_n$, where the relation between the Y_is and X is given by (3.21). The algorithm must produce a sequence $X^{(k)}, k = 1, \ldots, M$, that is sufficiently mixing to ensure that $g(\theta)$ approximates the conditional expectation of the full log-likelihood function (3.27) given the data. This problem is discussed at the next section.

3.2.4 Conditional diffusion process simulation.

In this section we present a method for generating a sample from

$$\{X_t; t \in [0, t_n]\} | (Y_1, \ldots, Y_n)$$

for a given value of the parameter vector θ, i.e., for simulating the diffusion X conditional on the observations $Y = (Y_1, \ldots, Y_n)$ of integrals of X over subintervals $[t_{j-1}, t_j]$, $j = 1, \ldots, n$ perturbed by measurement errors.

This can be done by means of a Metropolis-Hastings algorithm, see e.g. [Chib & Greenberg (1995] or [Gilks, Richardson & Spiegelhalter (1996]. However, if the sample path in the entire time interval $[0, t_n]$ is updated in one step, the rejection probability is typically very large. Therefore it is more efficient to randomly divide the time interval into subintervals and update the sample path in each of the subintervals conditional on the rest of the sample path. This corresponds to simulating a (conditional) diffusion bridge in each subinterval (except the end-intervals).

The method outlined in this section is a modification of the method in [Chib, Pitt & Shephard (2006], where we use the algorithm for approximate diffusion bridge simulation proposed by [Bladt & Sørensen (2009].

In the following the parameter value $\theta = (\psi, \tau^2)$ is fixed.

Algorithm 1

1. Generate an initial unrestricted stationary sample path, $\{X_t^{(0)} : t \in [0, t_n]\}$.

2. Set $l = 1$.

3. Generate a sample path $\{X_t^{(l)} : t \in [0, t_n]\}$ conditional on Y by updating the subsets of the sample path:

   (a) Randomly split the time interval from 0 to t_n in K blocks, and write these subsampling times as

   $$0 = \tau_0 \leq \tau_1 \leq \ldots \leq \tau_K = t_n,$$

   where each τ_i is one of the end-points of the integration intervals, t_j, $j = 0, \ldots, n$. Let $Y_{\{k\}}$ denote the collection of all observations Y_j for which $\tau_{k-1} < t_j \leq \tau_k$.

(b) Draw $X_0^{(l)}$ from the stationary distribution, ν_ψ, and simulate the conditional subpath

$$\{X_t^{(l)} : t \in [\tau_{k-1}, \tau_k]\} \,|\, Y_{\{k\}}, X_{\tau_{k-1}}^{(l)}, X_{\tau_k}^{(l-1)} \tag{3.27}$$

for $k = 1, \ldots, K-1$. Finally, simulate a sample path from

$$\{X_t^{(l)} : t \in [\tau_{K-1}, \tau_K]\} \,|\, Y_{\{K\}}, X_{\tau_{K-1}}^{(l)}.$$

4. $l = l + 1$.

5. Go to 3.

The idea (in the case of a diffusion bridge in the time interval $[0, 1]$) is to let one diffusion process move forward from time zero out of one given point, a, until it meets another diffusion process that independently moves backwards from time one out of another given point, b. Conditional on the event that the two diffusions intersect, the process constructed in this way is an approximation to a realization of a diffusion bridge between a and b.

The diffusions can be simulated by means of simple procedures, used the Milstein scheme. The method is therefore very easy to implement.

The resulting sample path is an approximation to a diffusion bridge in the sense that it has the distribution of a diffusion bridge from a to b conditional on the event that the bridge is hit by an independent diffusion with stochastic differential equation (3.20) and initial distribution with density $p_1(b, \cdot)$. Simulation studies in [Bladt & Sørensen (2009] indicate that the approximation is very good for bridges between points that are likely to appear on a sample path of the diffusion, which is the type of bridges that are relevant to this section.

Alternative methods that provide exact diffusion bridges have been proposed by [Beskos, Papaspiliopoulos & Roberts (2007] and [Beskos, Papaspiliopoulos & Roberts (2006]. When the drift and diffusion coefficients satisfy certain boundedness conditions, this algorithm is relatively simple, but under weaker condition it is more complex.

A simulation study in [Bladt & Sørensen (2009] indicates that for the method which we use here, the CPU-time is linear in the length of the interval where the diffusion bridge is defined, whereas for the method in [Beskos, Papaspiliopoulos & Roberts (2007], the CPU time increases exponentially with the interval length. This is an advantage of the method in [Bladt & Sørensen (2009] in the present context. MCMC algorithms for simulation of diffusion bridges were proposed by [Roberts & Stramer (2001], [Durham & Gallant (2002], and [Chib, Pitt & Shephard (2006].

To generate the random subintervals in step 3 (a) of Algorithm 1, we use the following algorithm, where the number of integration subintervals $[t_{j-1}, t_j]$ included in one of the random subintervals is a Poisson distributed random number plus 1. The draws in the algorithm are independent. First choose the expectation of the Poisson distribution, $\lambda \geq 1$.

Algorithm 2

1. Draw $k_1 \sim Poisson(\lambda) + 1$: if $k_1 \geq n$ set $k_1 = n$, $K = 1$ and stop, otherwise set $i = 2$.
2. Draw $k_i \sim Poisson(\lambda) + 1$, if $\sum_{j=1}^{i} k_j \geq n$ set $k_i = n$, $K = i$ and stop, else set $i = i + 1$ and repeat 2.

Finally define $\tau_i = t_{k_i}$, $i = 1, \ldots, K$.

We have discussed how to simulate diffusion bridges, but we need diffusion bridges conditional on the data Y. Sample paths of the conditional bridges (3.27) can be obtained by the following Metropolis-Hastings algorithm. By a (t, a, s, b)–bridge, we mean a diffusion bridge in the time interval $[t, s]$ with $X_t = a$ and $X_s = b$. After a burn-in period the following algorithm will output samples from a $(\tau_{k-1}, a, \tau_k, b)$–bridge conditional on $Y_{\{k\}}$, the data in $(\tau_{k-1}, \tau_k]$.

To formulate the algorithm we need to specify that the end-point τ_{k-1} is equal to t_j, and that there are n_k observations in the interval $(\tau_{k-1}, \tau_k]$, namely, $Y_{j+1}, \ldots, Y_{j+n_k}$.

Algorithm 3

1. Simulate a $(\tau_{k-1}, a, \tau_k, b)$–bridge, $X^{(0)}$, and set $l = 1$.
2. Propose a new sample paths by simulating a $(\tau_{k-1}, a, \tau_k, b)$–bridge, $X^{(l)}$.
3. Accept the proposed diffusion bridge with probability

$$\min \left(1, \prod_{i=1}^{n_k} \frac{\phi(Y_{j+i}; \int_{t_{j+i-1}}^{t_{j+i}} X_s^{(l)} ds, \tau^2)}{\phi(Y_{j+i}; \int_{t_{j+i-1}}^{t_{j+i}} X_s^{(l-1)} ds, \tau^2)} \right).$$

   Otherwise set $X^{(l)} = X^{(l-1)}$.

4. Set $l = l + 1$ and go to 2.

As previously, $\phi(x; \mu, \tau^2)$ denotes the density function of the Normal distribution with mean μ and variance τ^2.

3.2.5 The Ornstein-Uhlenbeck process.

In this section we apply the method developed in the previous section above to the Ornstein-Uhlenbeck process, which is a solution of the stochastic differential equation

$$dX_t = -\alpha X_t dt + \sigma dW_t, \tag{3.28}$$

where $\alpha > 0$ and $\sigma > 0$ are unknown parameters to be estimated, and W is a standard Wiener process. We investigate the bias of the estimators in a simulation study.

The likelihood and the EM-algorithm

The transformation (3.23) is here given by

$$h(x;\sigma) = \frac{x}{\sigma},$$

so $h^{-1}(x;\sigma) = \sigma x$. Hence $U_t = h(X_t;\sigma) = X_t/\sigma$, solves the stochastic differential equation

$$dU_t = -\alpha U_t dt + dW_t.$$

We have $\mu(u;\alpha,\sigma) = -\alpha u$, so

$$a(u;\alpha,\sigma) = -\frac{1}{2}\alpha u^2.$$

Thus the full log-likelihood function (3.27) is given by

$$\begin{aligned}&\log L(\theta; Y_1,\dots,Y_n, U_t, t \in [0,t_n]) \qquad (3.29)\\ &= \sum_{i=1}^{n} \log\phi\left(Y_i; \sigma\int_{t_{i-1}}^{t_i} U_s ds, \tau^2\right) + \frac{\alpha}{2}(U_0^2 - U_{t_n}^2 + t_n) - \frac{\alpha^2}{2}\int_0^{t_n} U_t^2 dt,\end{aligned}$$

where $\theta = (\alpha,\sigma,\tau^2)$.

Now, the EM algorithm works as follow.

E-STEP

The objective function $g(\theta)$ is for the Ornstein-Uhlenbeck process given by

$$\begin{aligned} g(\theta) &= \frac{1}{M-M_0}\sum_{k=M_0+1}^{M}\sum_{i=1}^{n} -\frac{(Y_i - \sigma\int_{t_{i-1}}^{t_i} U_t^{(k)} dt)^2}{2\tau^2} - \frac{n}{2}\log(2\pi\tau^2) + \frac{\alpha}{2}t_n \\ &+ \frac{\alpha}{2(M-M_0)}\sum_{k=M_0+1}^{M}((U_0^{(k)})^2 - (U_{t_n}^{(k)})^2) - \frac{\alpha^2}{2(M-M_0)}\sum_{k=M_0+1}^{M}\sum_{i=1}^{n}\int_{t_{i-1}}^{t_i}(U_t^{(k)})^2 dt.\end{aligned}$$

Here $U_t^{(k)} = X_t^{(k)}/\hat{\sigma}$, where $X_t^{(k)}$ is the m-th sample path of the process X simulated conditionally on the data Y using the Algorithms 1 – 3 with the parameter value obtained in the previous step $(\hat{\alpha}, \hat{\sigma}, \hat{\tau}^2)$.

M-STEP

The maximum $\hat{\theta}$ is obtained as the solution to the following system of equations

$$\frac{\partial g(\theta)}{\partial \alpha} = \frac{1}{2}t_n + \frac{\sum_{k=M_0+1}^{M}\left[(U_0^{(k)})^2 - (U_{t_n}^{(k)})^2\right]}{2(M-M_0)} - \frac{\alpha\sum_{k=M_0+1}^{M}\sum_{i=1}^{n}\int_{t_{i-1}}^{t_i}(U_t^{(k)})^2 dt}{M-M_0} = 0, \qquad (3.30)$$

$$\frac{\partial g(\theta)}{\partial \sigma} = \frac{\sum_{k=M_0+1}^{M}\sum_{i=1}^{n}(Y_i - \sigma\int_{t_{i-1}}^{t_i} U_t^{(k)} dt)(\int_{t_{i-1}}^{t_i} U_t^{(k)} dt)}{\tau^2(M-M_0)} = 0 \qquad (3.31)$$

and

$$\frac{\partial g(\theta)}{\partial \tau^2} = \frac{\sum_{k=M_0+1}^{M}\sum_{i=1}^{n}(Y_i - \sigma\int_{t_{i-1}}^{t_i} U_t^{(k)} dt)^2}{2\tau^4(M-M_0)} - \frac{n}{2\tau^2} = 0. \qquad (3.32)$$

From (3.30) we have

$$\hat{\alpha} = \frac{t_n(M - M_0) + \sum_{k=M_0+1}^{M} \left[(U_0^{(k)})^2 - (U_{t_n}^{(k)})^2\right]}{2\sum_{k=M_0+1}^{M} \sum_{i=1}^{n} \int_{t_{i-1}}^{t_i} (U_t^{(k)})^2 dt},$$

and from (3.31)

$$\hat{\sigma} = \frac{\sum_{k=M_0+1}^{M} \sum_{i=1}^{n} Y_i \int_{t_{i-1}}^{t_i} U_t^{(k)} dt}{\sum_{k=M_0+1}^{M} \sum_{i=1}^{n} (\int_{t_{i-1}}^{t_i} U_t^{(k)} dt)^2}. \qquad (3.33)$$

Now inserting $\hat{\sigma}$ given by (3.33) in (3.32) we obtain

$$\hat{\tau}^2 = \frac{(M - M_0)(\sum_{i=1}^{n} Y_i^2)\left[\sum_{k=M_0+1}^{M} \sum_{i=1}^{n} (\int_{t_{i-1}}^{t_i} U_t^{(k)} dt)^2\right] - \left[\sum_{k=M_0+1}^{M} \sum_{i=1}^{n} Y_i \int_{t_{i-1}}^{t_i} U_t^{(k)} dt\right]^2}{n(M - M_0)\sum_{k=M_0+1}^{M} \sum_{i=1}^{n} (\int_{t_{i-1}}^{t_i} U_t^{(k)} dt)^2}.$$

The Hessian matrix of $g(\theta)$ evaluated at $\hat{\theta}$ is negative define, so $\hat{\theta}$ is maximum.

A simulation study

In this section we present the result of a small simulation study, in which we simulated 1000 datasets and for each of them obtained estimates by means of the EM-algorithm proposed in the present paper. Each data set was obtained by simulating a sample path of length 1500 with initial distribution $X_0 \sim N(0, \sigma^2/(2\alpha))$, and then calculating data Y_i, $i = 1, \ldots, 1500$ by (3.21) with $t_i = i$, $i = 0 \ldots, n$. The parameter values were $\alpha = 0.1$, $\sigma = 0.5$ and $\tau^2 = 1.25$.

The EM-algorithm was run with $M = 10000$ and $M_0 = 1000$ and for three different values of λ, namely $\lambda = 10, 20, 30$. The average of the estimates obtained for the 1000 dataset are given in Table 3.1. The bias is small, and is overall most satisfactory for $\lambda = 20$. We have presented an EM-algorithm for obtaining maximum likelihood estimates of parameters in diffusion models when the data is a discrete time sample of the integral of the diffusion process contaminated by measurement errors, while no direct observations of the process itself are available. This was done by viewing the data as an incomplete observation, where the full data set includes a continuous time record of the diffusion process.

λ	α	σ	τ^2
10	0.106	0.523	1.229
20	0.101	0.507	1.235
30	0.084	0.458	1.252

Table 3.1: Average of parameter estimates obtained from 1000 simulated datasets with parameter values $\alpha = 0.1$, $\sigma = 0.5$ and $\tau^2 = 1.25$.

Chapter 4

Applications of Maximun likelihood estimation for Integrated diffusion processes

In this chapter we apply the method developed in the last section of the previous chapter to the Ornstein-Uhlenbeck process where the parameters of an Ornstein-Uhlenbeck model are estimated from a set of integrated paleo-temperature data obtained from an ice-core from Greenland and we present a similar investigation for the CIR/square root process is presented.

4.1 Analysis of ICE-CORE data by Integrated diffusion processes

4.1.1 Introduction

Glaciers and ice caps exert a strong influence on the Earth's physical environment. During the last glacial period and earlier glacial stages as well, the expanded ice sheets in the Northern Hemisphere experienced periodic surges, which released extensive icebergs into the North Atlantic, which, in turn, may have initiated abrupt oscillations in climate the so-called Dansgaard-Oeschger cycles and Heinrich Events see [Broecker (1994]. Perhaps the most versatile attribute of massive glacier ice, however, is its tendency to preserve a detailed record of past environmental changes, frequently at a high temporal resolution, through the successive accumulation of annual increments of snow, which, under overburden pressure, become converted to solid ice. Each annual increment of snow deposited on a cold ice-cap surface entraps atmospheric gases and associated impurities as it becomes converted to solid ice, preserving a sample of the physical constituents of the former air column over the ice surface. As they wax and wane they can instigate significant changes in eustatic sea-level, earth surface albedo and the shape of the land over which they are generated.

Since the 1960s several long ice cores have been extracted from the polar ice caps, including those from Camp Century, Dye-3, Crete, Milcent and Summit in Greenland, and from Byrd, Vostok and Dome in Antarctica. Numerous quantities have been measured in ice cores to make inferences about earth history, such as oxygen isotope ratios, carbon dioxide concentration, methane concentration, electrical conductivity, and dust particle concentration. As the oxygen isotope variations spanning the last glacial cycle and the Holocene derived from ice-core records for Greenland and Antarctica show strong similarities, this suggests that the dominant influence on oxygen isotope variations reflected in the ice-sheet records was regional climatic change. Differences in detail between the records probably reflect the effects of basal deformation in the ice as well as geographical gradients in atmospheric isotope ratios.

Probably the most important measurement used in the study of climate has been the ratio of oxygen 18 to oxygen 16 ($^{18}O/^{16}O$). The isotope ratio $^{18}O/^{16}O$ in the ice, measured as an average in each piece of ice, is a proxy for paleo-temperatures averaged over the time interval in which the snow that formed the ice fell. The difference, $\delta^{18}O$, between the ^{18}O content in the ice and the ^{16}O content in present day's ocean water has been shown empirically to to have an approximately constant linear relationship with the air temperature where the precipitation falls. The variation of the paleo-temperature can be modelled by a stochastic differential equation, and it is natural to model the ice-core data as an integrated diffusion process. In this chapter we use the method proposed in the previous chapter to model ice-core data.

4.1.2 Ice cores

An ice core is a core sample from the accumulation of snow and ice over many years that have recrystallized and have trapped air bubbles from previous time periods. The composition of these ice cores, especially the presence of hydrogen and oxygen isotopes, provides a picture of the climate at the time.

Because water molecules containing heavier isotopes exhibit a lower vapor pressure, when the temperature falls, the heavier water molecules will condense faster than the normal water molecules. The relative concentrations of the heavier isotopes in the condensate indicate the temperature of condensation at the time, allowing for ice cores to be used in local temperature reconstruction after certain assumptions.

Typical ice cores are removed from an ice sheet, most commonly from the polar ice caps of Antarctica and Greenland. As the ice forms from the incremental buildup of annual layers of snow, lower layers are older than upper, and an ice core contains ice formed over a range of years. The properties of the ice or inclusions within the ice can then be used to reconstruct a climatic record over the age range of the core.

The length of the record depends on the depth of the ice core and varies from a few years up to 800 kyr [1] for the EPICA [2] core. An ice core from the right

[1] for the unit of one thousand years

[2] European Project for Ice Coring in Antarctica

site can be used to reconstruct an uninterrupted and detailed climate record extending over hundreds of thousands of years, providing information on a wide variety of aspects of climate at each point in time. It is the simultaneity of these properties recorded in the ice that makes ice cores such a powerful tool in paleoclimate research.

Many materials can appear in an ice core. Layers can be measured in several ways to identify changes in composition. Small meteorites may be embedded in the ice. Volcanic eruptions leave identifiable ash layers. Dust in the core can be linked to increased desert area or wind speed.

Figure 4.1: Ice-core

Isotopic analysis of the ice in the core can be linked to temperature and global sea level variations. Analysis of the air contained in bubbles in the ice can reveal the palaeocomposition of the atmosphere, in particular CO_2 variations.

4.1.3 Connection with the temperature.

In the 1950s, Willi Dansgaard investigated the relationship between the mean annual temperature and the $\delta^{18}O$ of precipitation at a large number of locations worldwide. He found a close correspondence in modern samples of precipitation, and it is believed that this relationship holds in the past, at least qualitatively. The main reason for this is the basic fact that the maximum amount of moisture that air can hold drops with decreasing temperatures.

The ratio ($^{18}O/^{16}O$) provides a record of ancient water temperature. Water 10 to 15 degrees Celsius (18 to 27 degrees Fahrenheit) cooler than present rep-

Figure 4.2: The ice-core record

resents glaciation. Precipitation and therefore glacial ice contain water with a low ${}^{18}O$ content. Since large amounts of ${}^{16}O$ water are being stored as glacial ice, the ${}^{18}O$ content of oceanic water is high. Water up to 5 degrees Celsius warmer than today represents an interglacial, when the ${}^{18}O$ content is lower.

The difference, $\delta^{18}O$, between the ${}^{18}O$ content in the ice and the ${}^{16}O$ content in present day's ocean water is a measure of the ratio of stable isotopes $({}^{18}O :{}^{16} O)$. It is commonly used as a measure of the temperature of precipitation, as a measure of ground water/mineral interactions, as an indicator of processes that show isotopic fractionation, like methanogenesis. In paleosciences, $({}^{18}O :{}^{16} O)$ data from foraminifera and ice core are used as a proxy for temperature. The Definition is, in per mil:

$$\delta^{18}O = \left(\frac{\left(\frac{{}^{18}O}{{}^{16}O}\right)_{sample}}{\left(\frac{{}^{18}O}{{}^{16}O}\right)_{standard}} - 1 \right) * 1000‰$$

where the standard has a known isotopic composition, such as Vienna Standard Mean Ocean Water (VSMOW).

The ratio of ${}^{18}O$ to ${}^{16}O$ is used to tell the temperature of the surrounding water of the time solidified, indirectly. The ratio varies slightly depending on the temperature of the surrounding water, as well as other factors such as the water's salinity, and the volume of water locked up in ice sheets. $\delta^{18}O$ also reflects local evaporation and freshwater input, as rainwater is ${}^{16}O$ enriched as result of ${}^{16}0$ preferential evaporation from seawater. Consequently, the surface ocean contains greater amounts of ${}^{18}O$ around the subtropics and tropics where there is more evaporation, and lesser amounts of ${}^{18}O$ in the mid-latitudes where it rains more. Similarly, when water vapor condenses, heavier water molecules holding ^{18}O atoms tend to condense and precipitate first. The water vapor gradient heading from the tropics to the poles gradually becomes more and more depleted of ^{18}O.

The conversion of $\delta^{18}O$ values to palaeotemperature estimates.

Palaeotemperature estimates have been obtained from the records using: inferences based on the measured relationship between mean annual $\delta^{18}O$ of snow and of mean annual surface temperature over Greenland, modelled inversion of the borehole temperature profile constrained either by the dated isotopic profile and by using Monte Carlo simulation techniques.

A direct link between stable isotope ratios in precipitating snow and temperature of water vapour at the time of precipitation is thus implied, assuming unchanged temperature and humidity at the moisture source areas and uniform deposition of snow throughout the year. Over the Greenland ice-sheet at the present time, mean annual $\delta^{18}O$ of snow is closely related to mean annual surface temperature, T($^\circ C$), by the equation:

$$\delta^{18}O = 0.67 \times T - 13.7‰ \tag{4.1}$$

This relationship, however, may not hold for the temporal changes in isotope values measured along the Greenland ice-cores. The temporal relationship between the $\delta^{18}O$ and contemporaneous surface temperature (T) can, however, be inferred by modelling the borehole temperature profile, using parameters derived from instrumental measurements of modern temperature history and the well-dated near-surface $\delta^{18}O$ profile. The assumed T and $\delta^{18}O$ relationship is as follows

$$T = \alpha + \beta\delta^{18}O + \gamma\delta^{18}O^2 \tag{4.2}$$

with α, β, γ, and being determined, along with the geothermal heat flux, by a least-squares fitting of the modelled temperature profile to the measured temperature profile.

4.1.4 Ice-core data

Here, we consider as an Example a preliminary analysis of ice-cores data. Ice-cores from Greenland and Antarctica are cut in pieces each of which represents a time interval in the past, with time increasing as a function of the depth. The deep ice cores retrieved in Antarctica and Greenland are becoming increasingly important for the understanding of past climate.

The ice cores obtained in Antarctica have provided paleoclimatic records that cover more than 800 kyr of climate history whereas the Green land ice cores roughly cover the last glacial cycle (North Greenland Ice Core Project members, 2004). In order to interpret the climatic signal provided by the ice cores and to enable comparison with other paleo-climatic records accurate time scales are crucial. Because of their high accumulation rates, the Greenland ice cores are well suited for obtaining a chronology based on annual layer counting of the last glacial cycle. In addition, the Greenland ice cores very strongly reflect the abrupt climatic shifts of the last glacial period, the Dansgaard-Oeschger events, and they contain many reference horizons that enable comparison to other paleo-archives.

The most widely applied Greenland ice core time scales are the Meese-Sowers GISP2 (Greenland Icecore Project 2) stratigraphic time scale and the modeled ss09sea time scale that has been applied to the GRIP(Greenland Icecore Project) and NorthGRIP ice cores. Those time scales agree within 750 years back to 40 kyr, but beyond this point the disagreement becomes several thousands of years. So far, there has thus been no consensus for the Greenland ice core time scales in the glacial period. We consider a data set obtained from an ice core obtained from the Greenland ice-sheet, see [A. Svensson et al. (2008], where, was used the extension of the GICC05 (Greenland Ice Core Chronology 2005)[3] back to 60 kyr.

The record of integrated $\delta^{18}O$ goes 60,000 years back in time, and each data point is the average over a time interval of length 20 years. Thus there are 3000 data points. These data correspond to the Quaternary Period, that is the most recent of the three periods of the Cenozoic Era in the geologic time scale of the ICS[4]. It follows the Neogene Period, spanning 2.588 million years ago to the present. The Quaternary includes two geologic epochs:

1. The Pleistocene, that is epoch of glacial cycles, evolution of modern humans and megafauna extinction. The Pleistocene is divided into four stages or ages, the Gelasian, Calabrian, Ionian and Tarantian.

2. The Holocene,that is the epoch of the end of Ice Age and the emergence of modern civilization.

The Table 4.1 shows the subdivisions of the Quaternary period. Then, the data correspond only to the Holocen epoch and the last two ages of the Pleistocene epoch. The variation of the paleo-temperature can be modelled by a stochastic differential equation, see [Ditlevsen, Ditlevsen & Andersen (2002], where consider as a first approximation, the generic way of describing the variations on climatic time-scales is that each of the states can be described as resulting from a simple linear auto-regressive process

Epoch	Ages	Older (myr)
Holocene		0-0.0117
Pleistocene	Tarantian	0.0117-0.126
Pleistocene	Ionian	0.126-0.781
Pleistocene	Calabrian	0.781-1.806
Pleistocene	Gelasian	1.806-2.588

Table 4.1: Subdivisions of the Quaternary Period.

$$x_{n+1} - \bar{x} = (1 - \sigma\Delta t)(x_n - \bar{x}) + \sigma\nu_{n+1}. \qquad (4.3)$$

[3] GICC05 is a time scale based on annual layer counting of high-resolution records from Greenland ice cores.

[4] International Commission on Stratigraphy

This is the discrete version of linear Langevin equation. The $\delta^{18}O$ climate variable is denoted x_n, and the discrete time-steps are $t_n = n\Delta t$, where Δt is the resolution of the record. The first term on the righthand side represents the 'stiffness' of the stable climate state in restoring the equilibrium state $\bar{x}$ when the system is perturbed by noise. The second term on the right hand side is climatic noise which on these timescales act as stochastic forcing. The strength of the atmospheric forcing on the climate state is represented by stationary white noise $\sigma\nu(t)$ of intensity σ^2 discretized in equation (4.3) to stationary sequence $\{\sigma\nu_n\}$ where ν_n has variance Δt.

In [Ditlevsen, Ditlevsen & Andersen (2002] assumed that the record is generated by the process 4.3 and $\Delta t \to 0$, then the continuous version is the Ornstein-Uhlenbeck process.

Statistical inference for observations of integrated diffusions with application on ice-core records has been studied in [Ditlevsen & Sørensen (2004] where, prediction-based estimating functions are applied to estimate parameters in the underlying diffusion model.

The time series is plotted in Figure 4.3. It is obviously not stationary, so in a preliminary analysis we divide it into three subintervals, [-60000, -30000], [-30000, -10000], and [-10000, 0].

Since that the variation of the paleo-temperature can be modelled by a stochastic differential equation, it is natural to model the ice-core data as an integrated diffusion process, then we here consider the ice-core data as an observation of type

$$X_i = \frac{1}{\Delta}\int_{t_{i-1}}^{t_i} X_t d_t$$

and each of these intervals we model the observations by integrated diffusion observations where X is an Ornstein-Uhlenbeck process, and the integrals are over intervals of length 20. We apply the method developed in the previous chapter for the analysis of this time serie. The EM algorithm is run with $M = 10000, M_0 = 1000$, and $\lambda = 20$ for each of the three blocks of data. The resulting estimates are reported in Table 4.2. The parameter α^1 can be interpreted as a correlation time, and $\sigma/\sqrt{2\alpha}$ is the standard deviation of the stationary distribution. When compared to the plot, these parameter values seem reasonable. Now, we use the equation 4.2 and obtain a least squares ad-

Parameter	$[-10000, 0]$	$[-30000, 10000]$	$[-60000, -30000]$
α^-1	205.3	669.8	321.1
σ	0.0303	0.1167	0.1395
$\sigma/\sqrt{2\alpha}$	0.307	2.136	1.767
τ^2	0.1062638	0.6966518	0.225974

Table 4.2: Estimates obtained by the EM-algorithm for an Ornstein-Uhlenbeck model for three subintervals.

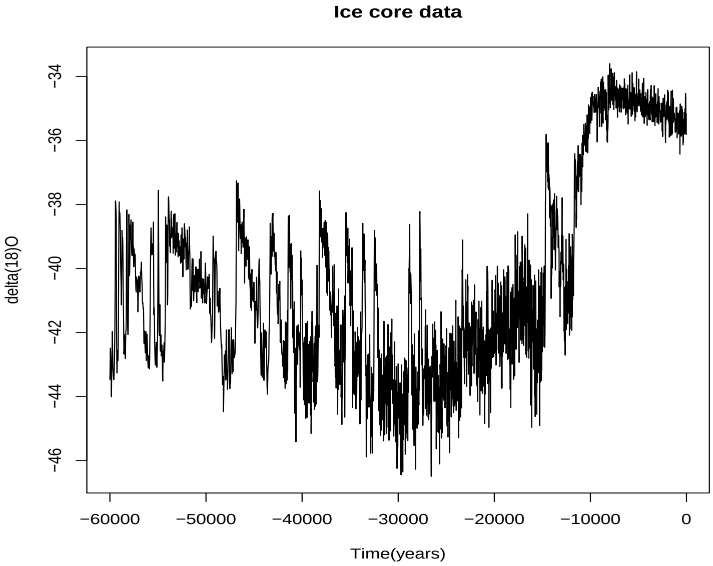


Figure 4.3: $\delta^{18}O‰$-values integrated over 20 years intervals obtained from ice core data from the Greenland ice-sheet.

justment for temperature, here, we consider only the last interval of observation, i. e., [-10000, 0], which is the actual epoch, then the corresponding equation is

$$T = 20.45 + 1493\delta^{18}O + 1.051 \times (10^{-10})\delta^{18}O \tag{4.4}$$

Now, using the results of the estimation of the integrated process of diffusion can make an estimate of annual observations using Milstein scheme and parameter estimates, then we can obtain the annual average temperatures for the correspondig period using equation 4.4, we present this temperatures in Figure 4.4.

4.2 Analysis of volatility by Integrated diffusion processes.

In this section we present a overview of stochastic volatility models, we apply the method of maximum likelihood estimation developed in the previous chapter for the CIR process a simulation study for this model is reported.

4.2.1 Introduction

Stochastic volatility models are used heavily within the fields of financial economics, probability theory and econometrics blending to produce methods that aid our understanding of option pricing, efficient portfolio allocation and accurate risk assessment and management.

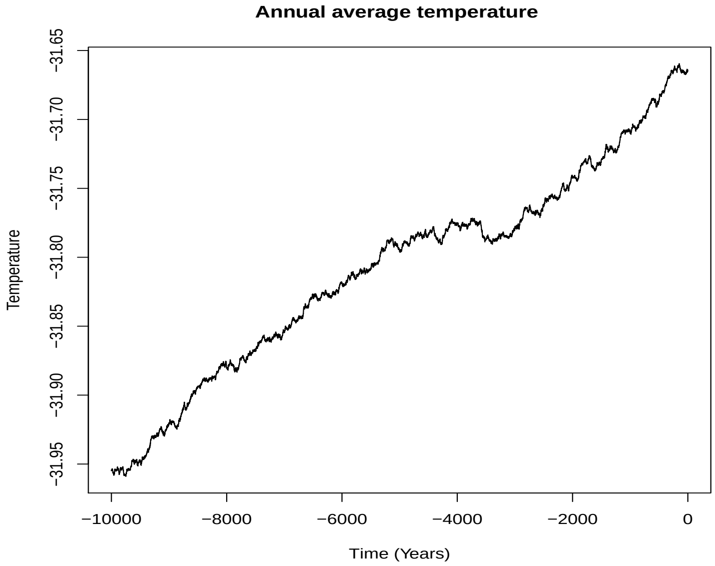


Figure 4.4: Annual Average temperature of the last 10,000 years in degrees celsius

Stochastic volatility models are one approach to resolve a shortcoming of the Black-Scholes model. In particular, these models assume that the underlying volatility is constant over the life of the derivative, and unaffected by the changes in the price level of the underlying security. However, these models cannot explain long-observed features of the implied volatility surface such as volatility smile and skew, which indicate that implied volatility does tend to vary with respect to strike price and expiry. By assuming that the volatility of the underlying price is a stochastic process rather than a constant, it becomes possible to model derivatives more accurately.

Since the mid-1980s continuous-time stochastics volatility has dominated the option pricing literature but early on econometricians struggled with the difficulties of estimating and testing these models. Only in the 1990s were novel simulation strategies developed to efficiently estimate stochastic volatility models. These computationally intensive methods enable us, given enough coding and computing time, to efficiently estimate a broad range of fully parametric stochastics volatility models. This has lead to refinements of the models, with many earlier tractable models being rejected from an empirical viewpoint. The resulting enriched stochastics volatility literature has brought us much closer to the empirical realities we face in financial markets.

From the late 1990s stochastics volatility models have taken center stage in the econometric analysis of volatility forecasting using high-frequency data based on realized volatility and related concepts. The reason is that the econometric analysis of realized volatility is tied to continuous-time processes, so stochastics volatility is central. The close connection between stochastics volatility and realized volatility has allowed financial econometricians to harness the enriched information set available through highfrequency data to improve, by an order of magnitude, the accuracy of their volatility forecasts over that traditionally offered by ARCH models based on daily observations. This has broadened the applications of stochastics volatility into the important arena of risk assessment and asset allocation.

4.2.2 Stochastic volatility models.

Here, we describe the most popular stochastic volatility models proposed in the literature to represent different aspects observed in volatility.

We suppose that the stock price S and its variance ν satisfy the following stochastic differential equations:

$$dS_t = \mu S_t \, dt + \sqrt{\nu_t} S_t \, dW_t \tag{4.5}$$

$$d\nu_t = \alpha_{S,t} \, dt + v\beta_{S,t}\sqrt{\nu_t} \, dB_t \tag{4.6}$$

with

$$\langle W_t, B_t \rangle = \rho$$

where μ is the instantaneous drift of stock price returns,v is the volatility of volatility is the volatility of volatility and ρ is the correlation between random stock price returns and changes in ν_t. W and B are Wiener processes.

The stochastic process given by equation (4.5) followed by the stock price is equivalent to the one assumed in the derivation of [Black & Scholes (1973]. This ensures that the standard time-dependent volatility version of the Black-Scholes formula may be retrieved in the limit $v \to 0$. In practical applications, this is a key requirement of a stochastic volatility option pricing model as practitioners intuition for the behavior of option prices is invariably expressed within the framework of the Black-Scholes formula.

The Heston model.

The Heston model corresponds to choosing

$$\alpha_{S,t} = -\lambda(\nu_t - \bar{\nu}$$

and

$$\beta_{S,t} = 1$$

in equations (4.5) and (4.6), then we have

$$dS_t = \mu S_t \, dt + \sqrt{\nu_t} S_t \, dW_t \tag{4.7}$$

$$d\nu_t = \lambda(\nu_t - \bar{\nu}\alpha_{S,t} \, dt + v\sqrt{\nu_t} \, dB_t \tag{4.8}$$

where λ is the speed of reversion of ν_t to its long term mean $\bar{\nu}$.

The process followed by ν_t may be recognized as a version of the square root process described by Cox, Ingersoll, and Ross (CIR). It is a special case of a so-called affine jump diffusion (AJD) which is roughly speaking a jump-diffusion process for which the drifts and covariances and jump intensities are linear in the state vector. AJD processes are analytically tractable in general. The solution technique involves computing an extended transform which in the Heston case is a conventional Fourier transform.

Chen model

A model for interest rate dynamics and is thus different than those considered previously. The dynamics of the instantaneous interest rate $\{r_t\}_{t\geq 0}$, the mean level of the interest rate $\{\alpha\}_{t\geq 0}$, and the volatility $\{\sigma_t\}_{t\geq 0}$ are specified by the following stochastic differential equations:

$$dr_t = (\theta_t - \alpha_t)\,dt + \sqrt{r_t}\,\sigma_t\,dW_t,$$

$$d\alpha_t = (\zeta_t - \alpha_t)\,dt + \sqrt{\alpha_t}\,\sigma_t\,dW_t,$$

$$d\sigma_t = (\beta_t - \sigma_t)\,dt + \sqrt{\sigma_t}\,\eta_t\,dW_t.$$

In finance, the Chen model is a mathematical model describing the evolution of interest rates. It is a type of ”three-factor model” (short rate model) as it describes interest rate movements as driven by three sources of market risk. It was the first stochastic mean and stochastic volatility model and it was published in 1994 by the economist Lin Chen.

GARCH model

The Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model is another popular model for estimating stochastic volatility. It assumes that the randomness of the variance process varies with the variance, as opposed to the square root of the variance as in the Heston model. The standard GARCH(1,1) model has the following form for the variance differential:

$$d\nu_t = \theta(\omega - \nu_t)dt + \xi\nu_t\,dB_t.$$

4.2.3 Maximum likelihood estimation for the CIR process and a stochastic volatility model

In this section we apply the method of the previous chapter to the CIR process, which solve

$$dX_t = (\alpha - \beta X_t)dt + \sigma\sqrt{X_t}dW_t, \tag{4.9}$$

The CIR process plays important roles in financial mathematics, where it has been used to describe the model the volatility, where the dynamics of the logarithm of the price, P_t , of a financial asset is given by

$$dP_t = (\kappa + \nu X_t)dt + \sqrt{X_t}dB_t,$$

where the volatility process X is given by equation (4.9), and where the standard Brownian motion B may possibly be correlated with the Brownian motion W in equation (4.9). If high frequency observations of the asset price are available at the time points $j\delta$,$j = 0, \dots, N$, then the integrated volatility over longer time intervals of length $\Delta = m\delta$ (e.g. hours or days)

$$\int_{(i-1)\Delta}^{i\Delta} X_t dt,$$

$i = 1, \dots, n = [N/m]$, can be estimated by the quadratic variation/realized volatility

$$V_i = \sum_{j=(i-1)m+1}^{im} (P_{j\delta} - P_{(j-1)\delta})^2$$

$i = 1, \dots, n$ We can therefore estimate the parameters α, β, σ in the volatility process (4.9) by treating the realized volatilities $V_i, i = 1, \dots, \dots, n$ as observations Y_i of the type (3.21) with X given by (4.9) and $t_i = i\Delta$.

A simulation study.

We simulated 1500 datasets, and for each of them obtained estimates by means of our EM-algorithm. Each data set was obtained by simulating a sample path of length 1500 with initial distribution $X_0 \sim \Gamma(2\alpha/\sigma^2, \sigma^2/\beta)$, then calculating data Y_i, $i = 1, \dots, 1500$ by (3.21). The EM-algorithm was run with $M = 10000$ and $M_0 = 1000$ and for three different values of λ, namely $\lambda = 10, 20, 30$.

4.2.4 The likelihood and the EM-algorithm

We apply the Lamperti transform

$$h(x;\sigma) = \frac{2\sqrt{x}}{\sigma},$$

so $h^{-1}(x;\sigma) = \frac{\sigma^2 x^2}{4}$. Hence $U_t = h(X_t;\sigma) = \frac{2\sqrt{X_t}}{\sigma}$, solves the stochastic differential equation

$$dU_t = \mu(U_t; \alpha, \beta, \sigma)dt + dW_t.$$

We have

$$\mu(u; \alpha, \beta, \sigma) = \frac{4\alpha - \beta\sigma^2 u^2 - \sigma^2}{2\sigma^2 u},$$

so

$$a(u; \alpha, \beta, \sigma) = \log(u)\left(\frac{2\alpha}{\sigma^2} - \frac{1}{2}\right) - \frac{\beta u^2}{4}.$$

Here, we have that the full log-likelihood function (3.27) is given by

$$\log L(\theta; Y_1, \dots, Y_n, U_t, t \in [0, t_n]) \tag{4.10}$$

$$= \sum_{i=1}^{n} \log\phi\left(Y_i; \frac{\sigma^2}{4}\int_{t_{i-1}}^{t_i}(U_s)^2 ds, \tau^2\right) + \log\left(\frac{U_{t_n}}{U_{t_0}}\right)\left(\frac{2\alpha}{\sigma^2} - \frac{1}{2}\right) + \frac{\beta}{4}(U_{t_0}^2 - U_{t_n}^2) + \frac{\alpha\beta t_n}{\sigma^2},$$

$$-\frac{\beta^2}{8}\int_{t_0}^{t_n} U_t^2 dt - \frac{2\alpha^2}{\sigma^4}\int_{t_0}^{t_n}\frac{1}{U_t^2}dt - \frac{3}{8}\int_{t_0}^{t_n}\frac{1}{U_t^2}dt + \frac{2\alpha}{\sigma^2}\int_{t_0}^{t_n}\frac{1}{U_t^2}dt,$$

where $\theta = (\alpha, \beta, \sigma, \tau^2)$.

Now, we apply the EM algorithm.

E-STEP

The objective function $g(\theta)$ is for the CIR process given by

$$\begin{aligned}
g(\theta) &= \frac{1}{M-M_0}\sum_{k=M_0+1}^{M}\sum_{i=1}^{n}-\frac{(Y_i-\frac{\sigma^2}{4}\int_{t_{i-1}}^{t_i}(U_t^{(k)})^2dt)^2}{2\tau^2}-\frac{n}{2}\log(2\pi\tau^2)+\frac{\alpha\beta t_n}{\sigma^2}\\
&+ \left(\frac{2\alpha}{\sigma^2}-\frac{1}{2}\right)\frac{1}{M-M_0}\sum_{k=M_0+1}^{M}\log\left(\frac{U_{t_n}^{(k)}}{U_{t_0^{(k)}}}\right)+\frac{\beta}{4(M-M_0)}\sum_{k=M_0+1}^{M}((U_{t_0}^{(k)})^2-(U_{t_n}^{(k)})^2)\\
&+ \frac{2\alpha\sigma^2-2\alpha^2}{\sigma^4(M-M_0)}\sum_{k=M_0+1}^{M}\sum_{i=1}^{n}\int_{t_{i-1}}^{t_i}\frac{1}{(U_t^{(k)})^2}dt-\frac{3}{8(M-M_0)}\sum_{k=M_0+1}^{M}\sum_{i=1}^{n}\int_{t_{i-1}}^{t_i}\frac{1}{(U_t^{(k)})^2}dt\\
&- \frac{\beta^2}{8(M-M_0)}\sum_{k=M_0+1}^{M}\sum_{i=1}^{n}\int_{t_{i-1}}^{t_i}(U_t^{(k)})^2dt.
\end{aligned}$$

Here $U_t^{(k)}=2\sqrt{X_t^{(k)}}/\hat{\sigma}$, where $X_t^{(k)}$ is the k-th sample path of the process X simulated conditionally on the data Y using the Algorithms 1 – 3 with the parameter value obtained in the previous step $(\hat{\alpha},\hat{\beta},\hat{\sigma},\hat{\tau}^2)$.

M-STEP

The maximum $\hat{\theta}$ is obtained as the solution to the following system of equations

$$\frac{\partial g(\theta)}{\partial\alpha}=\frac{2}{\sigma^2(M-M_0)}\sum_{k=M_0+1}^{M}\log\left(\frac{U_{t_n}^{(k)}}{U_{t_0^{(k)}}}\right)+\frac{\beta t_n}{\sigma^2}+\frac{2\sigma^2-4\alpha}{\sigma^4(M-M_0)}\sum_{k=M_0+1}^{M}\sum_{i=1}^{n}\int_{t_{i-1}}^{t_i}\frac{1}{(U_t^{(k)})^2}dt=0, \tag{4.11}$$

$$\frac{\partial g(\theta)}{\partial\beta}=\frac{1}{4(M-M_0)}\sum_{k=M_0+1}^{M}((U_{t_0}^{(k)})^2-(U_{t_n}^{(k)})^2)+\frac{\alpha t_n}{\sigma^2}-\frac{\beta}{8(M-M_0)}\sum_{k=M_0+1}^{M}\sum_{i=1}^{n}\int_{t_{i-1}}^{t_i}(U_t^{(k)})^2dt=0, \tag{4.12}$$

$$\begin{aligned}
\frac{\partial g(\theta)}{\partial\sigma} &= \frac{1}{M-M_0}\sum_{k=M_0+1}^{M}\sum_{i=1}^{n}\frac{(Y_i-\frac{\sigma^2}{4}\int_{t_{i-1}}^{t_i}(U_t^{(k)})^2dt)\sigma\int_{t_{i-1}}^{t_i}(U_t^{(k)})^2dt}{2\tau^2}-\frac{2\alpha\beta t_n}{\sigma^3} \qquad (4.13)\\
&+ \frac{4\alpha}{\sigma^3(M-M_0)}\sum_{k=M_0+1}^{M}\log\left(\frac{U_{t_n}^{(k)}}{U_{t_0^{(k)}}}\right)+\frac{4\alpha(2\alpha-\sigma^2}{\sigma^5(M-M_0)}\sum_{k=M_0+1}^{M}\sum_{i=1}^{n}\int_{t_{i-1}}^{t_i}\frac{1}{(U_t^{(k)})^2}dt=0,
\end{aligned}$$

and

$$\frac{\partial g(\theta)}{\partial\tau^2}=\frac{1}{M-M_0}\sum_{k=M_0+1}^{M}\sum_{i=1}^{n}-\frac{(Y_i-\frac{\sigma^2}{4}\int_{t_{i-1}}^{t_i}(U_t^{(k)})^2dt)^2}{2\tau^4}-\frac{n}{2\tau^2}=0. \tag{4.14}$$

The solution of system equations is given by

$$\hat{\alpha}=\frac{C_5(2C_2C_4+2C_1C_4+C_3t_n)}{C_6(C_4C_2-t_n^2)}, \tag{4.15}$$

$$\hat{\beta} = \frac{C_3C_2 + 2C_2t_n + 2C_1t_n}{C_4C_2 - t_n^2}, \tag{4.16}$$

$$\hat{\sigma} = \sqrt{4C_5/C_6}, \tag{4.17}$$

$$\hat{\tau}^2 = \frac{C_7C_6 - C_5^2}{n(M - M_0)C_6}, \tag{4.18}$$

where the values of the constans C_i is given in the Table 4.3.

constant	values
C_1	$\frac{1}{M-M_0}\sum_{k=M_0+1}^{M}\log\left(\frac{U^{(k)}_{t_n}}{U_{t^{(k)}_0}}\right)$
C_2	$\sum_{k=M_0+1}^{M}\sum_{i=1}^{n}\int_{t_{i-1}}^{t_i}\frac{1}{(U^{(k)}_t)^2}dt$
C_3	$\sum_{k=M_0+1}^{M}((U^{(k)}_{t_0})^2-(U^{(k)}_{t_n})^2)$
C_4	$\sum_{k=M_0+1}^{M}\sum_{i=1}^{n}\int_{t_{i-1}}^{t_i}(U^{(k)}_t)^2dt$
C_5	$\sum_{k=M_0+1}^{M}\sum_{i=1}^{n}\int_{t_{i-1}}^{t_i}Y_i(U^{(k)}_t)^2dt$
C_6	$\sum_{k=M_0+1}^{M}\sum_{i=1}^{n}(\int_{t_{i-1}}^{t_i}(U^{(k)}_t)^2dt)^2$
C_7	$\sum_{k=M_0+1}^{M}\sum_{i=1}^{n}Y_i$

Table 4.3: Values of the constans C_i.

Now, we present data for our illustration is simulated from the above CIR-model, which we simulated 1500 datasets and for each of them obtained estimates by means of the EM-algorithm proposed in the present paper. Each data set was obtained by simulating a sample path of length 1500 with initial distribution $X_0 \sim gamma(2\alpha/\sigma^2, \sigma^2/\beta)$, and then calculating data Y_i, $i = 1, \ldots, 1500$ by (3.21) with $t_i = i$, $i = 0. \ldots, n$. The parameter values were $\alpha = 0.5$, $\beta = 0.2$, $\sigma = 0.5$ and $\tau^2 = 1.25$.

The EM-algorithm was run with $M = 10000$ and $M_0 = 1000$ .We specify a design to represent typical weekly and daily financial data sets. Using $\Delta = t_i - t_{i-1} = 1$ for daily series and $\Delta = t_i - t_{i-1} = 5$ for weekly data. Here âĂă can be thought of as the time interval in the observed data and determines the bias in the discretized time gap. The average of the estimates obtained for the 1000 dataset are given in Table 4.4. We have presented an EM-algorithm for obtaining

λ	α	β	σ	τ^2
30	0.4802	0.2056	0.4787	1.2432
20	0.4727	0.2043	0.4698	1.2406
10	0.4587	0.1965	0.4609	1.2287

Table 4.4: Average of parameter estimates obtained from 1000 simulated datasets with parameter values $\alpha = 0.1$, $\sigma = 0.5$ and $\tau^2 = 1.25$.

maximum likelihood estimates of parameters in diffusion models when the data is a discrete time sample of the integral of the diffusion process contaminated by measurement errors, while no direct observations of the process itself are available. This was done by viewing the data as an incomplete observation, where the full data set includes a continuous time record of the diffusion process.

Chapter 5

Concluding and Outlook

The initial idea of this work was the estimation of Markov processes. This work is focused on propose methods for obtaining estimates of parameters in Markov jump processes and diffusion models when the data is a discrete time sample of the integral of the process.

We have presented two important results, the first, an algorithm for the estimation of full parameters to observations from several Markov jump processes which conditional on an underlying Markov jump process, in this work we have studied the case when the Markov jump processes have been discretely observed and the case when the underlying process has been observed continuously in a time interval and the case when the underlying process has been observed only at discrete time points.

In this part of the work we discussed the problems that are related to maximum likelihood estimation of the intensity matrix based on a discretely sampled Markov jump processes generalizing the results of Bladt and Sørensen in [Bladt & Sørensen (2005] on estimation of transition rates by a MCMC approach to observations from several Markov jump processes which conditional on underlying Markov jump process are independent with the same transition rates and apply our results to analysis of credit rating transition. An real example with credit rating data conditional on interest rate was presented.

As a natural extension, we can assume that the underlying process is a diffusion process and we will study the rates of jumps in the Markov jump processes.

For another hand, we have presented an EM-algorithm for obtaining maximum likelihood estimates of parameters in diffusion models when the data is a discrete time sample of the integral of the diffusion process contaminated by measurement errors, while no direct observations of the process itself are available. This was done by viewing the data as an incomplete observation, where the full data set includes a continuous time record of the diffusion process.

It is not difficult to generalize the method presented in this work to the situation, where the diffusion process is integrated with respect to a more general measure than the Lebesgue measure considered in this work. This would al-low analysis of e.g. weighted averages of diffusion processes, and discrete time observation would be a particular case. Note also that a Gibbs sampler could easily be set up in close analogy to the EM-algorithm used in the present work. This would be much closer to the approach in Chib, Pitt and Shephard.

Finally, we present two applications of the method developed to integrated diffusion processes to the Ornstein-Uhlenbeck process where the parameters of an Ornstein-Uhlenbeck model are estimated from a set of integrated paleo-temperature data obtained from an ice-core from Greenland and a similar investigation for the Cox-Ingersoll-Ross(CIR)/square root process was presented.

Appendix A: Code in Fortran of Markov jump processes

Here, we present the code that was used to the estimation of parameters in the analysis of credit data of chapter 2.

```
! program Markov jump processes

implicit none

! PARAMETERS

INTEGER dstates,numobs,numiter,numpro,gib_iter ,burn_MH,burn_Gibbs,numcom
parameter (gib_iter=10000)
parameter (numpro=696)
parameter (numcom=2169)
parameter (dstates=3)
parameter (numobs=10)
parameter (numiter=1000)
parameter (burn_Gibbs=1000)

! VARIABLES

REAL Col1,Col2,Col3,Col4,Col5,DELTA,TIMESPAN,timer,x,long,suma,XGMJ
REAL timeobs(10000),HV0(10000),HV1(10000),HV(10000),HVO(10000)
REAL HTMJPOBS(10000),HUMHJP(10000),timemjp(10000)
REAL GIMJP(10,10*dstates),BETA(10),ALPHA(10,10*dstates)
REAL P(10,10*dstates),QMJ(10,10*dstates),GIMJP0(10,10*dstates)
REAL MVQhechi(10,10),STC(10,dstates),TSTC(10,dstates)
REAL RHT(10,dstates),RH(10,dstates),MCH(10,10),AVERAGE(10,10*dstates)
REAL TIM(10,10*dstates),timemjpto(10000,numpro),MVEQC(10,10*dstates)
REAL Q(dstates,dstates),Q0(dstates,dstates),MVQ(dstates,dstates)
REAL BETA0(dstates),ALPHA0(dstates,dstates),QIM(dstates,dstates)
REAl STMJP(dstates),averaQ(dstates,dstates),mev(numobs)
REAL timefirst(numobs),tinstate(numobs),HoldUMJP(dstates)
```

```
! INTEGER
INTEGER i,j,l,cont,na,stari,initp,njumpsobs,den,mod,nstates,dimen
INTEGER NJT1,NJT0,NJT,NJTO,NJUMPS,NJUMH,nobser,ndate,company,gib,FIRM
INTEGER MJC(10000),MJPOBS(10000),MCU0(10000),MCU1(10000),MCU(10000)
INTEGER MJPES(10000),UMHJP(10000),TNJC(10,10*dstates),Y(numobs)
INTEGER STATES(10000,numpro),MJPEST(10000,numpro),NJC(10,10*dstates)
INTEGER ustates(numobs),numdata(numpro),MJP(numobs,numpro)
INTEGER JUMPO(10,10),NJTU(10,10*dstates),orgdata(numcom,10)
INTEGER YT(numobs,numpro),WT(numobs,numpro),JUMP1(dstates,dstates)
INTEGER Z(numobs,numpro),STEMJP(numcom,11),dimenpro(numpro)
INTEGER NJCU(10,10*dstates),MJMJP(dstates,dstates)
INTEGER numfirm(numcom),numdat(numcom),datMJP(numobs)
INTEGER JUMP0(dstates,dstates),J0(dstates,dstates),W(numobs)
INTEGER MJUMJP(dstates,dstates),NJCU1(10,10*dstates)

! ALLOCATABLE
INTEGER, allocatable :: JP0(:),JP1(:),JPO(:),UJP(:)
REAL, allocatable :: times1(:),times(:),times0(:)
REAL, allocatable :: HMJO(:),HMJ0(:),HMJ1(:),HU(:)

! CHARACTER
CHARACTER*10 codigo

! External Functions
REAL gengam,genexp
external gengam,genexp

! Reading the data of Markov Jump Processes.
OPEN(UNIT=3,FILE='datamoodys.txt',STATUS='old',ACTION="READ")
            READ(3,*) TIMESPAN
            READ(3,*) nstates
            READ(3,*) DELTA
endfile(10)
CLOSE (UNIT=10)
FIRM=0
i=1
do while(i<=numcom)
          read(3,*) numdat(i)
          do j=1,numdat(i)
                    read(3,*) numfirm(i),ndate,STEMJP(i,j),codigo
          enddo
          firm=firm+1
          i=i+1
enddo

! Using only the companies with 10 observations
company=1
do i=1,numcom
          if(numdat(i)==10)then
                    do j=1,numobs
                    MJP(j,company)=STEMJP(i,j)
                enddo
                company=company+1
        else
                company=company
        endif
enddo

! Read the data of underlyin process
OPEN(UNIT=10,FILE='datmvcr',STATUS='old',ACTION="READ")
do i=1,numobs
        read(10,*) Col1,Col2,Col3,Col4,Col5
        mev(i)=Col2
enddo
endfile(10)
CLOSE (UNIT=10)
call Clastates(numobs,mev,datMJP)
l=nstates*dstates
nobser=numobs
long=(numobs-1)*1.0
timer=(numobs-1)*1.0
TNJC=0
TSTC=0.0

! INITALIZE PARAMETERS MJP
call Ialbe(nstates,dstates,alpha,beta)

! INITALIZE PARAMETERS Underlying
call Unalbe(dstates,alpha0,beta0)

! Initial point of UMJP
  call ipoint(dstates,initp)

! Calculate the time spent in each state
do i=1,numobs-1
        timefirst(i)=(i-1)*1.0
        tinstate(i)=1.0
enddo
timefirst(numobs)=(numobs-1)*1.0
tinstate(numobs)=0.0

! MCMC for Intesity Matrix
! Initialize  Intensity Matrix
  call priorIM(nstates,dstates,ALPHA,BETA,GIMJP0)
  call priorQ(dstates,ALPHA0,BETA0,Q0)
GIMJP=GIMJP0
Q=Q0

! Initial Underlying Markov jump process
  call MJPCU(numobs,dstates,datMJP,timefirst,Q0,HV0,MCU0,NJT0)
```

```
do i=1,NJT0+1
          HUMHJP(i)=HV0(i)
          UMHJP(i)=MCU0(i)
enddo
          call timet(NJT0,HV0,timeobs)
NJUMH=NJT0

! Initial Markov jump processes
i=1
do while(i<=numpro)
call MJPCDO(numobs,nstates,dstates,NJT0,UMHJP,timeobs,timefirst,&
               &,GIMJP0,MJP(:,i),MJPES,timemjp,dimen)
    MJPEST(:,i)=MJPES
    timemjpto(:,i)=timemjp
    dimenpro(i)=dimen
    call auxiliar(numobs,MJPES,UMHJP,dimen-1,NJT0,timemjp,timeobs,timefirst
    YT(:,i)=Y
    WT(:,i)=W
enddo

! Initial auxiliar variables
call augdata(numobs,numpro,dstates,nstates,YT,WT,MJP,Q0,GIMJP0,Z)

! Gibbs Sample
averaQ=0.0
AVERAGE=0.0
gib=1
den=gib_iter-burn_Gibbs
do while(gib<=gib_iter)
          na=0
          call timet(NJUMH,HUMHJP,timeobs)
          i=1
          do while(i<=numpro)
                    call MJPCDO(numobs,nstates,dstates,NJUMH,UMHJP,timeobs,&
                               &,timefirst,GIMJP,MJP(:,i),MJPES,timemjp,dimen)
                    MJPEST(:,i)=MJPES
                    timemjpto(:,i)=timemjp
                    dimenpro(i)=dimen
                    i=i+1
          cont=1
          do while(cont<=numiter)
              call ipoint(dstates,initp)
              call MJPCU(numobs,dstates,datMJP,timefirst,Q,HV1,MCU1,NJT1)
                    if(cont==1 .and. gib==1)then
                              NJUMPS=NJT0
                              do i=1,NJT0+1
                                        HUMHJP(i)=HV0(i)
                                        UMHJP(i)=MCU0(i)
                              enddo
                    else
                    NJUMPS=NJUMH
                endif
                allocate(HU(NJUMPS+1))
                allocate(UJP(NJUMPS+1))
                do i=1,NJUMPS+1
                    HU(i)=HUMHJP(i)
                    UJP(i)=UMHJP(i)
                enddo
                allocate (HMJ1(NJT1+1))
                allocate(JP1(NJT1+1))
                do i=1,NJT1+1
                    HMJ1(i)=HV1(i)
                    JP1(i)=MCU1(i)
                enddo
call mhunderlying(NJUMPS,NJT1,nstates,dstates,numpro,GIMJP,UJP,JP1,&
                &,MJPEST,HU,HMJ1,timemjpto,dimenpro,NJUMH,UMHJP,HUMHJP,na)
                if(cont==numiter)then
                    RHT=0.0
                    NJTU=0
                    i=1
                do while(i<=numpro)
                  call statime(NJUMH,dstates,nstates,UMHJP,MJPEST(:,i),&
                        &,HUMHJP,timemjpto(:,i),dimenpro(i),RH)
                call stajump(NJUMH,dstates,nstates,UMHJP,MJPEST(:,i),&
                        &,HUMHJP,timemjpto(:,i),dimenpro(i),NJCU)
                    RHT=RHT+RH
                    NJTU=NJTU+NJCU
                    i=i+1
                    enddo
                else
                    continue
                endif
                deallocate(UJP,HU,JP1,HMJ1)
                cont=cont+1
    enddo
                call jumpsU(dstates,NJUMH,UMHJP,MJUMJP)
                call spentU(dstates,NJUMH,UMHJP,HUMHJP,HoldUMJP)
                call posteriorIM(nstates,dstates,NJTU,RHT,BETA,ALPHA,GIMJP)
                call posteriorQ(dstates,MJUMJP,HoldUMJP,ALPHA0,BETA0,Q)
    if(gib>burn_Gibbs)then
                do i=1,dstates
                    do j=1,dstates
                        averaQ(i,j)=(averaQ(i,j)+Q(i,j))
                    enddo
                enddo
                do i=1,nstates
                    do j=1,l
                        AVERAGE(i,j)=AVERAGE(i,j)+GIMJP(i,j)
                    enddo
                enddo
```

```
        else
                continue
        endif
gib=gib+1
enddo

averaQ=averaQ/den*1.0
AVERAGE=AVERAGE/den*1.0
endfile(1)
 close(1)
 call system ("R␣CMD␣BATCH␣intdat.r␣intda.out")
end

!!!!!!!!!!!!!!!!!!!!!
!!!!!Subroutines!!!!!
!!!!!!!!!!!!!!!!!!!!!

! Discrete Observations of MJP
subroutine Datapoint(jumps,time,numobs,hold,CMJP,Obs)
implicit none
INTEGER jumps,i,j,numobs,CMJP(jumps+1),Obs(0:numobs)
REAL time,hold(jumps+2)

Obs(0)=CMJP(1)
i=1
do while(i<=numobs)
        do j=1,jumps+1
                if(i*1.0>=hold(j))then
                        Obs(i)=CMJP(j)
                else
                        goto 1
                endif
        enddo
1 i=i+1
enddo
Obs(numobs)=CMJP(jumps+1)
return
end subroutine

! Calculate the MVE of 0MJP
subroutine MVEMJP(states,mjump,mspent,MVQ)
implicit none
INTEGER i,j,states,njumps,mjump(states,states)
REAL mspent(states),MVQ(states,states)

do i=1,states
        do j=1,states
                MVQ(i,j)=(mjump(i,j))*1.0/mspent(i)
        enddo
enddo
```

```
do i=1,states
        MVQ(i,i)=0.0

        DO J=1,states
                IF (i.NE.j) THEN
                        MVQ(i,i)=MVQ(i,i)-MVQ(i,j)
                ENDIF
        ENDDO
ENDDO
return
end subroutine

!! Simulate the initial point of UMJP
subroutine ipoint(states,point)
implicit none
INTEGER  states,point,j
REAL Unif

j=1
        call random_number(Unif)

do while(j<=states)
   if(((j-1)*1.0)/(states*1.0)<=Unif.and.((j)*1.0)/(states*1.0)>Unif)then
        point=j
        j=states+1
   else
        j=j+1
   endif
enddo
return
end subroutine

! Initial BETA and ALPHA of MJPO
subroutine Ialbe(n,d,alpha,beta)
implicit none
INTEGER n,l,i,j,d
REAL beta(n),alpha(n,d*n)

l=n*d

do i=1,n
        BETA(i)=1.0
        do j=1,l
                ALPHA(i,j)=1.0
        enddo
enddo
return
end subroutine

! Initial BETA and ALPHA of Underlyin
```

```
subroutine Unalbe(d,alp,be)
implicit none
INTEGER i,j,d
REAL be(d),alp(d,d)

do i=1,d
        BE(i)=1.0
        DO j=1,d
                ALP(i,j)=1.0
        enddo
enddo
return
end subroutine

! Generate a Markov jump process
subroutine UMJP(init,m,time,Q,Y,NJT,HT)
INTEGER m,i,j,Y(0:10000),NJT,JT0(m,m),init
REAL Q(m,m),NJ(m),HT(10000),P(m,m),T(0:100000),time,TT,HTT(m)
REAl genexp
external genexp

Y(0)=init
T(0)=0.0
NJT=0
! Transition Matrix
do i=1,m
        do j=1,m
                if(i.NE.j) then
                        P(i,j)=-(Q(i,j)/Q(i,i))
                else
                        P(i,i)=0.0
                endif
        enddo
enddo

TT=0.0
j=1
! Jump processes
do while(TT<=time)
        do i=1,m
                if(Y(j-1)==i)then
                        HT(j)=genexp(-1.0/Q(i,i))
                        T(j)=T(j-1)+HT(j)
                else
                        continue
                endif
        enddo
        TT=TT+HT(j)
        if(TT<time)then
                call newpoint(Y(j-1),P,m,Y(j))
        NJT=NJT+1
        j=j+1
    else
        continue
    endif
enddo
suma=0.0
do i=1,NJT
suma=suma+HT(i)
enddo
HT(NJT+1)=time-suma
HTT=0.0
j=1
do while(j<=m)
    do i=0,NJT
        if(Y(i)==j)then
            HTT(j)=HTT(j)+HT(i+1)
        else
            continue
        endif
    enddo
j=j+1
enddo
JT0=0
j=1
do while(j<=m)
    do i=0,NJT-1
        if(Y(i)==j)then
            if(Y(i+1).NE.j)then
                JT0(j,Y(i+1))=JT0(j,Y(i+1))+1
            else
                JT0(j,j)=0
            endif
        else
            continue
        endif
    enddo
j=j+1
enddo

return
end subroutine

! Generate the new point in the MJP
subroutine newpoint(old,P,numstates,new)
implicit none
REAL P(numstates,numstates),suma,U
INTEGER i,numstates,old,new

  call random_number(U)
```

```
suma=0.0
i=1
do while(i<=numstates)
        suma=P(old,i)+suma
        if(U<suma)then
                new=i
                i=numstates+1
        else
                i=i+1
        endif
enddo

return
end subroutine

! Draw a Q of Underlying from the prior distribution
subroutine priorQ(d,ALPHA,BETA,Q)
implicit none
INTEGER d,i,j,k,kk
REAL alpha(d,d),beta(d),Q(d,d)
!     .. External Functions ..
REAL gengam
external gengam
Q=0.0

do i=1,d
        do j=1,d
                Q(i,j)=gengam(beta(i),alpha(i,j))
         enddo
enddo

do i=1,d
        Q(i,i)=0.0
        DO J=1,d
                IF (i.NE.j) THEN
                        Q(i,i)=Q(i,i)-Q(i,j)
                ENDIF
        ENDDO
ENDDO

return
end subroutine

! Draw a Q of Underlying from the posterior distribution
subroutine posteriorQ(d,MJUMJP,HoldUMJP,ALPHA,BETA,Q)
implicit none
INTEGER d,i,j,k,kk,MJUMJP(d,d)
REAL alpha(d,d),beta(d),Q(d,d),HoldUMJP(d)
!     .. External Functions ..
REAL gengam
```

```
 external gengam

Q=0.0
do i=1,d
        do j=1,d
                Q(i,j)=gengam((beta(i)+HoldUMJP(i)),alpha(i,j)+MJUMJP(i,j))
         enddo
enddo

do i=1,d
        Q(i,i)=0.0
        DO J=1,d
                IF (i.NE.j) THEN
                        Q(i,i)=Q(i,i)-Q(i,j)
                ENDIF
        ENDDO
ENDDO
return
end subroutine

! Calculate the stats of jumps of underlying process
subroutine jumpsU(d,njumps,jumpU,MJUMJP)
implicit none
INTEGER d,njumps,jumpU(njumps+1),MJUMJP(d,d),i

MJUMJP=0
do i=1,njumps
        MJUMJP(jumpU(i),jumpU(i+1))=MJUMJP(jumpU(i),jumpU(i+1))+1
enddo
return
end subroutine

! Calculate the statistics of time of underlying process
subroutine spentU(d,njumps,jumpU,hU,HoldUMJP)
implicit none
INTEGER d,njumps,i,jumpU(njumps+1)
REAL hU(njumps+1),HoldUMJP(d)

HoldUMJP=0.0
i=1
do while(i<=njumps+1)
        HoldUMJP(jumpU(i))=HoldUMJP(jumpU(i))+hU(i)
        i=i+1
enddo
return
end subroutine

! Draw a QMJP from the prior distribution
subroutine priorIM(n,d,ALPHA,BETA,QMJ)
implicit none
```

```
INTEGER n,d,l,i,j,k,kk
REAL alpha(n,n*d),beta(n),QMJ(n,n*d)
!     .. External Functions ..
REAL gengam
  external gengam

 l=n*d
QMJ=0.0

do i=1,n
        do j=1,l
                QMJ(i,j)=gengam(beta(i),alpha(i,j))
        enddo
enddo

do i=1,n
        do j=1,l
                do k=1,d
                        if(j.eq.(i+(k-1)*n))then
                                QMJ(i,j)=0.0
                        endif
                enddo
        enddo
        do k=1,d
                kk=i+(k-1)*n
                        do j=kk-i+1,kk-i+n
                                if(j.ne.kk)then
                                        QMJ(i,kk)=QMJ(i,kk)-QMJ(i,j)
                                endif
                        enddo
        enddo
enddo
return
end subroutine

! Calculate the intensity matrix of MJPs
subroutine posteriorIM(n,d,MNJ,MTS,beta,alpha,QMJ)
implicit none
INTEGER n,d,l,MNJ(n,n*d),i,j,k,kk,mod
REAL alpha(n,n*d),beta(n),QMJ(n,n*d),MTS(n,d)
REAL gengam
external gengam
 l=n*d
QMJ=0.0
do i=1,n
        do j=1,l
                call modu(j,n,mod)
                QMJ(i,j)=gengam(beta(i)+MTS(i,mod+1),alpha(i,j)+MNJ(i,j))
        enddo
enddo
```

```
do i=1,n
        do j=1,l
                do k=1,d
                        if(j.eq.(i+(k-1)*n))then
                                QMJ(i,j)=0.0
                        endif
                enddo
        enddo
        do k=1,d
                kk=i+(k-1)*n
                        do j=kk-i+1,kk-i+n
                                if(j.ne.kk)then
                                        QMJ(i,kk)=QMJ(i,kk)-QMJ(i,j)
                                endif
                        enddo
        enddo
enddo
return
end

! Metropolis-hastings of underlying process
subroutine mhunderlying(jumpU0,jumpU1,n,d,numpro,QMJ,JP0,JP1,MJP,HMJ0,HMJ1,&
                        &,timeMJ,dimen,NJU,UMHJP,HUMHJP,num_acep)
implicit none
INTEGER jumpU0,jumpU1,jumpMJ,n,d,numpro,i,j,l,modd,modn,NJU,MH,UMHJP(10000)
INTEGER NJCU0(n,n*d),NJCU1(n,n*d),JP0(jumpU0+1),JP1(jumpU1+1),MJP(10000,numpro)
INTEGER num_acep,dimen(numpro),NJT0(n,n*d),NJCU(n,n*d),NJT1(n,n*d)
REAL QMJ(n,n*d),probability,prob,Unif,tim0(jumpU0+2),suma,tim1(jumpU1+2)
REAL HMJ0(jumpU0+1),HMJ1(jumpU1+1),timeMJ(10000,numpro),RH0(n,d),RH1(n,d)
REAL RHT1(n,d),RHT0(n,d),RH(n,d),HUMHJP(10000)

call timet(jumpU0,HMJ0,tim0)
call timet(jumpU1,HMJ1,tim1)
RHT0=0.0
RHT1=0.0
NJT0=0
NJT1=0
i=1
do while(i<=numpro)
        call statime(jumpU0,d,n,JP0,MJP(:,i),HMJ0,timeMJ(:,i),dimen(i),RH0)
        call stajump(jumpU0,d,n,JP0,MJP(:,i),HMJ0,timeMJ(:,i),dimen(i),NJCU0)
        call statime(jumpU1,d,n,JP1,MJP(:,i),HMJ1,timeMJ(:,i),dimen(i),RH1)
        call stajump(jumpU1,d,n,JP1,MJP(:,i),HMJ1,timeMJ(:,i),dimen(i),NJCU1)
        RHT0=RHT0+RH0
        RHT1=RHT1+RH1
        NJT0=NJT0+NJCU0
        NJT1=NJT1+NJCU1
        i=i+1
enddo
```

```
l=n*d
prob=1.0
RH=RHT1-RHT0
NJCU=NJT1-NJT0
call sumv(n,d,RH,suma)
do i=1,n
        do j=1,l
                call modu(j,n,modn)
                if(i==j-modn*n)then
                        prob=prob
                else
                 prob=QMJ(i,j)**(NJCU(i,j))
                 prob=prob*exp(-QMJ(i,j)*RH(i,modn+1))*prob
                endif
       enddo
enddo
probability=min(1.0,prob)
 call random_number(Unif)
if(Unif<=probability)then
        MH=1
        NJU=jumpU1
        do i=1,NJU+1
                UMHJP(i)=JP1(i)
                HUMHJP(i)=HMJ1(i)
        enddo
        num_acep=num_acep+1
else
        MH=0
        NJU=jumpU0
        do i=1,NJU+1
                UMHJP(i)=JP0(i)
                HUMHJP(i)=HMJ0(i)
        enddo
endif
return
end subroutine

! sum times
subroutine timet(t,V,tim)
implicit none
INTEGER t,i
real V(t+1),tim(t+2),sum

tim(1)=0.0
sum=0.0
i=2
do while (i<=(t+2))
        sum=sum+V(i-1)
        tim(i)=sum
        i=i+1
```

```
enddo

return
end subroutine

! Calculate the statistics of the time spent in each state cond
subroutine statime(a,d,n,U,MJ,TU,TMJ,dimen,RS)
implicit none
INTEGER a,b,n,d,U(a+1),MJ(10000),i,j,k,kk,l,npro,mark,dimen
REAL TU(a+1),TMJ(10000),RH(d,n),timesU(a+2),RS(n,d),suma

call timet(a,TU,timesU)
RS=0.0
i=1
j=1
suma=0.0
do while(i<=dimen)
        do while(j<=a+1)
                k=U(j)
                kk=MJ(i)
                if(TMJ(i+1)<=TimesU(j+1))then
                        RS(kk,k)=RS(kk,k)+TMJ(i+1)-suma
                        suma=TMJ(i+1)
                        goto 15
                else
                        RS(kk,k)=RS(kk,k)+timesU(j+1)-suma
                        suma=timesU(j+1)!
                        j=j+1
                endif
        enddo
15 i=i+1
enddo
return
end subroutine

! Calculate the statistics of the number of jump
subroutine stajump(a,d,n,U,MJ,TU,TMJ,dimen,numjump)
implicit none
INTEGER a,b,n,d,U(a+1),MJ(10000),i,intjump,numjump(n,n*d),npro,k,l,j,mod,dimen
REAL TU(a+2),TMJ(10000),timesU(a+2)

l=n*d
call timet(a,TU,timesU)
numjump=0
j=1
do while(j<=dimen)
        i=1
        do while(i<=a+1)
        if(timesU(i+1)>TMJ(j+1))then
         k=U(i)
         numjump(MJ(j),MJ(j+1)+(k-1)*n)=numjump(MJ(j),MJ(j+1)+(k-1)*n)+1
                i=a+2
        else
        continue
        endif
        i=i+1
        enddo
        j=j+1
enddo

do i=1,n
        do j=1,l
                call modu(j,n,mod)
                if(i==j-mod*n)then
                        numjump(i,j)=0
                else
                        continue
                endif
        enddo
enddo
return
end subroutine

! Simulate a MJP conditioned
subroutine MJPC(numobs,n,d,NJ,UMJ,times,QMJ,MJPCU,TSMJPCU,JMJPCU)
implicit none
INTEGER numobs,i,j,l,n,d,k,mod,NJ,MJPCU(numobs),UMJ(NJ+1),initp
INTEGER a,b,NJT,JMJPCU(n,n*d),CMJP(100000)
REAL QMJ(n,n*d),HTCU(numobs),P(n,n*d),times(NJ+2),time,PCU(n,n)
REAL QCU(n,n),suma,TSMJPCU(n,d),mean,x,CHT(100000)
REAL gengam,genexp
external gengam,genexp
INTEGER , allocatable:: CMJPA(:)
REAL , allocatable:: CHTUA(:)

JMJPCU=0
l=n*d
TSMJPCU=0.0
do i=1,n
        do j=1,l
                call modu(j,n,mod)
                if(i==j-mod*n)then
                        P(i,j)=0.0
                else
                        P(i,j)=-(QMJ(i,j)/QMJ(i,i+mod*n))
                endif
        enddo
enddo

call ipoint(n,initp)
```

```
CMJP(1)=initp
time=0.0
j=1
NJT=0
suma=0.0
! Jump processes
do while(time<=times(NJ+2))
        i=1
        do while(i<=NJ+1)
        if(times(i+1)>=time)then
                k=UMJ(i)
                do a=1,n
                        do b=1,n
                                PCU(a,b)=P(a,b+(k-1)*n)
                                QCU(a,b)=QMJ(a,b+(k-1)*n)
                        enddo
                enddo
                CHT(j)=genexp(-1.0/QCU(CMJP(j),CMJP(j)))
                time=time+CHT(j)
                i=NJ+2
                if(time<=times(NJ+2))then
                        TSMJPCU(CMJP(j),k)=TSMJPCU(CMJP(j),k)+CHT(j)
                else
                        TSMJPCU(CMJP(j),k)=TSMJPCU(CMJP(j),k)+times(NJ+2)-suma
                endif
                suma=suma+CHT(j)
        else
                continue
        endif
        i=i+1
enddo

if(time<times(NJ+2))then
        call newpoint(CMJP(j),PCU,n,CMJP(j+1))
        NJT=NJT+1
        j=j+1
else
        continue
endif
enddo
allocate(CMJPA(NJT+1))
allocate(CHTUA(NJT+2))

suma=0.0
do i=2,NJT+1
        suma=CHT(i-1)+suma
        CHTUA(i)=suma
enddo

do i=1,NJT+1
```

```
          CMJPA(i)=CMJP(i)
enddo

CHTUA(1)=0.0
CHTUA(NJT+2)=times(NJ+2)
call Datapoint(NJT,times(NJ+2),numobs-1,CHTUA,CMJPA,MJPCU)
deallocate(CMJPA,CHTUA)

return
end subroutine

! Calculate the enter of integer less o equal to jump
subroutine integerjump(timejump,numjump,times,intjump)
implicit none
REAL times(numjump+2),comp
INTEGER timejump,numjump,intjump,i

comp=timejump*1.0
i=1
do while(i<=(numjump+2))
if(comp<=times(i))then
 intjump=i
          goto 12
else
i=i+1
endif
enddo

12 return
end

! Transpose Matrix
subroutine Transmat(a,b,OM,TM)
implicit none
INTEGER a,b,i,j
REAL OM(a,b),TM(b,a)

do i=1,b
          do j=1,a
                    TM(i,j)=OM(j,i)
          enddo
enddo
return
end

! calculte module
subroutine modu(a,b,mod)
implicit none
REAL div
INTEGER a,b,mod,i
```

```
div=a*1.0/b*1.0

i=0
do while(i<=100000)
        if(i<=div.and.div<=i+1)then
                mod=i
                goto 13
        else
                i=i+1
        endif
enddo
13 return
end subroutine

! sum of elemenst of vector
subroutine sumv(rele,cele,vector,suma)
implicit none
INTEGER rele,cele,i,j
REAL vector(rele,cele),suma

suma=0.0
do i=1,rele
        do j=1,cele
                suma=suma+vector(i,j)
        enddo
enddo
return
end subroutine

! Distribution of MT
subroutine DMT(n,d,Q,QU,vector,vq)
implicit none
INTEGER n,d,l,mod,i,j,k,m,r
REAL Q(n,n*d),P(n,n*d),Vector((n-1)*n*d),QU(d,d),vq(d*(d-1))
REAL PU(d,d)

l=n*d
k=(n-1)*n*d
do i=1,n
        do j=1,l
                call modu(j,n,mod)
        if(i==j-mod*n)then
                P(i,j)=0.0
        else
                P(i,j)=-(Q(i,j)/Q(i,i+mod*n))
        endif
        enddo
enddo
m=1
```

```
do i=1,n
        do j=1,l
                call modu(j,n,mod)
        if(i==j-mod*n)then
                continue
        else
                vector(m)=P(i,j)
                m=m+1
        endif
        enddo
enddo

do i=1,d
        do j=1,d
        if(i==j)then
                PU(i,j)=0.0
        else
                PU(i,j)=-(QU(i,j)/QU(i,i))
        endif
        enddo
enddo

r=1
do i=1,d
        do j=1,d
        if(i==j)then
                continue
        else
                vq(r)=PU(i,j)
                r=r+1
        endif
        enddo
enddo
return
end subroutine

! calculate the MVE of MJPC
subroutine MVEMJPC(n,d,mjump,mspent,MVQ)
implicit none
INTEGER i,j,n,d,l,mjump(n,n*d),mod,k,kk
REAL mspent(n,d),MVQ(n,n*d)

l=n*d
do i=1,n
        do j=1,l
                call modu(j,n,mod)
                MVQ(i,j)=(mjump(i,j))*1.0/mspent(i,mod+1)
        enddo
enddo
```

```
do i=1,n
        do j=1,l
                do k=1,d
                        if(j.eq.(i+(k-1)*n))then
                                MVQ(i,j)=0.0
                        endif
                enddo
        enddo
        do k=1,d
                kk=i+(k-1)*n
                do j=kk-i+1,kk-i+n
                if(j.ne.kk)then
                        MVQ(i,kk)=MVQ(i,kk)-MVQ(i,j)
                endif
                enddo
        enddo
enddo
return
end subroutine

!  Simulate paths of MJPC
subroutine MJPCDO(nobs,n,d,NJ,UMJ,times,timed,QMJ,MJPDO,MJPCU,timemjp,dimen)
implicit none
INTEGER numobs,i,j,l,n,d,k,kk,mod,NJ,UMJ(NJ+1),initp,CMJP(100000)
INTEGER a,b,NJT,JMJPCU(n,n*d),state,mjpdo(nobs),CMJPC(100000)
INTEGER nobs,MJPCU(100000),crit,z,r,dimen
REAL QMJ(n,n*d),P(n,n*d),times(NJ+2),time,PCU(n,n),CHT(100000),CHTC(100000)
REAL QCU(n,n),suma,TSMJPCU(n,d),mean,timed(nobs),timemjp(100000),x,timec(100000)
INTEGER , allocatable:: CMJPA(:)
REAL , allocatable:: CHTUA(:),timecont(:)
REAL genexp
external genexp

CHTC=0.0
CHT=0.0
CMJPC=0
CMJP=0
JMJPCU=0
l=n*d
TSMJPCU=0.0,&
do i=1,n
        do j=1,l
                call modu(j,n,mod)
        if(i==j-mod*n)then
                P(i,j)=0.0
        else
                P(i,j)=-(QMJ(i,j)/QMJ(i,i+mod*n))
        endif
        enddo
enddo
```

```
! Jump processes
state=1
kk=1
NJT=0
 CMJP(1)=MJPDO(state)
 CMJPC(1)=MJPDO(state)
do while(state<nobs)
  crit=kk
 CMJPC(kk)=MJPDO(state)
time=timed(state),&
j=1
 CMJP(1)=MJPDO(state)
suma=timed(state)

do while(time<=timed(state+1))
i=1
        do while(i<=NJ+1)
                if(times(i+1)>=time)then
                        k=UMJ(i)
                do a=1,n
                        do b=1,n
                                PCU(a,b)=P(a,b+(k-1)*n)
                                QCU(a,b)=QMJ(a,b+(k-1)*n)
                        enddo
                enddo
                mean=-1.0/QCU(CMJP(j),CMJP(j))
                CHT(j)=genexp(mean)
                time=time+CHT(j)
                i=NJ+2
                        if(time<=timed(state+1))then
                                CHTC(kk)=CHT(j),&
                        else
                                CHTC(kk)=timed(state+1)-suma
                        endif
                        suma=suma+CHT(j)

                else
                        continue
                endif
                i=i+1
                enddo
                if(time<timed(state+1))then
                        z=1
                        do while(z<=NJ+1)
                        if(times(z+1)>=time)then
                                k=UMJ(z)
                                z=NJ+2
                        else
                                continue
                                endif
                                          z=z+1
                                enddo
                                call  newpoint(CMJP(j),PCU,n,CMJP(j+1))
                                CMJPC(kk+1)=CMJP(j+1)
                                j=j+1
                                kk=kk+1
                      else
                                continue
                      endif
          enddo
          if(CMJP(j)==MJPDO(state+1))then
                      state=state+1
                      kk=kk+1
          else
                      state=state
                      kk=crit
                      do  r=kk+1,9
                                CHTC(r)=0.0
                                CMJPC(r)=0
                      enddo
          endif
enddo

NJT=kk-2
call  timet(njt,CHTC,timec)
allocate(CHTUA(NJT+1))
allocate(CMJPA(NJT+1))
allocate(timecont(NJT+2))
 CHTUA=0.0
 CMJPA=0
j=1
do  i=1,NJT+1
          do  while(j<=99999)
                      CHTUA(i)=CHTC(j)+CHTUA(i)
                      CMJPA(i)=CMJPC(j)
                      if(CMJPC(j)==CMJPC(j+1))then
                                continue
                      else
                                j=j+1
                                goto  14
                      endif
                      j=j+1
          enddo
14  j=j
enddo

i=1
do  while(i<=(NJT+1))
          if(i==NJT+1)then
```

```
                dimen=NJT
        else
        if(CMJPA(i)==0)then
                dimen=i-1
                i=NJT+2
        else
                continue
        endif
        endif
        i=i+1
enddo

  call timet(njt ,CHTUA,timecont)
 do i=1,dimen
        MJPCU(i)=CMJPA(i)
        timemjp(i)=timecont(i)
enddo
timemjp(dimen+1)=timecont(dimen+1)
deallocate(CHTUA,CMJPA,timecont)
return
end subroutine

! Simulate paths of MJPC
subroutine MJPCU(nobs,d,DOMJP,times ,Q,CHTUA,CMJPA,NJT),&
implicit none
INTEGER nobs,i,j,l,n,d,k,kk,crit ,state ,NJ,NJMJP(d,d),IND,NJT
INTERGER DOMJP(nobs),CMJPC(10000),CMJPA(10000)
REAL times(nobs),Q(d,d),HT(10000),P(d,d),ST(10000),RH(d),CHTUA(10000)
REAl genexp,suma,time,x
external genexp

do i=1,d
        do j=1,d
                if(i==j)then
                        P(i,j)=0.0
                else
                        P(i,j)=-(Q(i,j)/Q(i,i))

                endif
        enddo
enddo

!Jump processes
state=1
kk=1
suma=0.0
IND=0
do while(state<nobs)
        NJ=0
        crit=kk
        CMJPC(kk)=DOMJP(state)
        suma=0.0
        time=times(state)
        do while(time<=times(state+1))
                do i=1,d
                if(CMJPC(kk)==i)then
                        HT(kk)=genexp(-1.0/Q(i,i))
                else
                        continue
                endif
                enddo
                time=time+HT(kk)
                if(time<times(state+1))then
                        call newpoint(CMJPC(kk),P,d,CMJPC(kk+1))
                        suma=suma+HT(kk)
                        kk=kk+1,&
                        NJ=NJ+1
                else
                        continue
                endif
        enddo
        if(CMJPC(kk)==DOMJP(state+1))then
                HT(kk)=1.0-suma
                kk=kk+1
                state=state+1
        else
                state=state
                kk=crit
        endif
enddo

NJT=kk-nobs
j=1
 CHTUA=0.0
do i=1,NJT+1
        do while(j<=10000)
                CHTUA(i)=HT(j)+CHTUA(i)
                CMJPA(i)=CMJPC(j)
                if(CMJPC(j)==CMJPC(j+1))then
                        continue         ,&
                else
                        j=j+1
                        goto 14
                endif
                j=j+1
        enddo
14 j=j
enddo

  call timet(NJT,CHTUA,ST)
```

```
RH=0.0

do i=1,kk-1
          RH(CMJPC(i))=RH(CMJPC(i))+HT(i)
enddo

NJMJP=0
do i=1,kk-2
          NJMJP(CMJPC(i),CMJPC(i+1))=NJMJP(CMJPC(i),CMJPC(i+1))+1
enddo
do i=1,d
          do j=1,d
                    if(i==j)then
                              NJMJP(i,j)=0
                    else
                              continue
                    endif
          enddo
enddo
return
end

! Auxiliar Variable
subroutine auxiliar(nobs,MJPS,UNMJP,NJM,NJU,timeMJ,timeU,time,Y,W)
implicit none
INTEGER nobs,NJM,NJU,MJPS(NJM+1),UNMJP(NJU+1),i,state,Y(nobs),W(nobs)
REAL timeMJ(NJM+2),timeU(NJU+2),time(nobs),Yt(nobs),Wt(nobs),suma

Y(1)=MJPS(1)
Yt(1)=0.0
W(1)=UNMJP(1)
Wt(1)=0.0
state=2
do while(state<=nobs)
          suma=time(state)
                    i=1
                    do while(i<=NJM+1)
                     if(timeMJ(i+1)<suma)then
                              i=i+1
                    else
                              if(i==1)then
                                        Y(state)=MJPS(i)
                                        Yt(state)=timeMJ(i)
                              else
                                        Y(state)=MJPS(i-1)
                                        Yt(state)=timeMJ(i)
                              endif
                              i=NJM+2
                    endif
```

```
                enddo
        state=state+1
enddo

state=2
do while(state<=nobs)
        suma=Yt(state)
        i=1
        do while(i<=NJU+1)
        if(timeU(i+1)<suma)then
                i=i+1
        else
                if(i==1)then
                        W(state)=UNMJP(i)
                        Wt(state)=timeU(i),&
                else
                        W(state)=UNMJP(i-1)
                        Wt(state)=timeU(i)
                endif
                i=NJU+2
        endif
        enddo
        state=state+1
enddo

return
end subroutine

! Augmentation data
SUBROUTINE augdata(nobs,npro,d,n,Y,W,MJP,Q,IM,Z)
implicit none
INTEGER nobs,npro,d,n,Y(nobs,npro),W(nobs,npro),Z(nobs,npro)
INTEGER i,j,WA(nobs,npro),l,mod,MJP(nobs,npro),k
REAL Q(d,d),IM(n,d*n),P(d,d),PMJP(n,d*n),vprob(d),suma

do i=1,d,&
        do j=1,d
                if(i==j)then
                        P(i,j)=0.0
                else
                        P(i,j)=-(Q(i,j)/Q(i,i))
                endif
        enddo
enddo

l=n*d
do i=1,n
        do j=1,l
                call modu(j,n,mod)
                if(i==j-mod*n)then
```

```
                        PMJP(i,j)=0.0
                else
                        PMJP(i,j)=-(IM(i,j)/IM(i,i+mod*n))
                endif
        enddo,&
enddo
do j=1,npro
        do i=1,nobs
                k=1
                suma=0.0
                do while(k<=d)
                  vprob(k)=PMJP(Y(i,j),MJP(i,j)+n*(k-1))*P(W(i,j),k)
                  suma=vprob(k)+suma
                        k=k+1
                enddo
                if(suma==0)then
                        Z(i,j)=W(i,j)
                else
                        vprob=vprob/suma
                        call newvec(W(i,j),vprob,d,Z(i,j))
                endif
        enddo
enddo
return
end subroutine

!  Simulate paths of MJPC
subroutine MJPCAUX(nobs,n,d,NJ,UMJ,times,timed,QMJ,MJPDO,MJPCU,&
                   &,timemjp,dimen)
implicit none
INTEGER numobs,i,j,l,n,d,k,kk,mod,NJ,UMJ(NJ+1),initp,nobs,dimen,z,r
INTEGER a,b,NJT,JMJPCU(n,n*d),state,mjpdo(nobs),crit
INTEGER CMJPC(100000),MJPCU(100000),CMJP(100000)
REAL QMJ(n,n*d),P(n,n*d),times(NJ+2),time,PCU(n,n)
REAL QCU(n,n),suma,TSMJPCU(n,d),mean,timed(nobs),x
REAL CHT(100000),CHTC(100000),timec(100000)timemjp(100000)
INTEGER , allocatable :: CMJPA(:)
REAL , allocatable :: CHTUA(:),timecont(:)
REAL genexp
external genexp

CHTC=0.0
CHT=0.0
CMJPC=0
CMJP=0
JMJPCU=0
l=n*d
TSMJPCU=0.0
do i=1,n
        do j=1,l
```

```
                call modu(j,n,mod)

                if(i==j-mod*n)then
                        P(i,j)=0.0
                else
                        P(i,j)=-(QMJ(i,j)/QMJ(i,i+mod*n))
                endif
        enddo
enddo

! Jump processes
state=1
kk=1
NJT=0
 CMJP(1)=MJPDO(state)
 CMJPC(1)=MJPDO(state)     !
!!!!!!!
do while(state<nobs)
  crit=kk
 CMJPC(kk)=MJPDO(state)
        time=timed(state)
        j=1
        CMJP(1)=MJPDO(state)
        suma=timed(state)
        do while(time<=timed(state+1))
                i=1
                do while(i<=NJ+1)
                if(times(i+1)>=time)then
                        k=UMJ(i)
                        do a=1,n
                                do b=1,n
                                        PCU(a,b)=P(a,b+(k-1)*n)
                                        QCU(a,b)=QMJ(a,b+(k-1)*n)
                                enddo
                        enddo
                        mean=-1.0/QCU(CMJP(j),CMJP(j))
                        CHT(j)=genexp(mean)
                        time=time+CHT(j)
                        i=NJ+2
                        if(time<=timed(state+1))then
                                CHTC(kk)=CHT(j),&
                        else
                                CHTC(kk)=timed(state+1)-suma
                        endif
                                suma=suma+CHT(j)
                else
                        continue
                endif

                i=i+1
                enddo

                if(time<timed(state+1))then
                        z=1
                        do while(z<=NJ+1)
                        if(times(z+1)>=time)then
                                k=UMJ(z)
                                z=NJ+2
                        else,&
                                continue
                        endif
                        z=z+1
                        enddo
                        call newpoint(CMJP(j),PCU,n,CMJP(j+1))
                        CMJPC(kk+1)=CMJP(j+1)
                        j=j+1
                        kk=kk+1
                else
                        continue
                endif
        enddo
        if(CMJP(j)==MJPDO(state+1))then
                state=state+1
                kk=kk+1
        else
                state=state
                kk=crit
                do r=kk+1,9
                        CHTC(r)=0.0
                        CMJPC(r)=0
                enddo
        endif

enddo
NJT=kk-2

call timet(njt,CHTC,timec)
allocate(CHTUA(NJT+1))
allocate(CMJPA(NJT+1))
allocate(timecont(NJT+2))
 CHTUA=0.0
 CMJPA=0
j=1
do i=1,NJT+1
        do while(j<=99999)
                CHTUA(i)=CHTC(j)+CHTUA(i)
                CMJPA(i)=CMJPC(j)
        if(CMJPC(j)==CMJPC(j+1))then
                continue
        else
```

```
                j=j+1
                goto 14
        endif
        j=j+1
        enddo
14 j=j
enddo
i=1
do while(i<=NJT)
        if(CMJPA(i)==0)then
                dimen=i-1
        i=NJT+1
        else
                continue
        endif
        i=i+1
enddo
 call timet(njt,CHTUA,timecont)

 do i=1,dimen
        MJPCU(i)=CMJPA(i)
        timemjp(i)=timecont(i)
enddo
timemjp(dimen+1)=timecont(dimen+1)
deallocate(CHTUA,CMJPA,timecont)
return
end subroutine

!  Clasification of states
subroutine Clastates(n,dat,states)
implicit none
INTEGER n,states(n),i
REAL dat(n),x1,x2,d1,d2,r

x1=minval(dat)
x2=maxval(dat)
r=(x2-x1)/3.0
d1=x1+r
d2=d1+r
do i=1,n
        if(dat(i)<=d1)then
                states(i)=1
        elseif(dat(i)>d1 .and. dat(i)<=d2)then
                states(i)=2
        else
                states(i)=3
        endif
enddo
return
end subroutine
```

Appendix B: Code in Fortran of Integrated Diffusion Processes

Here, we present the code that was used to parameter estimation of integrated diffusion processes.

```
  ! Program Integrate_Diffusions

 implicit none
INTEGER numdat, rep_metro, numaux, all, n, burn, rep_em
 parameter(numdat=1500)
 parameter(burn=1000)
 parameter(rep_metro=10000)
 parameter(rep_em=100)
INTEGER i, j, num_col, or, iter, acc, it, mc, ms(numdat), mm, p1, p2, dif
REAL coef(5), mu(rep_em+1), sig(rep_em+1), int_val, nor_val, mu0, average
REAL Col, Col1, Col2, Col3, Col4, cons, S0, del, delta, sigma0, as(5), acoef(5)
REAL sigma(rep_em+1), sig_up, mu_up, datup(0:numdat), normals(numdat)
REAL dat(0:numdat), datas(numdat), int_dif(numdat), dat_ini(0:numdat)
REAL tau0, tau(rep_em+1), rec_val, lambda, hes, val, dat2(0:numdat)
REAL up_sigma, up_mu, up_tau, gennor, init, inte, intec
REAL, allocatable::dat_pro(:), Bridge(:), current(:), path(:)
REAL, allocatable::B2(:), normalpar(:), bri_exa(:)
 character*20 answer, file1, file2
 external advnst, gennor

 ! External Functions
REAL genchi, genbet
 integer ignpoi
EXTERNAL genchi, ignpoi, genbet
 ! External subroutines
 external setgnm, genmn

 ! SIZE OF BLOCK
delta=1.0
lambda=20.0
```

```
! REAL PARAMETERS
sigma0=0.5
mu0=0.1
tau0=sqrt(1.25)

! INITIAL PARAMETERS

tau(1)=tau0
mu(1)=0.1
sigma(1)=0.5


! Simuling the hidden data
call  simil(init ,delta ,numdat,mu0,sigma0 ,dat)
call mean(dat ,numdat+1,average)
dat=dat-average
call  int_com(numdat,delta ,dat ,tau0 ,int_dif)
int_dif=int_dif-average
write(1,*) dat
write(2,*) int_dif
endfile(1)
endfile(2)
close(1)
close(2)

! EM-ALGORITHM
iter=1

do while(iter<=rep_em)
  call  simil(init ,delta ,numdat,mu(iter),sigma(iter),datup)
  call mean(datup,numdat+1,average)
  acc=0
  coef=0.0
  intec=0.0
  it=1
  do while(it<=rep_metro)
    call sel_block(lambda,numdat,mc,ms)
    mm=1
    do while(mm<=mc)
      call pps(mm,numdat,ms,p1,p2)
      dif=p2-p1
      allocate(Bridge(dif+1),current(dif+1),path(dif+1),B2(dif))
      allocate(normalpar(dif),bri_exa(dif+1))
      if(mm==1)then
        del=1.0/(dif*1.0)
        call  ou2(datup(p2),dif,mu(iter),delta,sigma(iter),B2)
        call  com_bri1(datup(p2),dif,B2,Bridge)
      else if(1<mm .and. mm<mc)then
        if(dif==2)then
```

```
                goto 1
        endif
        call  bri(datup(p1),datup(p2),dif-1,mu(iter),delta,sigma(iter),Bridge)
        else
                del=1.0/(dif*1.0)
                call ou1(datup(p1),dif,mu(iter),delta,sigma(iter),B2)
                call   com_bri2(datup(p1),dif,B2,Bridge)
        endif
        1 call  initial(p1,p2,dif,numdat,datup,current)
        call   accept(delta,acc,tau(iter),p1,dif+1,numdat,int_dif,&
                     &,current,Bridge,path,normalpar)
!update the data
        call up_data(numdat,p1,p2,path,datup)
        call up_normal(numdat,p1,p2,normalpar,normals)
        deallocate(Bridge,current,path,B2,normalpar,bri_exa)
        mm=mm+1
    enddo
    call mean(datup,numdat+1,average)
    if(it>burn)then
        call  par_coeff(rep_metro,burn,sigma(iter),numdat,delta,int_dif,datup,as)
               coef=as+coef
    endif
    it=it+1
  enddo
  call update(coef,sigma0,numdat,up_mu,up_sigma,up_tau)
  iter=iter+1
  mu(iter)=up_mu
  sigma(iter)=up_sigma
  tau(iter)=up_tau

enddo

end program

!!!!!!!!!!!!!!!!!!!!!
!!!!!Subroutines!!!!!
!!!!!!!!!!!!!!!!!!!!!

! Milstein scheme
subroutine simil(ini,del,a,mu,sig,data)!(init,delta,numdat,mu0,sigma0,dat)
implicit none
INTEGER a,i
REAL data(0:a),sig,mu,j,del,nor(a)
REAL gennor,inter,ini,sv,inva_den
external gennor

sv=sqrt(del)
inter=0.0
inva_den=sqrt((sig**2)/(2*mu))
print*,'in',inva_den
```

```
data(0)=gennor(0.0,inva_den)

do i=1,a
        nor(i)=sig*gennor(0.0,sv)
        data(i)=data(i-1)*(1.0-mu*del)+nor(i)
        inter=inter+nor(i)
enddo

inter=inter/(a*1.0)

return
end subroutine

! Integrated data

subroutine integrate(del,y1,y2,tau,rect,nor,fun)
implicit none
INTEGER i
REAL rect,y1,y2,fun,gennor,tau,nor,del
external gennor

nor=gennor(0.0,tau)
rect=((y2+y1)*del)/(2.0)
fun=rect+nor
return
end subroutine

! complete integrate data

subroutine int_com(n,del,dat,tau,intd) !(numdat, delta , dat , tau0 , int_dif)
implicit none
INTEGER i,n
REAL dat(0:n),tau,intd(n),rec_val,nor_val,int_val,del

do i=1,n
        call integrate(del,dat(i-1),dat(i),tau,rec_val,nor_val,int_val)
        intd(i)=int_val
enddo
return
end subroutine

! partial coeff of polynomial
subroutine par_coeff(rep,burn,sigma,numdat,del,intdat,dat,as)
implicit none
INTEGER numdat,numaux,rep,burn,i
REAL dat(0:numdat),coef(0:6),as(5),intdat(numdat),intdat2(numdat)
REAL del,dat2(0:numdat),nor2(numdat),tau,gennor,sigma
REAL inte1,inte2,rect1,rect2,norm,fun,a1,a2,a3,a4,a5

do i=0,numdat
```

```
        dat2(i)=dat(i)**2
enddo

a3=0.0
inte1=0.0
inte2=0.0
do i=1,numdat
        call integrate(del,dat(i-1),dat(i),0.2,rect1,norm,fun)
        call integrate(del,dat2(i-1),dat2(i),0.2,rect2,norm,fun)
        a3=intdat(i)*rect1+a3
         inte1=inte1+rect1**2
        inte2=inte2+rect2
enddo

do i=1,numdat
        intdat2(i)=intdat(i)**2
enddo
as(1)=((dat2(0)-dat2(numdat))+((sigma)**2)*(numdat))/(rep-burn)
as(2)=(2*inte2)/(rep-burn)
as(3)=a3/(sigma*(rep-burn))
as(4)=inte1/((sigma**2)*(rep-burn))
as(5)=sum(intdat2)/(rep-burn)

return
end subroutine

! coeff polynomial
subroutine update(as,sig,n,mu,sigma,tau)
implicit none
INTEGER n
REAL as(5),mu,sigma,tau,sig

mu=as(1)/as(2)
sigma=as(3)/as(4)
tau=sqrt((as(5)*as(4)-as(3)**2)/(n*as(4)))

return
end subroutine

! calculate the mean
subroutine mean(dat,numdat,average)
implicit none
INTEGER i,numdat
REAL dat(0:numdat-1),average

average=0.0
do i=0,numdat-1
        average=dat(i)+average
enddo
average=average/(1.0*numdat)
```

```
return
end subroutine

! Metropolis-Hasting
subroutine accept(del,acc,tau,p1,numdat,ndat,intdat,dat1,dat2,&
                  &,path,normals)
INTEGER acc,numdat,i,p1,ndat
REAL intdat(ndat),dat1(numdat),dat2(numdat),num,dem,prob_acc,unif,aux
REAL rect1,rect2,fun,mean1(numdat-1),mean2(numdat-1),tau,path(numdat)
REAL nor1,nor2,normal1(numdat-1),normal2(numdat-1),del,normals(numdat-1)

prob_acc=1.0
do i=1,numdat-1
        call integrate(del,dat1(i),dat1(i+1),tau,rect1,nor1,fun)
        mean1(i)=rect1
        normal1(i)=nor1
        call integrate(del,dat2(i),dat2(i+1),tau,rect2,nor2,fun)
        mean2(i)=rect2
        normal2(i)=nor2
        num=exp((-(intdat(i+p1)-mean2(i))**2)/(2*(tau**2)))
        dem=exp((-(intdat(i+p1)-mean1(i))**2)/(2*(tau**2)))
        prob_acc=(num/dem)*prob_acc
enddo

prob_acc=min(prob_acc,1.0)
call random_number(unif)

if(unif<prob_acc)then
        path=dat2
        normals=normal2
        acc=acc+1
else
        path=dat1
        normals=normal1

endif

return
end subroutine

! choose the size of block
subroutine sel_block(l,b,mc,ms)
implicit none
integer mc,st,b,R,ms(1:b),poi,i
real l

mc=1
st=1
ms=0.0
```

```
call genpoi(l,poi)
ms(mc)=1+poi
if(ms(mc)>=b)then
        ms(mc)=b
else
        mc=2
        do while(st<2)
                call genpoi(l,poi)
                ms(mc)=1+poi

                if(sum(ms(1:mc))>=b)then
                        ms(mc)=b-sum(ms(1:mc-1))
                        st=2
                else
                        mc=mc+1
                endif
        enddo
endif

do i=mc+1,b
        ms(i)=0
enddo

return
end subroutine

! Subpaths
subroutine pps(a,b,v,p1,p2)
implicit none
INTEGER p1,p2,a,b,v(1:b)

if(a==1)then
        p1=0
        p2=v(a)
else
        p1=sum(v(1:a-1))
        p2=sum(v(1:a))
endif

return
end subroutine

! Direct path
subroutine ou1(ini,b,dri,de,sig,B1)
implicit none
integer b,i
real ini,nor(b),B1(b),dri,sig,de,sd
real gennor
external gennor
```

```
sd=sqrt(de)
do i=1,b
        nor(i)=gennor(0,sd)
enddo

B1(1)=ini*(1-dri*de)+sig*nor(1)
i=2
do while(i<=b)
        B1(i)=B1(i-1)*(1-dri*de)+sig*nor(i)
        i=i+1
enddo

return
end subroutine

! Inverse path
subroutine ou2(ini,b,dri,de,sig,B2) !(datup(p2),dif,&
               &,mu(1),delta,sigma(1),B2)
implicit none

integer b,c,i
real ini,nor(b),B1(b),dri,sig,de,B2(b),sd
real gennor
external gennor

i=1
do while(i<=b)
sd=sqrt(de)
        nor(i)=gennor(0,sd)
i=i+1
enddo

B1(1)=ini*(1-dri*de)+sig*nor(1)

i=2
do while(i<=b)
        B1(i)=B1(i-1)*(1-dri*de)+sig*nor(i)
        i=i+1
enddo
do i=1,b
        B2(i)=B1(b+1-i)
enddo

return
end subroutine


! Bridge
subroutine bri(inf,inb,a,dri,de,sig,Bridge)
implicit none
```

```
integer a,cr,i
real dri,de,sig,inf,inb
real X(a),Y(a),Bridge(a+2),brid(a)

call ou1(inf,a,dri,de,sig,X)
cr=0
do while(cr<1)

        call ou2(inb,a,dri,de,sig,Y)
        i=1
        do while(i<=a)
!     first case

                if (X(1)<Y(1))then
                        cr=0
                        if(X(i)>Y(i))then
                                cr=i
                                i=a+1
                        endif
                endif

!   second case
                if (X(1)>Y(1))then
                        cr=0

                        if(X(i) <Y(i))then
                        cr=i
                        i=a+1
                        endif
                endif
                i=i+1
        enddo
enddo

Bridge(1)=inf
Bridge(a+2)=inb
do i=1,cr-1
        brid(i)=X(i)
enddo
do i=cr,a
        brid(i)=Y(i)
enddo

do i=2,a+1
Bridge(i)=brid(i-1)
enddo

return
end subroutine
```

```fortran
! The initial
subroutine initial(inf,sup,num,numdat,dat,current)
implicit none
INTEGER inf,sup,num,i,numdat
REAL dat(0:numdat),current(num+1)

current(1)=dat(inf)
current(num+1)=dat(sup)
do i=2,num
        current(i)=dat(inf+i-1)
enddo

return
end subroutine

! Update data
subroutine up_data(numdat,p1,p2,path,updata)
implicit none
INTEGER p2,p1,numdat,i
REAL path(p2-p1+1),updata(0:numdat)

do i=p1,p2
        updata(i)=path(i-p1+1)
enddo

return
end subroutine

! Update normals
subroutine up_normal(numdat,p1,p2,parnor,normals)
implicit none
INTEGER  p2,p1,numdat,i
REAL parnor(p2-p1),normals(numdat)

do i=p1+1,p2
        normals(i)=parnor(i-p1)
enddo

return
end subroutine

! Poisson
subroutine genpoi(lambda,poisson)
implicit none
REAL lambda,unif1,unif2,ex
INTEGER poisson

poisson=0
unif1=1.0
ex=exp(-lambda)
```

```
do while(unif1>=ex)
        call random_number(unif2)
        unif1=unif1*unif2
        poisson=poisson+1
enddo

return
end subroutine


! Complete the bridge to first block
subroutine com_bri1(point,dif,bri,bridge)!(datup(p2),dif,B2,Bridge)
implicit none
INTEGER i,dif
REAL point,bri(dif),bridge(dif+1)

do i=1,dif
        bridge(i)=bri(i)
enddo
bridge(dif+1)=point

return
end subroutine

! Complete the bridge to last block
subroutine com_bri2(point,dif,bri,bridge)
implicit none
INTEGER i,dif
REAL point,bri(dif),bridge(dif+1)

do i=2,dif+1
        bridge(i)=bri(i-1)
enddo
bridge(1)=point

return
end subroutine

! Hessian
subroutine hessian(mu,sigma,coef,hes)
implicit none
REAL mu,sigma,val,hes,coef(5)

if(coef(3)<0.0)then
        coef(3)=-coef(3)
endif
hes=(mu**2)*(coef(4)**2)-mu*coef(3)*coef(4)+coef(3)**2
val=-2*coef(4)/(sigma**2)

return
```

```
end subroutine

! Function
subroutine eva_fun(coef,mu,sig,val)
implicit none
REAL coef(5),mu,sig,val

val=(coef(3)*mu)/(sig**2)-(coef(4)*(mu**2))/(sig**2)+coef(5)*mu

return
end subroutine
! Exact
subroutine ou(a1,a2,a,d,theta,sig,Y)!(dat_tran(num_col),dat_tran(num_col+1
implicit none
INTEGER a,b,i,c
REAL X(a+2),Y(a+2),W(a+1),sd,theta,sig,d,gennor,a1,a2,bri(a)
external gennor

d=1.0/((a+1)*1.0)
X(1)=a1

sd=(sig*sqrt((1-exp(-2*theta*d)))/(2*theta))

do i=1,a+1
        W(i)=gennor(0,sd)
enddo
do i=2,a+2
        X(i)=exp(-theta*d)*X(i-1)+W(i-1)
enddo

do i=1,a+2
        Y(i)=X(i)+(a2-X(a+2))*(exp(theta*((i-1)*d))-exp(-theta*((i-1)*d)))/
enddo

do i=1,a
        bri(i)=Y(i+1)
enddo

return
end subroutine
```

Bibliography

A. Svensson, K. K. A.; Clausen, M. B. H. B.; Dahl-Jensen, D.; Davies, S. M.; Johnsen, S. J.; Muscheler, R.; Parrenin, F.; Rasmussen, S. O.; Röthlisberger, R.; Seierstad, I.; Steffensen, J. P. & Vinther, B. M. (2008). "A 60 000 year Greenland stratigraphic ice core chronology". *Climate of the Past*, 4:47–57.

Aït-Sahalia, Y. (1996). "Testing Continuous-Time Models of the Spot Interest Rate". *Review of Financial Studies*, 9:385–426.

Aït-Sahalia, Y. & Mykland., P. A. (2004). "Estimators of diffusions with randomly space discrete observations: A general Theory". *Ann. Statist.*, 32:2186–2222.

Andersen, T. & Bollerslev, T. (1998). "Answering the skeptics: yes, standard volatility models do provide accurate forecasts". *Int. Econ. Rev.*, 39:885–905.

Barndorff-Nielsen, O. E. & Shephard, N. (2002). "Econometric analysis of realized volatility and its use in estimating stochastic volatility models". *J. R. Statist. Soc B*, 64:253–280.

Beskos, A.; Papaspiliopoulos, O. & Roberts, G. O. (2006). "Retrospective exact simulation of diffusion sample paths with applications". *Bernoulli*, 12:1077–1098.

Beskos, A.; Papaspiliopoulos, O. & Roberts, G. O. (2007). "A new factorization of diffusion measure with view towards simulation". To be published.

Beskos, A.; Papaspiliopoulos, O.; Roberts, G. O. & Fearnhead, P. (2006). "Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes". *J. R. Statist.Soc. B*, 68:333–382.

Billingsley, P. (1961). *Statistical Inference for Markov Processes.* University of Chicago Press, Chicago.

Black, F. & Scholes, M. (1973). "The pricing of options and corporate liabilities". *Journal of Political Economy*, 81:631âĂŞ659.

Bladt, M. & Sørensen, M. (2005). "Statistical inference for discretely observed Markov jump processes". *J. R. Statist. Soc.*, 67:395–410.

Bladt, M. & Sørensen, M. (2006). "Efficient estimation of transition rates between credit ratings from observations at discrete time points". *Center for Analytical Finance. University of Aarhus. Working paper*, 218.

Bladt, M. & Sørensen, M. (2009). "Simple simulation of diffusion bridges with application to likelihood inference for diffusions". Working paper, Dept. Math. Sciences, Univ. of Copenhagen.

Bollerslev, T. & Zhou, H. (2002). "Estimating stochastic volatility diffusion using conditional moments of integrated volatility". *Journal of Econometrics*, 109:33–65.

Broecker, W. (1994). "Massive iceberg dischargers as triggers for global climate change". *Nature*, 372:421–424.

Chacko, G. & Viceira, L. (2003). "Spectral GMM Estimation of Continous-Time Processes". *Journal of Econometrics*, 116:259–292.

Chib, S. & Greenberg, E. (1995). "Understanding the Metropolis-Hastings algorithm". *The American Statistician*, 49:327–335.

Chib, S.; Pitt, M. K. & Shephard, N. (2006). "Likelihood based inference for diffusion driven state space models". Working paper.

Comte, F.; Genon-Catalot, V. & Rozenholc, Y. (2009). "Nonparametric adaptive estimation for integrated diffusions". *Stochastic processes and their applications*, 119:811–834.

Dacunha-Castelle, D. & Florens-Zmirou, D. (1986). "Estimation of the coefficients of a diffusion from discrete observations". *Stochastics*, 19:263âĂŞ284.

Dehay, D. & Yao, J. (2006). "On likelihood estimation for a discretely observed jump process". *C.R. Acad. Sci. Paris*, 342:341–344.

Dempster, A. P.; Laird, N. M. & Rubin, D. B. (1977). "Maximum likelihood from incomplete data via the EM algorithm (with discussion)". *J. Roy. Statist. Soc. B*, 39:1 – 38.

Ditlevsen, P. D.; Ditlevsen, S. & Andersen, K. K. (2002). "The fast climate fluctuations during the stadial and interstadial climate states". *Annals of Glaciology*, 35:457–462.

Ditlevsen, S. & Sørensen, M. (2004). "Inference for observations of integrated diffusion processes". *Scand. J. Statist.*, 31:417–429.

Doob, J. (1953). *Stochastic Processes.* Wiley, New York.

Durham, G. B. & Gallant, A. R. (2002). "Numerical techniques for maximum likelihood estimation of continuous-time diffusion processes". *J. Business & Econom. Statist.*, 20:297–338.

Elerian, O. (1998). "A note on the existence of a closed form conditional density for the Milstein scheme". *Working Paper, Nuffield College, Oxford University.*

Elerian, O.; Chib, S. & Shephard, N. (2001). “Likelihood inference for discretely observed non-linear diffusions”. *Econometrica*, 69:959–993.

Eraker, B. (2001). “MCMC Analysis of Diffusion Models with Application to Finance”. *Journal of Business and Economic Statistics*, 19:177–191.

Forman, J. L. & Sørensen, M. (2008). “The Pearson diffusions: A class of statistically tractable diffusion processes”. *Scand. J. Statist.*, 35:438–465.

Gallant, A. R. & Tauchen, G. (1996). “Which Moments to Match”. *Econometric Theory*, 12:657–681.

Gilks, W. R.; Richardson, S. & Spiegelhalter, D. J. (1996). *Markov Chain Monte Carlo in Practice.* Chapman & Hall, London.

Gloter, A. (2006). “Parameter estimation for a discretely observed integrated diffusion process”. *Scand. J. Statist.*, 33:83–104.

Grimmett, G. & Stirzaker, D. (2001). *Probability and Random Process.* Oxford University Press Inc., New York.

Hansen, L. & J.Scheinkman (1995). “Back to the Future: Generating Moment Implications for Continous-Time Markov Processes”. *Econometrica*, 63:767–804.

Ito, K. & McKean, H. (1965). *Diffusion Processes and their Sample Paths.* Springer.

Jacobsen, M. (1982). “Statistical analysis of countin@ARTICLEKes, AUTHOR =”M. Kessler”, TITLE =”Estimation of an ergodic diffusion from discrete observations”, JOURNAL =”Scand. J. Stat.”, VOLUME =24, PAGES =”211âĂŞ229”, YEAR =1997g processes”. *Lect. Notes Statist.*, 12.

Jensen, B. & Poulsen., R. (2002). “Transition Densities of Diffusion Processes: Numerical Comparison of Approximation Techniques”. *Journal of Derivatives*, 9:18–32.

Karatzas, I. & Shreve, S. E. (1991). *Brownian Motion and Stochastic Calculus.* Springer, New York.

Karlin, S. & Taylor, H. M. (1975). *A First Course in Stochastic Processes.* Academic Press, Orlando.

Karlin, S. & Taylor, H. M. (1981). *A Second Course in Stochastic Processes.* Academic Press, Orlando.

Keiding, N. (1974). “Estimation in the birth process”. *Biometrika*, 61:71–80.

Keiding, N. (1975). “Maximum likelihood estimation in the birth-and-death process”. *Ann. Statist.*, 3:363–372.

Kessler, M. (1997). “Estimation of an ergodic diffusion from discrete observations”. *Scand. J. Stat.*, 24:211âĂŞ229.

Kuchler, U. & Sørensen, M. (1999). *Exponential Families of Stochastic Processes.* Springer, New York.

Øksendal, B. (1998). *Stochastic Differential Equations.* Springer, New York.

P. Fitzsimmons, J. P. & Yor, M. (1992). "Markovian bridges: constuction, palm interpretation and splicing.". *Seminar on Stochastic Processes.*, pages 101–134.

Pedersen, A. R. (1995). "A new approach to maximum likelihood estimation for stochastic differential equations based on discrete observations". *Scand. J. Statist.*, 22:55–71.

R. Israel, J. R. & Wei, J. (2001). "Finding generators for Markov chains via empirical transition matrices, with applications to credit ratings". *Mathematical Finance*, 11:245–265.

R. Jarrow, D. L. & Turnbull., S. M. (1997). "A Markov model for the term structure of credit risk spreads". *Rev. Finan. Stud.*, 10:481–523.

Roberts, G. O. & Stramer, O. (2001). "On inference for partially observed nonlinear diffusion models using Metropolis-Hastings algorithms". *Biometrika*, 88:603–621.

Sørensen, M. (1997). "Estimating functions for discretely observed diffusions: a review". In Basawa, I. V.; Godambe, V. P. & Taylor, R. L., editors, *Selected Proceedings of the Symposium on Estimating Functions*, pages 305–325. Hayward: Institute of Mathematical Statistics. IMS Lecture Notes – Monograph Series, Vol. 32.

Sørensen, M. (2000). "Prediction-Based Estimating Functions". *Econometrics Journal*, 3:123–147.

Sørensen, M. (2010). "Estimating functions for diffusion-type processes". In Kessler, M.; Lindner, A. & Sørensen, M., editors, *Statistical Methods for Stochastic Differential Equations.* Chapman & Hall. Forthcoming.

T. Andersen, T. Bollerslev, F. X. D. & Labys, P. (2001). "The distribution of realized exchange rate volatility". *J. Amer. Statist. Assoc.*, 96:42–55.

Yoshida, N. (1992). "Estimation for diffusion process from discrete observation". *J. Multivariate Anal.*, 41:220–242.