Sim.DiffProc: A Package for Simulation of Diffusion Processes in R
Kamal Boukhetala, Arsalane Guidoum

To cite this version:
Kamal Boukhetala, Arsalane Guidoum. Sim.DiffProc: A Package for Simulation of Diffusion Processes in R. 2011. <hal-00629841>

HAL Id: hal-00629841
https://hal.archives-ouvertes.fr/hal-00629841
Submitted on 6 Oct 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Sim.DiffProc: A Package for Simulation of Diffusion Processes in R

Kamal Boukhetala
Department of Probabilities & Statistics
University of Science and Technology Houari Boumediene
E-mail: kboukhetala@usthb.dz

Arsalane Guidoum
Department of Probabilities & Statistics
University of Science and Technology Houari Boumediene
E-mail: starsalane@gmail.com


Abstract

The Sim.DiffProc package provides a simulation of diffusion processes and the differences methods of simulation of solutions for stochastic differential equations (SDEs) of the Itô’s type, in financial and actuarial modeling and other areas of applications, for example the stochastic modeling and simulation of pollutant dispersion in shallow water using the attractive center, and the model of two diffusions in attraction, which can modeling the behavior of two insects, one attracts the other. The simulation of the processes of diffusion, through stochastic differential equations to allow simulated a random variable \( \tau_c \) "first passage time" of the particle through a sphere of radius \( c \), two methods are used in the estimation problem of the probability density function of a random variable \( \tau_c \); the histograms and the kernel methods. The R package Sim.DiffProc is introduced, providing a simulation and estimation for the stationary distribution of the stochastic process that describes the equilibrium of some dynamics.

Key words: attractive model, diffusion process, simulations, stochastic differential equation, stochastic modeling, R language.
1 Introduction

Stochastic differential equations (SDEs) are a natural choice to model the time evolution of dynamic systems. These equations have a variety of applications in many disciplines and can be a powerful tool for the modeling and the description of many phenomena. Examples of these applications are physics, astronomy, economics, financial mathematics, geology, genetic analysis, ecology, neurology, biology, biomedical sciences, epidemiology, political analysis and social processes, and many other fields of science and engineering. The stochastic differential equations, with slight notational variations, are standard in many books with applications in different fields see [23, 15, 1, 25, 14] to name only a few.

The first is to recall the theory and implement methods for the simulation of paths of diffusion processes \( \{X_t, t \geq 0\} \) solutions to stochastic differential equations (SDEs), the sense that only SDEs with Gaussian noise \( B_t \) are considered i.e., processes for which the writing

\[
\frac{dX_t}{dt} = \mu(\theta, t, X_t) + \sigma(\vartheta, t, X_t)B_t
\]

With the gaussian noise \( B_t \) is the formal derivative of the standard Wiener process \( W_t \) (i.e., \( B_t = \frac{dW_t}{dt} \)), the write formally for the process \( \{X_t, t \geq 0\} \) is

\[
dx_t = \mu(\theta, t, X_t)dt + \sigma(\vartheta, t, X_t)dW_t,
\]

with some initial condition \( X_0 \).


This work is organized as follows. Section 2 gives a simulation for some very popular stochastic differential equations models (trajectory of the Brownian motion, Ornstein-Uhlenbeck process), and present the different numerical methods (schemes) of simulation of SDEs (See [21]), simulation the random variable \( X_t \) at time \( t \) by a simulated diffusion processes used the numerical methods and estimate the stationary distribution for \( X_t \) by histograms and kernel methods. Section 3 the diffusion processes are used to modeling the behavior of the dispersal phenomenon and dynamic models for insects orientation, two important models can be indicated. The graphical user interface (GUI) for Sim.DiffProc package are provided in Section 4.

2 Diffusion processes

We consider the model as the parametric Itô stochastic differential equation

\[
dx_t = \mu(\theta, t, X_t)dt + \sigma(\vartheta, t, X_t)dW_t, \quad t \geq 0, X_0 = \zeta
\]
where \( \{ W_t, t \geq 0 \} \) is a standard Wiener process, \( \mu : \Theta \times [0, T] \times \mathbb{R} \to \mathbb{R} \), called the drift coefficient, and \( \sigma : \Xi \times [0, T] \times \mathbb{R} \to \mathbb{R}^+ \), called the diffusion coefficient, are known functions except the unknown parameters \( \theta \) and \( \vartheta \), \( \Theta \subset \mathbb{R} \), \( \Xi \subset \mathbb{R} \) and \( \mathbb{E}(\zeta^2) < \infty \), the estimation problems for the parameters \( \theta \) and \( \vartheta \) can be seen in some papers \([2, 4, 11]\). The drift coefficient is also called the trend coefficient or damping coefficient or translation coefficient. The diffusion coefficient is also called volatility coefficient. Under global Lipschitz and the linear growth conditions on the coefficients \( \mu \) and \( \sigma \), there exists a unique strong solution of the above Itô SDEs, called the diffusion process or simply a diffusion, which is a continuous strong Markov semimartingale. The drift and the diffusion coefficients are respectively the instantaneous mean and instantaneous standard deviation of the process. All over the text, the stochastic differential equation (1), they are supposed to be measurable.

**Assumption 1:** (Global Lipschitz) For all \( x, y \in \mathbb{R} \) and \( t \in [0, T] \), there exists a constant \( K < \infty \) such that

\[
|\mu(t, x) - \mu(t, y)| + |\sigma(t, x) - \sigma(t, y)| < K|x - y|.
\]

**Assumption 2:** (Linear growth) For all \( x, y \in \mathbb{R} \) and \( t \in [0, T] \), there exists a constant \( C < \infty \) such that

\[
|\mu(t, x)| + |\sigma(t, x)| < C(1 + |x|),
\]

The linear growth condition controls the behaviour of the solution so that \( X_t \) does not explode in a finite time.

### 2.1 Simulation of Stochastic Differential Equations SDEs

In this section, we present some of the well-known and widely used diffusion process solutions to the stochastic differential equation. There are two main objectives in the simulation of the trajectory of a process solution of a stochastic differential equation, either interest is in the whole trajectory or in the expected value of some functional of the diffusion process (moments, distributions, etc) which usually are not available in explicit analytical form. Numerical Methods are usually based on discrete approximations of the continuous solution to a stochastic differential equation. The following examples use the \texttt{Sim.DiffProc} package for simulation some diffusion process.

To install the \texttt{Sim.DiffProc} package on your version of \texttt{R}, type the following line in the \texttt{R} console.

\[
R> \text{install.packages("Sim.DiffProc")}
\]

If you don’t have enough privileges to install software on your machine or account, you will need the help of your system administrator. Once the package has been installed, you can actually use it by loading the code with

\[
R> \text{library(Sim.DiffProc)}
\]

A short list of help topics, corresponding to most of the commands in the package, is available by typing \[
R> \text{library(help = "Sim.DiffProc")}
\]
2.1.1 Simulation of the trajectory of the Brownian motion

Simulation of the trajectory of the Brownian motion: The very basic ingredient of a model describing stochastic evolution is the so-called Brownian motion or Wiener process. There are several alternative ways to characterize and define the Wiener process $W = \{W_t, t \geq 0\}$, and one is the following: it is a Gaussian process with continuous paths and with independent increments such that $W(0) = 0$ with probability 1, $E(W_t) = 0$, and $\text{var}(W_t - W_s) = t - s$ for all $0 \leq s < t$. In practice, what is relevant for our purposes is that $W(t) - W(s) \sim N(0, t - s)$. Given a fixed time increment $\Delta t > 0$, one can easily simulate a trajectory of the Wiener process in the time interval $[t_0, T]$. Indeed, for $W_{\Delta t}$ it holds true that

$$W(\Delta t) = W(\Delta t) - W(0) \sim N(0, \Delta t) \sim \sqrt{\Delta t} \cdot N(0, 1)$$

and the same is also true for any other increment $W(t + \Delta t) - W(t)$; i.e.,

$$W(t + \Delta t) - W(t) \sim N(0, \Delta t) \sim \sqrt{\Delta t} \cdot N(0, 1)$$

For $i = 0, 1, \ldots, N - 1$, with initial deterministic value $x_0$. Usually the time increment $\Delta t = t_{i+1} - t_i$ is taken to be constant (i.e., $\Delta t = (T - t_0)/N$)

R> BMN(N = 10000, t0 = 0, T = 1, C = 1)
R> BMN2D(N = 10000, t0 = 0, T = 1, x0 = 0, y0 = 0, Sigma = 1)

Figure 1: Simulation examples to illustrate the trajectory of the Brownian motion used the function BMN, and 2-dimensional Brownian motion used BMN2D.
2.1.2 Simulation a Ornstein-Uhlenbeck or Vasicek process

The Ornstein-Uhlenbeck or Vasicek process is the unique solution to the following stochastic differential equation

\[ dX_t = r(\theta - X_t)dt + \sigma dW_t, \quad X_0 = x_0, \]  

(2)

where \( \sigma \) is interpreted as the volatility, \( \theta \) is the long-run equilibrium value of the process, and \( r \) is the speed of reversion. As an application of the Itô lemma, we can show the explicit solution of (2) by choosing \( f(t, x) = xe^{rt} \), we obtain

\[ X_t = \theta + (x_0 - \theta)e^{-rt} + \sigma \int_{t_0}^{t} e^{-r(t-s)} dW_s. \]

It can be seen that for \( \theta = 0 \) the trajectory of \( X_t \) is essentially a negative exponential perturbed by the stochastic integral. One way of simulating trajectories of the Ornstein-Uhlenbeck process is indeed via the simulation of the stochastic integral.

\[ \text{R> HWV}(N = 1000, t0 = 0, T = 10, x0 = -5, \theta = 0, r = 1, \sigma = 0.5) \]

\[ \text{R> HWVF}(N = 1000, M = 100, t0 = 0, T = 10, x0 = -5, \theta = 0, r = 1, \sigma = 0.5) \]

Figure 2: Simulated path of the Ornstein-Uhlenbeck process \( dX_t = -X_t dt + 0.5dW_t \) used the function \text{HWV}, and 100 trajectories of the \( dX_t = -X_t dt + 0.5dW_t \) used \text{HWVF}.

We sketch some very popular SDEs models and example of simulations:
### Models and Expressions

<table>
<thead>
<tr>
<th>Models</th>
<th>Expressions</th>
<th>R code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arithmetic Brownian Motion</td>
<td>$dX_t = \theta dt + \sigma dW_t$</td>
<td>ABM</td>
</tr>
<tr>
<td>Geometric Brownian Motion</td>
<td>$dX_t = \theta X_t dt + \sigma X_t dW_t$</td>
<td>GBM</td>
</tr>
<tr>
<td>Cox-Ingersoll-Ross</td>
<td>$dX_t = (r - \theta X_t) dt + \sigma \sqrt{X_t} dW_t$</td>
<td>CIR</td>
</tr>
<tr>
<td>Constant Elasticity of Variance</td>
<td>$dX_t = \mu X_t dt + \sigma X_t dW_t$</td>
<td>CEV</td>
</tr>
<tr>
<td>Radial Ornstein-Uhlenbeck</td>
<td>$dX_t = (\theta X_t^{-1} - X_t) dt + \sigma dW_t$</td>
<td>ROU</td>
</tr>
<tr>
<td>Chan-Karloyi-Logstaff-Sanders</td>
<td>$dX_t = (r + \theta X_t) dt + \sigma X_t dW_t$</td>
<td>CKLS</td>
</tr>
<tr>
<td>Hyperbolic Diffusion</td>
<td>$dX_t = -\theta X_t(1 + X_t^2)^{-0.5} dt + \sigma dW_t$</td>
<td>Hyproc</td>
</tr>
<tr>
<td>Jacobi diffusion</td>
<td>$dX_t = -\theta(X_t - 0.5) dt + \sqrt{\theta X_t(1 - X_t)} dW_t$</td>
<td>JDP</td>
</tr>
</tbody>
</table>

#### 2.1.3 Example of use

In particular, it is possible to generate $M$ independent trajectories of the same process with one single call of the function by just specifying a value for $M$

```r
R> ABM(N = 1000, t0 = 0, T = 1, x0 = 0, theta = 3, sigma = 2)
R> GBM(N = 1000, T = 1, t0 = 0, x0 = 1, theta = 2, sigma = 0.5)
R> CIR(N = 1000, M = 1, t0 = 0, T = 1, x0 = 1, theta = 0.2, r = 1, sigma = 0.5)
R> CEV(N = 1000, M = 1, t0 = 0, T = 1, x0 = 1, mu = 0.3, sigma = 2, gamma = 1.2)
R> ROU(N = 1000, M = 1, T = 1, t0 = 0, x0 = 1, theta = 0.05)
R> CKLS(N = 1000, M = 1, T = 1, t0 = 0, x0 = 1, r = 0.3, theta = 0.01, sigma = 0.1, gamma = 0.2)
R> Hyproc(N = 1000, M = 1, T = 100, t0 = 0, x0 = 3, theta = 2)
R> JDP(N = 1000, M = 1, T = 100, t0 = 0, x0 = 0, theta = 0.05)
```

#### 2.2 Numerical Methods for SDEs

The idea is the following given an Itô process $\{X_t, 0 \leq t < T\}$ solution of the stochastic differential equation

$$dX_t = f(X_t) dt + g(X_t) dW_t, \quad X_0 = x_0,$$

where $W_t$ represents the standard Wiener process and initial value $x_0$ is a fixed value. In many literatures, whose partial list can be seen in the references of the present paper, numerical schemes for SDE were proposed, which recursively compute sample paths (trajectories) of solution $X_t$ at step-points. Numerical experiments for these schemes can be seen in some papers [16, 20, 24].

In the following, we present numerical schemes. They adopt an equidistant discretization of the time interval $[t_0, T]$ with stepsize

$$\Delta t = \frac{(T - t_0)}{N}, \quad \text{for fixed natural number} \quad N.$$
Furthermore,
\[ t_n = n\Delta t, \quad n \in \{1, 2, \ldots, N\} \]
denotes the \( n \)-th step-point. We abbreviate \( X_n = X_{t_n} \).

The following three random variables will be used in the \((n + 1)\) time step of the schemes:

\[
\begin{align*}
\Delta W_n &= W_{t_{n+1}} - W_{t_n}, \\
\Delta Z_n &= \int_{t_n}^{t_{n+1}} \int_{t_n}^{s} dW_r ds, \\
\Delta Z_n &= \int_{t_n}^{t_{n+1}} \int_{t_n}^{s} dr dW_s.
\end{align*}
\]

They are obtained as sample values of normal random variables using the transformation \[21\]
\[
\begin{align*}
\Delta W_n &= \xi_{n,1}(\Delta t)^{1/2}, \\
\Delta Z_n &= \frac{1}{2} \left( \xi_{n,1} + \frac{\xi_{n,2}}{\sqrt{3}} \right) (\Delta t)^{3/2}, \\
\Delta Z_n &= \frac{1}{2} \left( \xi_{n,1} - \frac{\xi_{n,2}}{\sqrt{3}} \right) (\Delta t)^{3/2}.
\end{align*}
\]

and, together with them, we further use \( \Delta \tilde{W}_n = \xi_{n,2}(\Delta t)^{1/2} \), where \( \xi_{n,1}, \xi_{n,2} \) are mutually independent \( N(0, 1) \) random variables.

### 2.2.1 Numerical schemes

**Euler-Maruyama scheme** (Maruyama 1955):
\[
X_{n+1} = X_n + f_n \Delta t + g_n \Delta W_n
\]

**Milstein scheme** (Milstein 1974):
\[
X_{n+1} = X_n + f_n \Delta t + g_n \Delta W_n + \frac{1}{2} g_n^r g_n^2 ((\Delta W_n)^2 - \Delta t)
\]

**Milstein Second scheme** (Milstein 1974):
\[
X_{n+1} = X_n + f_n \Delta t + g_n \Delta W_n + \frac{1}{2} g_n^r g_n^2 ((\Delta W_n)^2 - \Delta t) + f_n^r g_n \Delta Z_n \\
+ \left( g_n^r f_n + \frac{1}{2} g_n^r g_n^2 \right) \Delta Z_n + \frac{1}{6} (g_n^r g_n^2 + g_n^r g_n^2) ((\Delta W_n)^3 - 3\Delta t \Delta W_n)
\]
Taylor scheme [20]:
\[
X_{n+1} = X_n + f_n \Delta t + g_n \Delta W_n + \frac{1}{2} g_n' g_n ((\Delta W_n)^2 - \Delta t) + f_n' g_n \Delta Z_n \\
\hspace{1cm} + \left( g_n' f_n + \frac{1}{2} g_n'' g_n^2 \right) \Delta Z_n + \frac{1}{6} \left( g_n'' g_n + 2g_n'' (\Delta W_n)^3 - 3 \Delta t \Delta W_n \right) \\
\hspace{1cm} + \frac{1}{2} \left( f_n'' f_n + \frac{1}{2} f_n'' g_n^2 \right) (\Delta t)^2
\]
(7)

Heun scheme (McShane 1974):
\[
X_{n+1} = X_n + \frac{1}{2} [F_1 + F_2] \Delta t + \frac{1}{2} [G_1 + G_2] \Delta W_n,
\]
(8)
where
\[
F_1 = F(X_n), \hspace{1cm} G_1 = g(X_n), \\
F_2 = F(X_n + F_1 \Delta t + G_1 \Delta W_n), \hspace{1cm} G_2 = g(X_n + F_1 \Delta t + G_1 \Delta W_n), \\
F_x = \left[ f - \frac{1}{2} g' g \right](x).
\]

Improved 3-stage Runge-Kutta scheme [24]:
\[
X_{n+1} = X_n + \frac{1}{4} [F_1 + 3F_3] \Delta t + \frac{1}{4} [G_1 + 3G_3] \Delta W_n \\
\hspace{1cm} + \frac{1}{2\sqrt{3}} \left[ f_n' g_n - g_n f_n' - \frac{1}{2} \frac{g_n'' g_n^2}{g_n} \right] \Delta t \Delta \tilde{W}_n,
\]
(9)
where
\[
F_1 = F(X_n), \hspace{1cm} G_1 = g(X_n), \\
F_2 = F \left( X_n + \frac{1}{3} F_1 \Delta t + \frac{1}{3} G_1 \Delta W_n \right), \hspace{1cm} G_2 = g \left( X_n + \frac{1}{3} F_1 \Delta t + \frac{1}{3} G_1 \Delta W_n \right), \\
F_3 = F \left( X_n + \frac{2}{3} F_2 \Delta t + \frac{2}{3} G_2 \Delta W_n \right), \hspace{1cm} G_3 = g \left( X_n + \frac{2}{3} F_2 \Delta t + \frac{2}{3} G_2 \Delta W_n \right), \\
F_x = \left[ f - \frac{1}{2} g' g \right](x).
\]

2.2.2 Example of use

The following examples for different methods of simulation of SDEs use the \texttt{snssde} function

\texttt{snssde(N, M, T = 1, t0, x0, Dt, drift, diffusion, Output = FALSE, \\
Methods = c("SchEuler", "SchMilstein", "SchMilsteinS", 
\hspace{1cm} "SchTaylor", "SchHeun", "SchRK3"))}
with: SchEuler \(4\), SchMilstein \(5\), SchMilsteinS \(6\), SchTaylor \(7\), SchHeun \(8\), SchRK3 \(9\).

Consider for example the stochastic process \(\{X_t, t \geq 0\}\) solution of

\[
dX_t = (\mu t X_t - X_t^3)dt + \sigma dW_t, \quad X_0 = x_0,
\]

For this process, \(f(X_t) = (\mu t X_t - X_t^3)\) and \(g(X_t) = \sigma\). Suppose we fix an initial value \(X_0 = 0\) and the set of parameters \(\mu = 0.03\) and \(\sigma = 0.1\). The following algorithm can be used to simulate one trajectory of the process \(X_t\) using the Euler algorithm (figure 3), and 100 trajectories used Milstein scheme (figure 4).

R> f_x <- expression( 0.03 * t * x - x^3 )
R> g_x <- expression( 0.1 )
R> snssde(N = 1000, M = 1, t0 = 0, x0 = 0, Dt = 0.1, drift = f_x,
    diffusion = g_x, Methods = "SchEuler")
R> output <- data.frame(time,X)
R> output

Figure 3: Simulated one trajectory of the process \(dX_t = (0.03t X_t - X_t^3)dt + 0.1dW_t\) used Euler algorithm.

Figure 4: Simulated 100 trajectories of the process \(dX_t = (0.03t X_t - X_t^3)dt + 0.1dW_t\) used Milstein scheme.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>0.000000e+00</td>
</tr>
<tr>
<td>2</td>
<td>0.1</td>
<td>3.162278e-02</td>
</tr>
<tr>
<td>3</td>
<td>0.2</td>
<td>6.324555e-06</td>
</tr>
</tbody>
</table>
2.3 Stationary distribution in the SDEs models

The stationary distribution of the stochastic process that describes the equilibrium of some dynamics. For these disciplines, the interest is in the shape of the stationary distributions and the statistical indexes related to them (mean, mode, etc.). The stochastic differential equations usually have a linear drift of the form \( f(x) = r(\theta - x) \) with \( r > 0 \). For this reason, the models are called linear feedback models. The diffusion coefficient \( g(x) \) may be constant, linearly depending on \( x \), or of polynomial type, leading, respectively, to Gaussian, Gamma, or Beta stationary distributions.

2.3.1 Type N model (Normal Distribution)

For this model, we have:

- Drift coefficient: \( f(X_t) = r(\theta - X_t), r > 0 \).
- Diffusion coefficient: \( g(X_t) = \sqrt{2\sigma}, \sigma > 0 \).
- Sde: \( dX_t = r(\theta - X_t)dt + \sqrt{\sigma}dW_t \).
- Stationary density: \( \pi(x) = \frac{1}{\sqrt{2\pi\delta}}exp\left(-\frac{(x - \theta)^2}{2\delta}\right), \delta = \frac{\sigma}{r} \).
- Statistics: mean, mode = \( \theta \), variance = \( \delta \).

2.3.2 Type G model (Gamma Distribution)

For this model, we have:

\[1\text{These correspond to the following Pearson family of distributions: Gamma = type III, Beta = type I, and Gaussian = limit of type I, III, IV, V, or VI.}\]
Drift coefficient: \( f(X_t) = r(\theta - X_t), r > 0. \)

Diffusion coefficient: \( g(X_t) = \sqrt{\sigma X_t}, \sigma > 0. \)

Sde: \( dX_t = r(\theta - X_t)dt + \sqrt{\sigma X_t}dW_t. \)

Stationary density: \( \pi(x) = \left( \frac{x}{\delta} \right)^{-1+\frac{\theta}{\delta}} e^{-\frac{x}{\delta}} \frac{\Gamma\left(\frac{\theta}{\delta}\right)}{\Gamma\left(\frac{1-\theta}{\delta}\right)} \delta = \frac{\sigma}{2r}. \)

Statistics: mean = \( \theta \), mode = \( \theta - \delta \), variance = \( \delta \theta \).

### 2.3.3 Type B model (Beta Distribution)

For this model, we have:

Drift coefficient: \( f(X_t) = r(\theta - X_t), r > 0. \)

Diffusion coefficient: \( g(X_t) = \sqrt{\sigma X_t(1-X_t)}, \sigma > 0. \)

Sde: \( dX_t = r(\theta - X_t)dt + \sqrt{\sigma X_t(1-X_t)}dW_t. \)

Stationary density: \( \pi(x) = \frac{\Gamma\left(\frac{1}{\delta}\right)}{\Gamma\left(\frac{\theta}{\delta}\right) \Gamma\left(\frac{1-\theta}{\delta}\right)} x^{-1+\frac{\theta}{\delta}} (1-x)^{-1+\frac{1-\theta}{\delta}}, \delta = \frac{\sigma}{2r}. \)

Statistics: mean = \( \theta \), mode = \( \frac{\theta - \delta}{1-2\delta}, \) variance = \( \frac{\theta(1-\theta)}{1+\delta}. \)

### 2.3.4 Example of use

simulation \( M \)-sample (\( M = 100 \)) for the random variable \( X_t \) at time \( t \) by a simulated diffusion processes, using the function \texttt{AnaSimX}.

\texttt{AnaSimX(N, M, t0, Dt, T = 1, X0, v, drift, diff, Output = FALSE, Methods = c("Euler", "Milstein", "MilsteinS", "Ito-Taylor", "Heun", "RK3"))}

Consider for example the stochastic process the type \( N : dX_t = 2(1 - X_t)dt + dW_t. \)

\begin{verbatim}
R> r = 2
R> theta = 1
R> sigma = 1
R> f_x <- expression(r * (theta - x))
R> g_x <- expression(sqrt(sigma))
R> AnaSimX(N = 1000, M = 100, t0 = 0, Dt = 0.01, T = 10, X0 = -6, v = 7, drift = f_x, diff = g_x, Methods = "Euler")
\end{verbatim}
Figure 5: Simulation 100-samples of the random variable $X_t$ at time $v_t = 7$ by a simulated 100 trajectories of $dX_t = 2(1 - X_t)dt + dW_t$, used Euler algorithm, with $X_0 = -6$.

R> X

```
[1] 0.42457787 0.89022987 0.53798985 0.85601970 0.87574259
[6] 0.13023321 0.78466182 1.67646649 1.86814912 0.91651605
[11] 0.40458587 1.25308171 0.50399329 1.02546173 1.02647338
[16] 0.26630316 1.15195695 0.66666422 0.85174517 0.76291999
[21] 0.44828044 0.27588758 1.01659444 0.99617363 1.00829577
[26] 0.86676310 0.64266320 0.26960628 0.98420215 0.67478779
[31] 0.87145246 1.27619465 0.77522554 0.94704303 1.42153343
[36] 0.98755623 0.77798967 0.08774791 1.16267498 0.84184574
[41] 1.03631156 0.57812454 0.88114625 1.11535893 1.4038370
[46] 1.16252625 0.87735322 1.55669661 1.30152170 1.02433169
[51] 1.23616081 1.51367164 1.61080710 0.99608594 0.59199640
[56] 1.35987961 1.81565309 0.93158510 1.24933297 0.94759695
[61] 0.90637034 0.28367297 1.06232243 0.39463121 0.95167687
[66] 0.68991567 0.52513906 0.82081028 1.11318080 1.25055621
[71] 0.76350623 1.29918398 0.32244246 0.75747775 1.05616919
[76] 1.13505537 1.37253188 0.84595736 1.23985729 1.06358587
[81] 1.20910711 1.54841177 0.94479993 1.52138324 0.93499736
[86] 1.50113064 0.43927942 -0.09752953 0.99846875 1.69256126
```
Estimate parameters of the normal distribution by the method of maximum likelihood. The function Ajdnorm needs as input the random variable $X$, and initial values for optimizer (mean, sd) with the confidence level required. Two methods are used for estimation for the stationary distribution, the histograms used the function hist_general, and the kernel methods by used Kern_general.

```r
hist_general(Data, Breaks, Law = c("exp", "GAmma", "chisq", "Beta", "fisher", "student", "weibull", "Normlog", "Norm"))
Kern_general(Data, bw, k, Law = c("exp", "GAmma", "chisq", "Beta", "fisher", "student", "weibull", "Normlog", "Norm"))
```

```r
R> Ajdnorm(X, starts = list(mean = 1, sd = 1), leve = 0.95)

Profiling...

$summary
Maximum likelihood estimation

Call:
mle(minuslogl = lik, start = starts)

Coefficients:

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>0.9285440</td>
<td>0.04218993</td>
</tr>
<tr>
<td>sd</td>
<td>0.4197845</td>
<td>0.02983209</td>
</tr>
</tbody>
</table>

-2 log L: 109.0828

$coef

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>sd</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>0.9285440</td>
<td>0.4197845</td>
</tr>
</tbody>
</table>

$AIC

[1] 113.0828

$vcov

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>sd</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
$\text{mean} \quad 1.779990 \times 10^{-3} \quad -1.799151 \times 10^{-10}$

$\text{sd} \quad -1.799151 \times 10^{-10} \quad 8.899539 \times 10^{-4}$

$\text{confint}$

$2.5 \% \quad 97.5 \%$

$\text{mean} \quad 0.8450425 \quad 1.0120456$

$\text{sd} \quad 0.3674690 \quad 0.4858145$

R> hist_general(Data = X, Breaks = 'Sturges', Law = "Norm")
R> Kern_general(Data = X, bw = "Bcv", k="gaussian", Law = "Norm")

Figure 6: Estimation stationary distribution used histograms and kernel methods.

3 Attractive models

The problem of dispersion is a very complex phenomenon is many problems dealing with environment, biology, physics, chemistry, etc . . . , the dynamical behavior of such phenomenon is a random process, often hard to modeling mathematically. This problem, have been proposed by many authors [17, 19, 18, 3, 5]. For many dispersal problems, the diffusion processes are used to modeling the behavior of the dispersal phenomenon. For example, two important models can be indicated, the first is proposed by [5], and describes the dispersion of a pollutant on an area of shallow water (attractive model for one diffusion process), the second model is the two diffusions processes in attraction proposed by [6] (attractive model for two diffusions processes), which describes the dynamical behavior of a two insects, one attracts the other.
3.1 Attractive model for one diffusion process

Consider a shallow water area with depth \( L(x, y, z, t) \), horizontal \( U_w(x, y, z, t) \) and \( V_w(x, y, z, t) \), \( S_w(x, y, z, t) \) the velocities of the water in respectively the \( x- \), \( y- \) and \( z- \) directions, and \( U_a(x, y, z) \), \( V_a(x, y, z) \), \( S_a(x, y, z) \) the velocities of a particle caused by an attractive mechanism. Let \((X_t, Y_t, Z_t)\) be the position of a particle injected in the water at time \( t = t_0 \) at the point \((x_0, y_0, z_0)\).

For a single particle, we propose the following dispersion models family \([5]\):

\[
\begin{align*}
\frac{dX_t}{dt} &= \left( -U_a + U_w + \frac{\partial L}{\partial x} D + \frac{\partial D}{\partial x} \right) dt + \sqrt{2D} dW_{1,t}, \\
\frac{dY_t}{dt} &= \left( -V_a + V_w + \frac{\partial L}{\partial y} D + \frac{\partial D}{\partial y} \right) dt + \sqrt{2D} dW_{2,t}, \\
\frac{dZ_t}{dt} &= \left( -S_a + S_w + \frac{\partial L}{\partial z} D + \frac{\partial D}{\partial z} \right) dt + \sqrt{2D} dW_{3,t},
\end{align*}
\]

with:

\[
U_a = \frac{Kx}{\left(\sqrt{x^2 + y^2 + z^2}\right)^{s+1}}, \quad V_a = \frac{Ky}{\left(\sqrt{x^2 + y^2 + z^2}\right)^{s+1}}, \quad S_a = \frac{Kz}{\left(\sqrt{x^2 + y^2 + z^2}\right)^{s+1}}.
\]

and \(s \geq 1, \ K > 0, \ W_{1,t}^1, W_{2,t}^2, W_{3,t}^3\) are Brownian motions.

\(U_w(x, y, z, t)\) and \(V_w(x, y, z, t)\), \(S_w(x, y, z, t)\) are neglected and the dispersion coefficient \(D(x, y, z)\) is supposed constant and equal to \(\frac{1}{2}\sigma^2\), \(\sigma > 0\).

Using Itô’s transform for (10), it is shown that the radial process \(R_t = \| (X_t, Y_t, Z_t) \|\) is a Markovian diffusion, solution of the stochastic differential equation, given by:

\[
\frac{dR_t}{dt} = \left( \frac{\sigma^2 R_t^{s+1} - K}{R_t} \right) dt + \sigma \tilde{W}_t, \quad t \in [0, T]
\]

where \(2K > \sigma^2\) and \(\| . \|\) is the Euclidean norm and \(\tilde{W}_t\) is a determined Brownian motion.

We take interest in the random variable \(\tau_c^{(s)}\) first passage time (See \([7, 8]\)) of the particle through a sphere of radius \(c\), centered at the origin of \(\mathbb{R}^3\)-space. The random variable \(\tau_c^{(s)}\) is defined by :

\[
\tau_c^{(s)} = \inf\{ t \geq 0 | R_t \leq c \text{ and } R_0 = r \}
\]

This variable plays an important role in the prediction of the rate of particles reaching the attractive center. We denote by \(M_{s,\sigma}\) the family of models defined by (10) or (11).

3.1.1 Code example

For example the simulation of the model \(M_{(s=1,\sigma)}\) can be made by discretization of equations (10) or (11) with \(s = 1\), a simulate and estimate the density function of the random variable \(\tau_c^{(1)} = 1/\tau_c^{(1)}\) (12).

Simulation the models (10) by used the function RadialP3D_1 with \(s = 1\).
R> RadialP3D_1(N = 5000, t0 = 0, Dt = 0.001, X0 = 1, Y0 = 0.5, Z0 = 0.5,
v = 0.2, K = 2, Sigma = 0.2)

Figure 7: Simulation 3-dimensional attractive model $M_{s=1,\sigma=0.2}$ for one diffusion process.

Simulation 100-sample for the first passage time “FPT” of the model $M_{s=1,\sigma}$ used the function `tho_M1`, and estimate the density function of the $\tilde{\tau}_c^{(1)}$ by the Gamma law.

R> tho_M1(N = 1000, M = 100, t0 = 0, T = 1, R0 = 1, v = 0.2, K = 4, sigma =0.9,
Methods = "Euler")

R> FPT

[1] 0.071 0.124 0.046 0.117 0.165 0.126 0.178 0.059 0.218 0.118
[11] 0.305 0.064 0.135 0.188 0.271 0.161 0.099 0.123 0.086 0.137
[21] 0.135 0.101 0.135 0.265 0.204 0.084 0.139 0.186 0.133 0.099
[31] 0.340 0.099 0.133 0.073 0.160 0.217 0.091 0.181 0.139 0.097
[41] 0.118 0.235 0.133 0.371 0.096 0.301 0.106 0.206 0.137 0.233
[51] 0.082 0.156 0.135 0.154 0.060 0.113 0.077 0.132 0.230 0.135
[61] 0.212 0.169 0.238 0.236 0.211 0.098 0.176 0.117 0.205 0.176
[71] 0.220 0.116 0.167 0.073 0.220 0.160 0.099 0.169 0.102 0.197
[81] 0.128 0.172 0.080 0.196 0.120 0.219 0.161 0.283 0.254 0.113
[91] 0.392 0.050 0.138 0.111 0.074 0.216 0.202 0.098 0.091 0.056

R> Ajdgamma(X = 1/FPT, starts = list(shape = 1, rate = 1), leve = 0.95)

Profiling...
$summary
Maximum likelihood estimation
Call:
\[
\text{mle(minuslogl = lik, start = starts)}
\]

Coefficients:
\[
\begin{align*}
\text{Estimate} & \quad \text{Std. Error} \\
\text{shape} & \quad 5.0320084 \quad 0.6927984 \\
\text{rate} & \quad 0.6426731 \quad 0.09304918 \\
-2 \log L & \quad 514.6746 \\
\end{align*}
\]

$\text{coef}$
\[
\begin{align*}
\text{shape} & \quad 5.0320084 \\
\text{rate} & \quad 0.6426731 \\
\end{align*}
\]

$\text{AIC}$
\[
[1] \quad 518.6746
\]

$v\text{cov}$
\[
\begin{align*}
\text{shape} & \quad 0.4799716 \quad 0.06130030 \\
\text{rate} & \quad 0.0613003 \quad 0.00865815 \\
\end{align*}
\]

$\text{confint}$
\[
\begin{align*}
2.5 \% & \quad 97.5 \% \\
\text{shape} & \quad 3.7969651 \quad 6.5185981 \\
\text{rate} & \quad 0.4768285 \quad 0.8423622 \\
\end{align*}
\]

R> hist\_general(Data = 1/FPT, Breaks = 'Sturges', Law = "GAmma")
R> Kern\_general(Data = 1/FPT, bw = ‘Ucv’, k = "gaussian", Law = "GAmma")

Figure 8: Estimation the density function of $\tilde{\tau}^{(1)}_{c=0.2}$ by the Gamma law, used histograms and kernel methods.
3.2 Attractive model for two diffusions processes

In the following paragraph, we will propose a model of two diffusions in attraction \( M_{\mu(t)}^{\sigma} (V_t^{(1)}) \) and \( M_0^{\sigma} (V_t^{(2)}) \), can modeling the behavior of two insects, one attracts the other. Considers \( V_t^{(1)} = (X_{t,1}, X_{t,2}, X_{t,3}) \) and \( V_t^{(2)} = (Y_{t,1}, Y_{t,2}, Y_{t,3}) \) two random processes of diffusion, which one supposes respectively representing the positions of a male insect and an insect female, and the male is attracted by the female. The behavior of the female is supposed to be a process of Brownian motion, defined by the following equation

\[
dV_t^{(2)} = \sigma I_{3 \times 3} dW_t
\]

Whereas the behavior of the male is supposed to be a process of diffusion, whose drift is directed, at every moment \( t \), towards the position of the female, and who is given by

\[
dV_t^{(1)} = dV_t^{(2)} + \mu_{m+1}(\|D_t\|) D_t dt + \sigma I_{3 \times 3} d\tilde{W}_t,
\]

where \( W_t \) and \( \tilde{W}_t \) are two Brownian motion independent, and

\[
\begin{align*}
D_t &= V_t^{(1)} - V_t^{(2)} \\
\mu_m(\|d\|) &= -\frac{K}{\|d\|^m}
\end{align*}
\]

\( K \) and \( m \) are positive constants.

The model suggested is following form

\[
dV_t^{(1)} \Leftrightarrow \begin{cases} 
    dX_{t,1} = dY_{t,1} - \frac{K(X_{t,1} - Y_{t,1})}{\sqrt{(X_{t,1} - Y_{t,1})^2 + (X_{t,2} - Y_{t,2})^2 + (X_{t,3} - Y_{t,3})^2}}^m dt + \sigma d\tilde{W}_t^1 \\
    dX_{t,2} = dY_{t,2} - \frac{K(X_{t,2} - Y_{t,2})}{\sqrt{(X_{t,1} - Y_{t,1})^2 + (X_{t,2} - Y_{t,2})^2 + (X_{t,3} - Y_{t,3})^2}}^m dt + \sigma d\tilde{W}_t^2 \\
    dX_{t,3} = dY_{t,3} - \frac{K(X_{t,3} - Y_{t,3})}{\sqrt{(X_{t,1} - Y_{t,1})^2 + (X_{t,2} - Y_{t,2})^2 + (X_{t,3} - Y_{t,3})^2}}^m dt + \sigma d\tilde{W}_t^3
\end{cases}
\]

\[
dV_t^{(2)} \Leftrightarrow \begin{cases} 
    dY_{t,1} = \sigma dW_t^1 \\
    dY_{t,2} = \sigma dW_t^2 \\
    dY_{t,3} = \sigma dW_t^3
\end{cases}
\]

Using Itô’s transform for (16) and (17), it is shown that the process \( X_t = \|(V_t^{(1)}, V_t^{(2)})\| \) is a Markovian diffusion, solution of the stochastic differential equation, given by:

\[
dX_t = \frac{\sigma^2 X_t^{m-1} - K}{X_t^m} dt + \sigma d\overline{W}_t, \quad t \in [0, T]
\]

where \( K > \sigma^2 \) and \( \|\| \) is the Euclidean norm and \( \overline{W}_t \) is a determined Brownian motion.
This model makes it possible to carry out a dynamic simulation of the real phenomenon. Using these simulations, one can also estimate the density of probability of the moment of the first meeting $\tau(V^{(1)}_t, V^{(2)}_t)$ between the two insects, defined by

$$\tau(V^{(1)}_t, V^{(2)}_t) = \lim_{c \to 0} \inf\{t \geq 0 | \|D_t\| \leq c\}$$

3.2.1 Code example

For example the simulation two dimensional of the phenomenon $M^\sigma_{\mu(\cdot)}(V^{(1)}_t) \leftrightarrow M^\sigma_{\mu(\cdot)}(V^{(2)}_t)$, can be made by discretization of equations (16) and (17), by used the function `TowDiffAtra2D`

```R
R> TowDiffAtra2D(N = 5000, t0 = 0, Dt = 0.001, T = 1, X1_0 = 2,
X2_0 = 2, Y1_0 = -0.5, Y2_0 = -1, v = 0.05,
K = 3, m = 0.1, Sigma = 0.3)
```

![Figure 9: Illustration of simulation two dimensional of a trajectory of the interaction between two insects.](image)

Simulation three dimensional of the phenomenon by used the function `TowDiffAtra3D`, and simulation 100-sample for the moment of the first meeting $\tau(V^{(1)}_t, V^{(2)}_t)$ between the two insects used the function `tho_02diff`.

```R
R> TowDiffAtra3D(N = 5000, t0 = 0, Dt = 0.001, T = 1, X1_0 = 1,
X2_0 = 0.5, X3_0 = 0, Y1_0 = -0.5, Y2_0 = 0.5,
Y3_0 = -1, v = 0.05, K = 3, m = 0.1, Sigma = 0.15)
```
Figure 10: Illustration of simulation three dimensional of a trajectory of the interaction between two insects.

\[
\begin{align*}
R & \text{> tho}_02\text{diff}(N = 1000, M = 100, t0 = 0, Dt = 0.001, T = 1, \\
& X1_0 = 1, X2_0 = 1, Y1_0 = 0.5, Y2_0 = 0.5, \\
& v = 0.05, K = 4, m = 0.2, Sigma = 0.2) \\
R & \text{> FPT} \\
& \begin{bmatrix}
0.143 & 0.109 & 0.123 & 0.106 & 0.133 & 0.123 & 0.189 & 0.120 & 0.270 & 0.140 & 0.198 \\
0.144 & 0.107 & 0.092 & 0.165 & 0.178 & 0.124 & 0.146 & 0.113 & 0.136 & 0.158 & 0.169 \\
0.108 & 0.160 & 0.143 & 0.201 & 0.122 & 0.091 & 0.154 & 0.096 & 0.189 & 0.198 & 0.147 \\
0.147 & 0.128 & 0.105 & 0.192 & 0.106 & 0.139 & 0.174 & 0.134 & 0.105 & 0.090 & 0.165 \\
0.284 & 0.098 & 0.136 & 0.092 & 0.093 & 0.077 & 0.149 & 0.171 & 0.125 & 0.151 & 0.122 \\
0.221 & 0.199 & 0.154 & 0.140 & 0.145 & 0.217 & 0.106 & 0.097 & 0.121 & 0.131 & 0.153 \\
0.152 & 0.168 & 0.127 & 0.100 & 0.120 & 0.130 & 0.130 & 0.181 & 0.175 & 0.166 & 0.102 & 0.136 \\
0.127 & 0.136 & 0.137 & 0.081 & 0.112 & 0.204 & 0.113 & 0.109 & 0.185 & 0.130 & 0.186 \\
0.193 & 0.140 & 0.106 & 0.191 & 0.131 & 0.203 & 0.123 & 0.128 & 0.132 & 0.158 & 0.126
\end{bmatrix}
\]
R \text{> Ajdgamma(X = 1/FPT, starts = list(shape = 1, rate = 1), leve = 0.95)}
\text{Profiling...}
$\text{summary}$
\text{Maximum likelihood estimation}
\text{Call:}
\text{mle(minuslogl = lik, start = starts)}
\text{Coefficients:}
Estimate Std. Error
shape  15.525968 2.1834585
rate   2.077233 0.2968927
-2 log L: 403.2067
$coef
    shape  rate
15.525968 2.077233
$AIC
[1] 407.2067
$vcov
      shape  rate
shape 4.7674910 0.6378467
rate  0.6378467 0.08814528
$confint
      2.5 % 97.5 %
shape 11.657589 20.252660
rate  1.551179  2.719831

R> hist_general(Data = 1/FPT, Breaks = 'Sturges', Law = "GAmma")
R> Kern_general(Data = 1/FPT, bw = 'Ucv', k = "gaussian", Law = "GAmma")

Figure 11: Estimation the density of probability of the moment of the first meeting between the two insects, used histograms and kernel methods.
4 Graphical User Interface for Sim.DiffProc

Unlike S-PLUS, R does not incorporate a statistical graphical user interface (GUI), but it does include tools for building GUIs. Based on the tcltk package (which furnishes an interface to the Tcl/Tk GUI toolkit), the Sim.DiffProcGUI package provides a graphical user interface for some functions in the Sim.DiffProc package, the design objectives of the Sim.DiffProcGUI were as follows: to support Sim.DiffProc, through an easy-to-use, extensible, crossplatform GUI, keep things relatively simple, and to render visible, in a reusable form. The Sim.DiffProcGUI package uses a simple and familiar menu/dialog-box interface. Top-level menus include File, Edit, Brownian Motion, Stochastic Integral, Stochastic Models, Parametric Estimation, Numerical Solution of SDE, Statistical Analysis, and Help.

Each dialog box includes a Help button, which leads to a relevant help page, the R-Sim.DiffProc console also provides the ability to edit, enter, and re-execute commands. Data sets in this GUI are simply R data frames, and can be read from attached packages or imported from files, although several data frames may reside in memory.

Once R is running, simply loading the Sim.DiffProcGUI package by typing the command library("Sim.DiffProcGUI") into the R Console starts the GUI. After loading the package, the GUI window should appear more or less as in the following figure.

![Figure 12: Graphical User Interface for Sim.DiffProc package at start-up.](image)

---

2 R is a programming language and software environment for statistical computing and graphics, are available for download from CRAN at URL: [http://www.r-project.org/](http://www.r-project.org/)

3 The Sim.DiffProcGUI package, described in this paper, is based on the tcltk package [12][13], which provides an interface to Tcl/Tk [26].
R> library(Sim.DiffProcGUI)

This and other screen images were created under Windows seven, if you use another
version of Windows (or, of course, another computing platform), then the appearance of
the screen may differ.\footnote{The examples in this document use the Windows version of R, R, however, is available on other
computing platforms as well (Macintosh computers and Unix/Linux systems), and the use of R and the
Sim.DiffProcGUI package on these other systems is very similar to their use under Windows.}

The Sim.DiffProcGUI package and R Console windows float freely on the desktop. You
will normally use the menus and dialog boxes of the Sim.DiffProcGUI to read, manipulate,
and simulated or analyze data.

5 Conclusion

This paper introduces new package Sim.DiffProc for a simulation of diffusion processes in
R language, and graphical user interface (GUI) for this package, the actual development
of computing tools (software and hardware) has motivated us to analyze by simulation.
Many theoretical problems on the stochastic differential equations have become the object
of practical research, as statistical analysis and the simulation the solution of SDE, enabled
many searchers in different domains to use these equations to modeling and to analyse
practical problems. The dispersion problem that we have treated in this paper is a good
example which shows the important use of the SDE in the practice, this problem is very
hard, hence the SDE approach seems to be a good approximation to treat such problem,
the difficulty to obtain the exact solution of the SDE, the simulation gives information on
the density function of the random variable first passage time $\tau_c$ and enables to estimate a
density function, this estimation is based, on either, the histograms and the kernel methods.
These results can be applied, as a first approximation, of the dispersal phenomenon, in
presence of an attractive source. The density function of $\tau_c$ can be used to determine the
rate of the pollutant particles in a neighborhood of the attractive centre, it has been shown
by graphical and numerical simulations. The simulation studies implemented in R language
seems very preferment and efficient, because it is a statistical environment, which permits
to realize and to validate the simulations.

Acknowledgments

The work described in this paper was supported by the project entitled by "Statistical
and Stochastic Modeling of the Complex Structures" with code B00220080009, within the
framework National Commission of Evaluation and Futurology of the University Research
(CNEPRU), Ministry for Higher Education and Scientific Research, Algeria.
References


