Itô and Stratonovich Stochastic Calculus with Sim.DiffProc Package Version 3.1

by Arsalane Chouaib Guidoum[1] and Kamal Boukhetala[2]

December 30, 2015

Abstract

We provide a detailed hands-on tutorial for the R Development Core Team [2015] add-on package Sim.DiffProc [Guidoum and Boukhetala, 2015], for symbolic and floating point computations in stochastic calculus and stochastic differential equations (SDEs). The package implement is introduced and it is explains how to use the `snssde1d`, `snssde2d` and `snssde3d` main functions in this package, for simulate uni- and multidimensional SDEs, notice that, in this version of the package, multidimensional SDEs need to have diagonal noise.

1 Background and motivation

Differential equations are used to describe the evolution of a system. SDEs arise when a random noise is introduced into ordinary differential equations (ODEs). Let us consider first an example to illustrate the need for simulated and to analyze the properties of solution of SDEs. Many (or even most) processes in nature and technology are driven by (temperature, energy, velocity, concentration,. . .) changes. Such processes are called diffusion (or dispersion) processes because the quantity considered (e.g., temperature) is distributed to an equilibrium state is established (i.e., until the differences that drive the process are minimized). There are many examples of diffusion processes. Diffusion is responsible for the distribution of sugar throughout a cup of coffee. Diffusion is the mechanism by which oxygen moves into our cells. Diffusion is of fundamental importance in many disciplines of physics, economics, mathematical finance, chemistry, and biology: diffusion is relevant to the sintering process (powder metallurgy, production of ceramics), the chemical reactor design, catalyst design in the chemical industry, doping during the production of semiconductors, and the transport of necessary materials such as amino acids within biological cells. The diffusion processes $\{X_t, t \geq 0\}$ solutions to SDEs, with slight notational variations, are standard in many books with applications in different fields, see, e.g., Soong [1973], Rolski et all [1998], Øksendal [2000], Klebaner [2005], Henderson and Plaschko [2006], Racicot and Théoret [2006], Allen [2007], Jedrzejewski [2009], Platen and Bruti-Liberati [2010], Stefano [2011], Heinz [2011],. . . .

If X_t is a differentiable function defined for $t \geq 0$, $f(x,t)$ is a function of x and t, and the following relation is satisfied for all t, $0 \leq t \leq T$,

$$\frac{dX_t}{dt} = X'_t = f(X_t, t), \quad \text{and} \quad X_0 = x_0, \tag{1}$$

then X_t is a solution of the ODE with the initial condition x_0. The above equation can be written in other forms (by continuity of X'_t):

$$X_t = X_0 + \int_0^t f(X_s, s)ds,$$

Before we give a rigorous definition of SDEs, we show how they arise as a randomly perturbed ODEs and give a physical interpretation.

The White noise process ξ_t is formally defined as the derivative of the Wiener process,

$$\xi_t \equiv \frac{dW_t}{dt} \equiv W'(t). \tag{2}$$

[1] Faculty of Mathematics.
Department of Probabilities & Statistics. University of Science and Technology Houari Boumediene.
BP 32 El-Alia, U.S.T.H.B, Algeria.
acguidoum@usthb.dz

[2] Faculty of Mathematics. University of Science and Technology Houari Boumediene.
BP 32 El-Alia, U.S.T.H.B, Algeria.
kboukhetala@usthb.dz

It does not exist as a function of t in the usual sense, since a Wiener process is nowhere differentiable. If $g(x,t)$ is the intensity of the noise at point x at time t, then it is agreed that $\int_0^T g(X_t,t)\xi_t dt = \int_0^T g(X_t,t)W'(t)dt = \int_0^T g(X_t,t)dW_t$, is Itô integral [Itô, 1944]. SDEs arise, for example, when the coefficients of ordinary equation (1) are perturbed by White noise. If X_t denotes the population density, then the population growth can be described by the ODE: $dX_t/dt = aX_t(1-X_t)$. The growth is exponential with birth rate a, when this density is small, and slows down when the density increases. Random perturbation of the birth rate results in the equation: $dX_t/dt = (a+\sigma\xi_t)X_t(1-X_t)$, or the SDE:

$$dX_t = aX_t(1-X_t)dt + \sigma X_t(1-X_t)dW_t, \quad X_0 = x_0.$$

There are thus two widely used types of stochastic calculus, Stratonovich and Itô (see Kloeden and Platen [1991a,b]), differing in respect of the stochastic integral used. Modelling issues typically dictate which version in appropriate, but once one has been chosen a corresponding equation of the other type with the same solutions can be determined. Thus it is possible to switch between the two stochastic calculus. Specifically, the processes $\{X_t, t \geq 0\}$ solution to the Itô SDE:

$$dX_t = f(t,X_t)dt + g(t,X_t)dW_t \tag{3}$$

where $\{W_t, t \geq 0\}$ is the standard Wiener process or standard Brownian motion, the drift $f(t,X_t)$ and diffusion $g(t,X_t)$ are known functions that are assumed to be sufficiently regular (Lipschitz, bounded growth) for existence and uniqueness of solution see Øksendal [2000]; has the same solutions as the Stratonovich[1] SDE:

$$dX_t = \underline{f}(t,X_t)dt + g(t,X_t)\circ dW_t \tag{4}$$

with the modified drift coefficient which is defined by:

$$\underline{f}(t,X_t) = f(t,X_t) - \frac{1}{2}g(t,X_t)\frac{\partial g}{\partial x}(t,X_t)$$

Many theoretical problems on the SDEs have become the object of practical research, enabled many searchers in different domains to use these equations to modeling and to analyse practical problems. We seek to motivate further interest in this specific field by introducing the `Sim.DiffProc` package [Guidoum and Boukhetala, 2015] to simulate the solution of a user defined Itô or Stratonovich uni- and multidimensional SDEs, estimate parameters from data and visualize statistics, and other features that will be explained in another vignettes (see `vignette(package="Sim.DiffProc")`), for example the determination of the first passage time in SDEs...; freely available on the Comprehensive R Archive Network (CRAN) at `http://CRAN.R-project.org/package=Sim.DiffProc`. There already exist a number of packages that can perform for stochastic calculus in R; see `sde` [Stefano, 2015] and `yuima` project package for SDEs [Stefano et all, 2014] a freely available on CRAN, this packages provides functions for simulation and inference for stochastic differential equations. It is the accompanying package to the book of Stefano [2008].

To install `Sim.DiffProc` package on your version of R($\geq$ 2.15.1), type the following line in the R console.

```
> 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:

```
> library(Sim.DiffProc)
```

A short list of help topics, corresponding to most of the commands in this package, is available by typing:

```
> library(help = "Sim.DiffProc")
```

This vignette contains only a brief introduction to using `Sim.DiffProc` package to simulate the solution of a user defined Itô or Stratonovich stochastic differential equations.

2 Itô vs Stratonovich SDE's

We can write an d-dimensional SDE in Itô form as:

$$d\boldsymbol{X_t} = \boldsymbol{F(t,X_t)}dt + \mathbf{G(t,X_t)}d\boldsymbol{W_t} \tag{5}$$

[1] To distinguish Stratonovich SDE from the Itô SDE we insert a $\circ$ before the differential dW_t in equation (4).

or in Stratonovich form as:

$$d\boldsymbol{X_t} = \boldsymbol{F(t, X_t)}dt + \mathbf{G(t, X_t)} \circ d\boldsymbol{W_t} \tag{6}$$

where $\boldsymbol{F(.)} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is called the *drift* of the SED's, $\mathbf{G}(.) : \mathbb{R}^d \rightarrow \mathbb{R}^{d\times m}$ is called the *diffusion* of the SDE's, and $\boldsymbol{W_t}$ is an m-dimensional process having independent[2] scalar Wiener process components. It is possible to convert from one interpretation to the other in order to take advantage of one of the approaches as appropriate: in the scalar case ($d = 1$), if the Itô SDE is as given in (3) then the Stratonovich SDE is given by (4). In other words (5) and (6), under different rules of calculus, have the same solution, for example: $dX_t = \mu X_t dt + \sigma X_t dW_t$, has solution: $X_t = X_0 \exp\left(\left(\mu - 0.5\sigma^2\right) t + \sigma W_t\right)$, as dose $dX_t = \left(\mu - 0.5\sigma^2\right) X_t dt + \sigma X_t \circ dW_t$. Obviously, in the case of additive noise ($g(.)$ independent of $x \Rightarrow \partial g/\partial x = 0$) the Itô and Stratonovich representations are equivalent ((5) $\equiv$ (6)). For multidimensional SDE's the relationship between the two representations is given by:

$$\underline{\boldsymbol{F}}_i(t, X_t) = \boldsymbol{F}_i(t, X_t) - \frac{1}{2}\sum_{j=1}^{d}\sum_{k=1}^{m}\mathbf{G}_{jk}(t, X_t)\frac{\partial \mathbf{G}_{ik}}{\partial X_j}(t, X_t), \qquad i = 1, \dots, d.$$

More in detail, the user can specify:

- The Itô or the Stratonovich SDE's to be simulated.
- The SDE's structural parameter value. i.e., the drift and diffusion coefficient of SDE's.
- The number of the SDE's solution trajectories to be simulated.
- The numerical integration method: Euler-Maruyama, Predictor-corrector, Milstein, Second Milstein, Itô Taylor order 1.5, Heun order 2; Runge-Kutta 1,2 and 3-stage. There a rich literature on simulation of solutions of the SDE's, e.g., Kloeden and Platen [1989, 1995], Kloeden et all [1994], Saito and Mitsui [1993], Kasdin [1995], Andreas [2003a,b, 2004, 2007, 2010].
- The time interval $[t_0, T]$ to be considered.
- The integration stepsize (discretization).

To obtain:

- Numerical solution of SDE's.
- Plot(s) of the solution trajectories.
- Plot(s) of the trajectories empirical mean, together with their $\alpha\%$ confidence bands.
- Monte-Carlo statistics of the solution process at the end time T, i.e. mean, median, quantiles, moments, skewness, kurtosis, $\alpha\%$ confidence bands,....

2.1 The `snssde1d()` function

Assume that we want to describe the following SDE in Itô[3] form:

$$dX_t = \frac{1}{2}\mu^2 X_t dt + \mu X_t dW_t, \qquad X_0 = x0 \tag{7}$$

in Stratonovich form:

$$dX_t = \frac{1}{2}\mu^2 X_t dt + \mu X_t \circ dW_t, \qquad X_0 = x0 \tag{8}$$

In the above $\boldsymbol{F}(t, x) = \frac{1}{2}\mu^2 x$ and $\mathbf{G}(t, x) = \mu x$, according to the notation of the (5) in the case $d = 1$ and W_t is a standard Wiener process ($m = 1$). This can be described in `Sim.DiffProc` by specifying the drift and diffusion coefficients as plain R expressions passed as strings which depends on the state variable `x` and time variable `t`, by specifying only one trajectorie (`M=1`) in $[t_0, T] = [0, 1]$, with integration stepsize $\Delta t = 0.001$ (by default: `Dt=(T-t0)/N`), $\mu = 0.5$ and $X_0 = 10$. specifying the type of SED by `type="ito"` or `type="str"` (by default `type="ito"`), and the numerical method used (by default `method="euler"`).

```
> f <- expression( (0.5*0.5^2*x) )
> g <- expression( 0.5*x )
> mod1 <- snssde1d(drift=f,diffusion=g,x0=10,M=1,N=1000)
> mod2 <- snssde1d(drift=f,diffusion=g,x0=10,M=1,N=1000,type="str")
> mod1
```

[2] In this version of the package, multidimensional SDE's need to have diagonal noise.
[3] The equivalently of (7) the following Stratonovich SDE: $dX_t = \mu X_t \circ dW_t$.

```
Ito Sde 1D:
        | dX(t) = (0.5 * 0.5^2 * X(t)) * dt + 0.5 * X(t) * dW(t)
Method:
        | Euler scheme of order 0.5
Summary:
        | Size of process       | N  = 1000.
        | Number of simulation  | M  = 1.
        | Initial value         | x0 = 10.
        | Time of process       | t in [0,1].
        | Discretization        | Dt = 0.001.

> mod2

Stratonovich Sde 1D:
        | dX(t) = (0.5 * 0.5^2 * X(t)) * dt + 0.5 * X(t) o dW(t)
Method:
        | Euler scheme of order 0.5
Summary:
        | Size of process       | N  = 1000.
        | Number of simulation  | M  = 1.
        | Initial value         | x0 = 10.
        | Time of process       | t in [0,1].
        | Discretization        | Dt = 0.001.
```

which can be plotted using the command `plot`, and the result is shown in Figure 1.

```
> plot(mod1)
> plot(mod2)
```

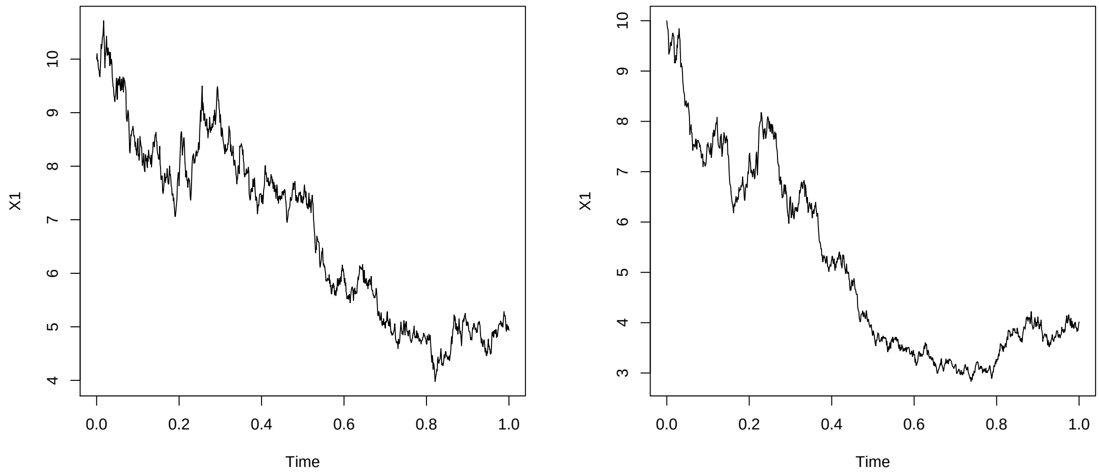

Figure 1: The `plot` function is used to draw a trajectory of a simulated `'snssde1d'` object.

If we simulate 50 trajectories and let the settings above unchanged (except for the number of simulations, of course); Using Monte-Carlo simulations, the following statistical measures (`S3 method` for class `'snssde1d'`) can be approximated for the X_t process at the end time T, i.e. X_T:

1. the expected (mean) value $\mathbb{E}(X_T)$; using the command `mean`.
2. the variance $\mathrm{var}(X_T)$.
3. the median $\mathrm{Med}(X_T)$; using the command `median`.
4. the quartile of X_T; using the command `quantile`.
5. the skewness and the kurtosis of X_T; using the command `skewness` and `kurtosis`.

6. the moments of X_T; using the command `moment`.
7. the $\alpha\%$ confidence bands of X_T; using the command `bconfint`.

Can be use the `summary` function to produce result summaries of the results of class `'snssde1d'`,

```
> mod1 <- snssde1d(drift=f,diffusion=g,x0=10,M=50,N=1000)
> mod2 <- snssde1d(drift=f,diffusion=g,x0=10,M=50,N=1000,type="str")
> summary(mod1)

        Monte-Carlo Statistics for X(t) at final time T = 1

                                X
Mean                    12.155176
Variance                25.821668
Median                  11.019853
First quartile           8.473991
Third quartile          15.059542
Skewness                 0.750526
Kurtosis                 2.810960
Moment of order 2       25.305235
Moment of order 3       98.478635
Moment of order 4     1874.231273
Moment of order 5    15398.110084
Bound conf Inf (95%)     5.328406
Bound conf Sup (95%)    23.116779

> summary(mod2)

        Monte-Carlo Statistics for X(t) at final time T = 1

                                X
Mean                     9.336365
Variance                23.003131
Median                   8.718121
First quartile           6.411597
Third quartile          10.280165
Skewness                 1.737797
Kurtosis                 7.162433
Moment of order 2       22.543069
Moment of order 3      191.725270
Moment of order 4     3789.958540
Moment of order 5    65590.226020
Bound conf Inf (95%)     3.565010
Bound conf Sup (95%)    19.095316
```

The flow of trajectories can be seen in Figure 2, reports the sample mean (red lines) of the solutions of the Itô SDE (7) and Stratonovich SDE (8), their empirical 95% confidence bands (from the 2.5*th* to the 97.5*th* percentile; blue lines), we can proceed as follows:

```
> plot(mod1,plot.type="single")
> lines(time(mod1),mean(mod1),col=2,lwd=2)
> lines(time(mod1),bconfint(mod1,level=0.95)[,1],col=4,lwd=2)
> lines(time(mod1),bconfint(mod1,level=0.95)[,2],col=4,lwd=2)
> legend("topleft",c("mean path",paste("bound of", 95,"% confidence")),
+        inset = .01,col=c(2,4),lwd=2,cex=0.8)
> dev.new()
> plot(mod2,plot.type="single")
> lines(time(mod2),mean(mod2),col=2,lwd=2)
> lines(time(mod2),bconfint(mod2,level=0.95)[,1],col=4,lwd=2)
> lines(time(mod2),bconfint(mod2,level=0.95)[,2],col=4,lwd=2)
> legend("topleft",c("mean path",paste("bound of", 95,"% confidence")),
+        inset = .01,col=c(2,4),lwd=2,cex=0.8)
```

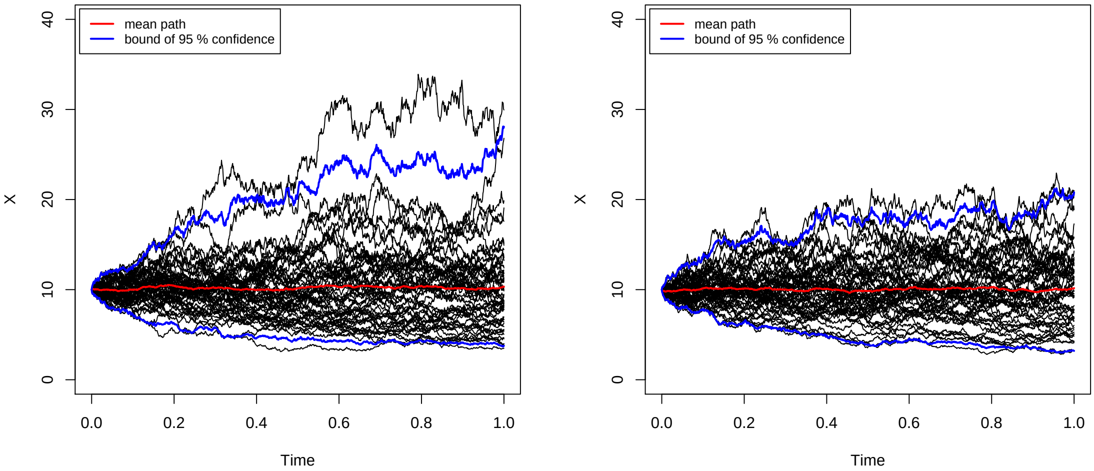

Figure 2: 50 trajectories of Itô SDE `'mod1'` (Left), and Stratonovich SDE `'mod2'` (Right).

2.1.1 Attractive model for one-diffusion processes

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 Hadeler et all [1980], Helland [1983], Heemink [1990], Boukhetala [1996]. For many dispersal problems, the diffusion processes are used to modeling the behavior of the dispersal phenomenon. 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 [Boukhetala, 1996]:

$$\begin{cases} dX_t = \left(-U_a + U_w + \frac{\frac{\partial L}{\partial x}}{L} D + \frac{\partial D}{\partial x}\right) dt + \sqrt{2D} dW_{1,t} \\ dY_t = \left(-V_a + V_w + \frac{\frac{\partial L}{\partial y}}{L} D + \frac{\partial D}{\partial y}\right) dt + \sqrt{2D} dW_{2,t} \\ dZ_t = \left(-S_a + S_w + \frac{\frac{\partial L}{\partial z}}{L} D + \frac{\partial D}{\partial z}\right) dt + \sqrt{2D} dW_{3,t} \end{cases} ,t \in [0,T] \qquad (9)$$

with:

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

where $s \geq 1$ and $K > 0$, $(W_{1,t}, W_{2,t}, W_{3,t})$ three independent Brownian motions. $U_w(x,y,z,t)$, $V_w(x,y,z,t)$ and $S_w(x,y,z,t)$ 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 system (9), 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:

$$dR_t = \left(\frac{0.5\sigma^2 R_t^{s-1} - K}{R_t^s}\right) dt + \sigma d\widetilde{W}_t, \qquad (10)$$

where: $2K > \sigma^2$ condition to ensure attractiveness; $\|.\|$ is the Euclidean norm and $\widetilde{W}_t$ is a Brownian motion. We simulate 50 trajectories to radial process (10) by `snssde1d` function, and the graphical representation can be seen in Figure 3,

```
> K = 4; s = 1; sigma = 0.2
> fx <- expression( ((0.5*sigma^2 *x^(s-1) - K)/ x^s) )
> gx <- expression( sigma )
> mod <- snssde1d(drift=fx,diffusion=gx, x0=3, M=50, N=1000)
> mod
```

```
Ito Sde 1D:
        | dX(t) = ((0.5 * sigma^2 * X(t)^(s - 1) - K)/X(t)^s) * dt + sigma * dW(t)
Method:
        | Euler scheme of order 0.5
Summary:
        | Size of process       | N  = 1000.
        | Number of simulation  | M  = 50.
        | Initial value         | x0 = 3.
        | Time of process       | t in [0,1].
        | Discretization        | Dt = 0.001.
```

```
> summary(mod)

        Monte-Carlo Statistics for X(t) at final time T = 1

                            X
Mean                 1.051489
Variance             0.296256
Median               1.086686
First quartile       0.809420
Third quartile       1.337744
Skewness             0.119612
Kurtosis             6.305873
Moment of order 2    0.290331
Moment of order 3    0.019288
Moment of order 4    0.553453
Moment of order 5    0.305881
Bound conf Inf (95%) 0.100276
Bound conf Sup (95%) 1.778940
```

```
> plot(mod,plot.type="single")
```

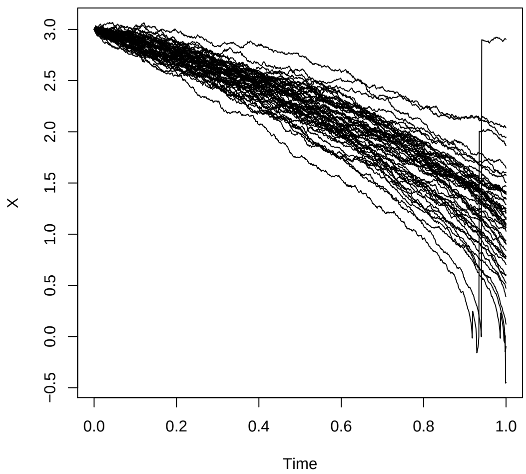

Figure 3: Flow paths for an attractive model of one-diffusion processes.

2.2 The snssde2d() function

A system of two SDE's for the couple (X_t,Y_t) driven by two independent Brownian motions $(W_{1,t}, W_{2,t})$. Remember that this version of the package handles SDE's with diagonal noise only. The following 2-dimensional SDE's into matrix form with a vector of drift expressions and a diffusion matrix in Itô form:

$$\begin{pmatrix} dX_t \\ dY_t \end{pmatrix} = \begin{pmatrix} f_x(t,X_t,Y_t) \\ f_y(t,X_t,Y_t) \end{pmatrix} dt + \begin{pmatrix} g_x(t,X_t,Y_t) & 0 \\ 0 & g_y(t,X_t,Y_t) \end{pmatrix} \begin{pmatrix} dW_{1,t} \\ dW_{2,t} \end{pmatrix} \tag{11}$$

in Stratonovich form:

$$\begin{pmatrix} dX_t \\ dY_t \end{pmatrix} = \begin{pmatrix} f_x(t, X_t, Y_t) \\ f_y(t, X_t, Y_t) \end{pmatrix} dt + \begin{pmatrix} g_x(t, X_t, Y_t) & 0 \\ 0 & g_y(t, X_t, Y_t) \end{pmatrix} \circ \begin{pmatrix} dW_{1,t} \\ dW_{2,t} \end{pmatrix} \tag{12}$$

We illustrate the usage of the `snssde2d` function to simulate the solution of a Itô (11) or Stratonovich (12) SDE's two dimensional, by a simple example and two applications.

2.2.1 Basic example 1

Assume that we want to describe the following SDE (2d) in Itô form:

$$\begin{cases} dX_t = 4(-1 - X_t)Y_t dt + 0.2 dW_{1,t} \\ dY_t = 4(1 - Y_t)X_t dt + 0.2 dW_{2,t} \end{cases} \tag{13}$$

for (13), we simulate a flow of 50 trajectories, with integration stepsize $t = 0.001$, and using stochastic Runge-Kutta methods 3-stage,

```
> fx   <- expression(4*(-1-x)*y)
> gx   <- expression(0.2)
> fy   <- expression(4*(1-y)*x)
> gy   <- expression(0.2)
> mod2d <- snssde2d(driftx=fx,diffx=gx,drifty=fy,diffy=gy,x0=1,y0=-1,M=50,
+                   Dt=0.001,method="rk3")
> mod2d

Ito Sde 2D:
        | dX(t) = 4 * (-1 - X(t)) * Y(t) * dt + 0.2 * dW1(t)
        | dY(t) = 4 * (1 - Y(t)) * X(t) * dt + 0.2 * dW2(t)
Method:
        | Runge-Kutta method of order 3
Summary:
        | Size of process        | N   = 1000.
        | Number of simulation   | M   = 50.
        | Initial values         | (x0,y0) = (1,-1).
        | Time of process        | t in [0,1].
        | Discretization         | Dt = 0.001.

> summary(mod2d)


        Monte-Carlo Statistics for (X(t),Y(t)) at final time T = 1

                             X         Y
Mean                 -0.712963  0.451724
Variance              0.014018  0.068494
Median               -0.710900  0.488832
First quartile       -0.807099  0.270641
Third quartile       -0.646953  0.630542
Skewness              0.299710 -0.117374
Kurtosis              2.692274  2.382129
Moment of order 2     0.013738  0.067124
Moment of order 3     0.000497 -0.002104
Moment of order 4     0.000529  0.011175
Moment of order 5     0.000059 -0.001198
Bound conf Inf (95%) -0.901761 -0.077090
Bound conf Sup (95%) -0.483406  0.915786
```

for plotted (with time) using the command `plot`, and in the plane (O, X, Y) using the command `plot2d`. The result is shown in Figure 4,

```
> plot(mod2d,pos=2)
> plot2d(mod2d)
```

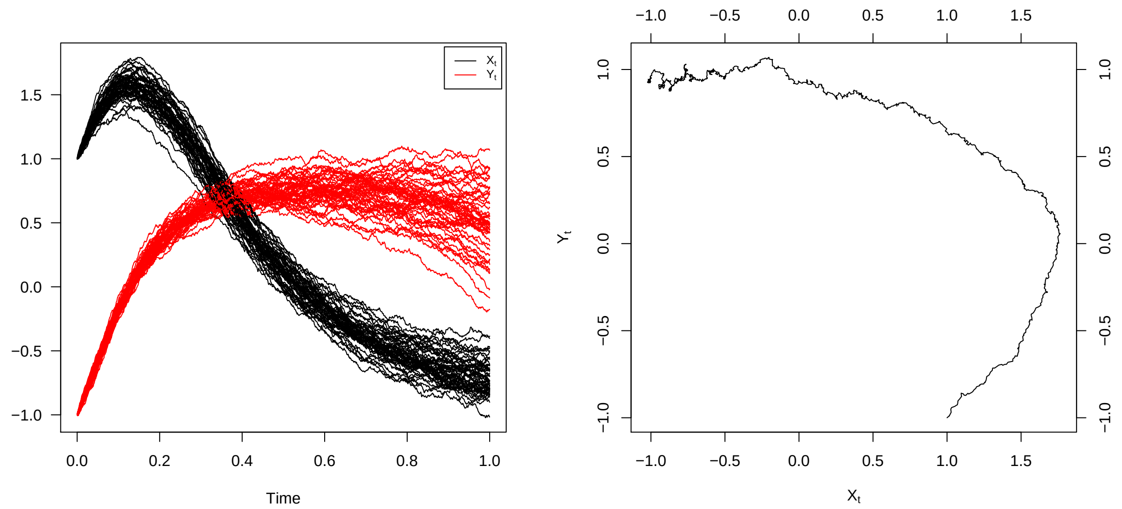

Figure 4: Simulation 50 trajectories of (13) (Left), representation of (13) in a plane (O, X, Y) (Right).

2.2.2 Kalman-Bucy Filter

Assume that the signal and the observation processes satisfy linear Itô SDE's [Klebaner, 2005, p. 379], with time time-dependent non-random coefficients, given by:

$$\begin{cases} dX_t = a_1(t)X_t dt + b_1(t)dW_{1,t} \\ dY_t = a_2(t)X_t dt + b_2(t)dW_{2,t} \end{cases} \tag{14}$$

with two independent Brownian motions $(W_{1,t}, W_{2,t})$, and initial conditions $(X_0, Y_0) = (0, 0)$, by specifying the drift and diffusion coefficients of two process X_t and Y_t as plain R expressions passed as strings which depends on the two state variables `(x,y)` and time variable t, with $a_1(t) = 2t$, $a_2(t) = 0.5t$ and $b_1(t) = b_2(t) = 0.1t$, integration stepsizeand $\Delta t = 0.001$ and numerical method used by default `"euler"`. Which can easily be implemented in R as follows:

```
> a1   <- function(t) 2*t
> a2   <- function(t) 0.5*t
> b1   = b2 <- function(t) 0.1*t
> fx     <- expression(a1(t)*x)
> gx     <- expression(b1(t))
> fy     <- expression(a2(t)*x)
> gy     <- expression(b2(t))
> mod2d <- snssde2d(driftx=fx,diffx=gx,drifty=fy,diffy=gy)
> mod2d

Ito Sde 2D:
        | dX(t) = a1(t) * X(t) * dt + b1(t) * dW1(t)
        | dY(t) = a2(t) * X(t) * dt + b2(t) * dW2(t)
Method:
        | Euler scheme of order 0.5
Summary:
        | Size of process        | N  = 1000.
        | Number of simulation   | M  = 1.
        | Initial values         | (x0,y0) = (0,0).
        | Time of process        | t in [0,1].
        | Discretization         | Dt = 0.001.
```

for plotted (with time) using the command `plot`, and the result is shown in Figure 5,

```
> plot(mod2d,union=TRUE,pos=3)
```

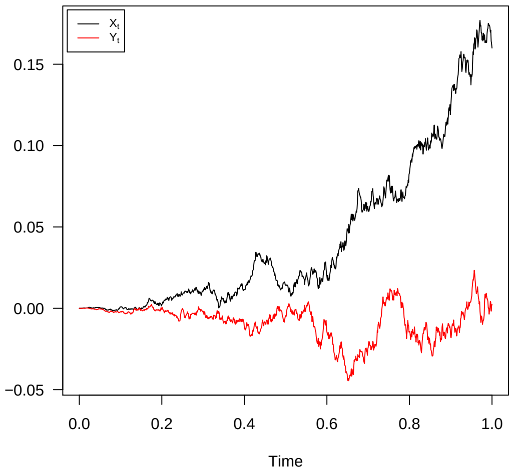

Figure 5: Kalman-Bucy Filter with time time-dependent non-random coefficients.

2.2.3 The stochastic Van-der-Pol equation

The Van der Pol equation is an ordinary differential equation that can be derived from the Rayleigh differential equation by differentiating and setting $\dot{x} = y$, see Van der Pol [1922], Naess and Hegstad [1994], Leung [1995] and for more complex dynamics in Van der Pol equation see Zhujun et all [2006]. It is an equation describing self-sustaining oscillations in which energy is fed into small oscillations and removed from large oscillations. This equation arises in the study of circuits containing vacuum tubes and is given by:

$$\ddot{X} - \mu(1 - X^2)\dot{X} + X = 0, \tag{15}$$

where x is the position coordinate (which is a function of the time t), and μ is a scalar parameter indicating the nonlinearity and the strength of the damping. Consider additive stochastic perturbations of the Van der Pol equation, and random excitation force of such systems by White noise ξ_t, with delta-type correlation functions $\mathbb{E}(\xi_t \xi_{t+h}) = 2\sigma\delta(h)$

$$\ddot{X} - \mu(1 - X^2)\dot{X} + X = \xi_t, \tag{16}$$

where $\mu > 0$. Its solution cannot be obtained in terms of elementary functions, even in the phase plane. The White noise ξ_t is formally derivative of Wiener process W_t (2). The representation as a system of two first order equations follows the same idea as in the deterministic case by letting $\dot{x} = y$, from physical equation (16) we get the above system:

$$\begin{cases} \dot{X} = Y \\ \dot{Y} = \mu\left(1 - X^2\right)Y - X + \xi_t \end{cases} \tag{17}$$

the system (17) can be mathematically translated by a system of Stratonovitch equations,

$$\begin{cases} dX_t = Y_t dt \\ dY_t = \left(\mu(1 - X_t^2)Y_t - X_t\right)dt + 2\sigma \circ dW_{2,t} \end{cases} \tag{18}$$

implemented in R as follows:

```
> mu = 4; sigma=0.1
> fx <- expression( y )
> gx <- expression( 0 )
> fy <- expression( (mu*( 1-x^2 )* y - x) )
> gy <- expression( 2*sigma)
> mod2d <- snssde2d(driftx=fx,diffx=gx,drifty=fy,diffy=gy,type="str",T=100,N=10000)
> mod2d

Stratonovich Sde 2D:
        | dX(t) = Y(t) * dt + 0 o dW1(t)
        | dY(t) = (mu * (1 - X(t)^2) * Y(t) - X(t)) * dt + 2 * sigma o dW2(t)
```

```
Method:
        | Euler scheme of order 0.5
Summary:
        | Size of process        | N  = 10000.
        | Number of simulation   | M  = 1.
        | Initial values         | (x0,y0) = (0,0).
        | Time of process        | t in [0,100].
        | Discretization         | Dt = 0.01.
```

which can be plotted in the plane (O, X, Y) using the command `plot2d`, and the result is shown in Figure 6 and 7:

```
> plot2d(mod2d)
> plot(mod2d,pos=3)
```

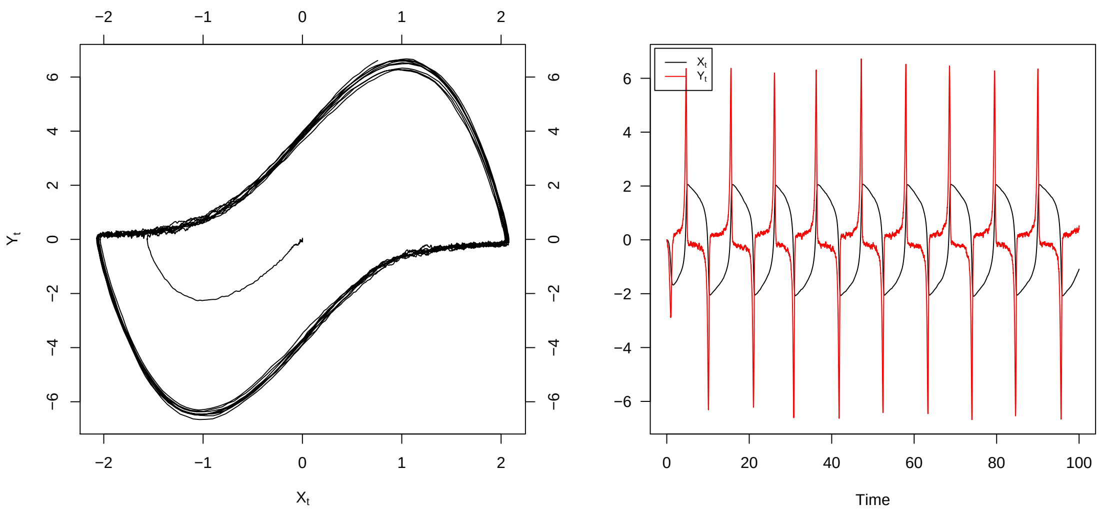

Figure 6: $2D$ stochastic Van-der-Pol equation (Left). Relaxation oscillation in the Van der Pol oscillator (Right) ($\mu = 4$ and $\sigma = 0.1$).

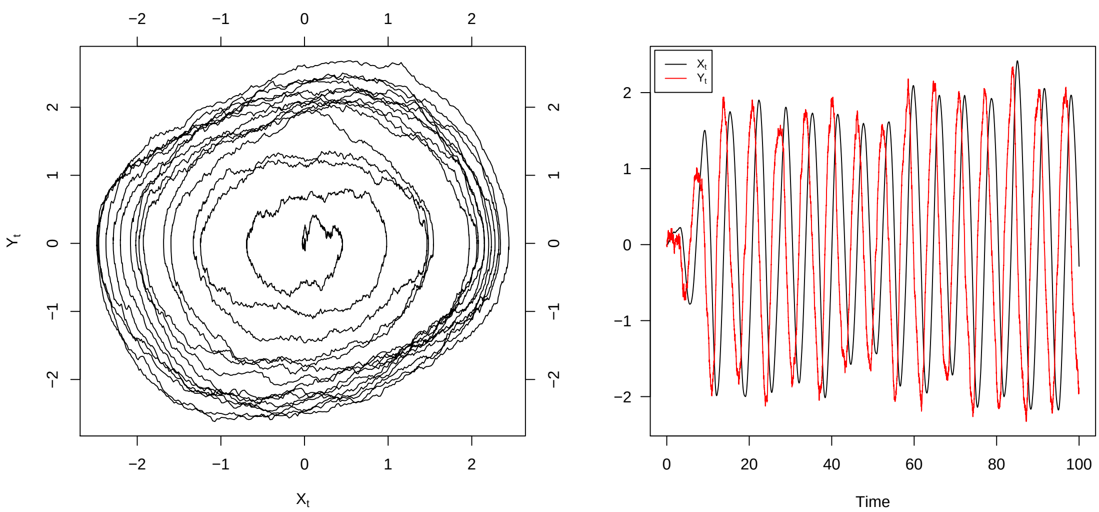

Figure 7: $2D$ stochastic Van-der-Pol equation (Left). Relaxation oscillation in the Van der Pol oscillator (Right) ($\mu = 0.2$ and $\sigma = 0.1$).

2.3 The `snssde3d()` function

A system of three SDE's for the triple (X_t, Y_t, Z_t) driven by three independent standard Brownian motions ($W_{1,t}$,$W_{2,t}$,$W_{3,t}$). The following 3-dimensional SDE's into matrix form with a vector of drift expressions and a diffusion matrix in Itô form:

$$\begin{pmatrix} dX_t \\ dY_t \\ dZ_t \end{pmatrix} = \begin{pmatrix} f_x(t, X_t, Y_t, Z_t) \\ f_y(t, X_t, Y_t, Z_t) \\ f_z(t, X_t, Y_t, Z_t) \end{pmatrix} dt + \begin{pmatrix} g_x(t, X_t, Y_t, Z_t) & 0 & 0 \\ 0 & g_y(t, X_t, Y_t, Z_t) & 0 \\ 0 & 0 & g_z(t, X_t, Y_t, Z_t) \end{pmatrix} \begin{pmatrix} dW_{1,t} \\ dW_{2,t} \\ dW_{3,t} \end{pmatrix} \tag{19}$$

in Stratonovich form:

$$\begin{pmatrix} dX_t \\ dY_t \\ dZ_t \end{pmatrix} = \begin{pmatrix} f_x(t, X_t, Y_t, Z_t) \\ f_y(t, X_t, Y_t, Z_t) \\ f_z(t, X_t, Y_t, Z_t) \end{pmatrix} dt + \begin{pmatrix} g_x(t, X_t, Y_t, Z_t) & 0 & 0 \\ 0 & g_y(t, X_t, Y_t, Z_t) & 0 \\ 0 & 0 & g_z(t, X_t, Y_t, Z_t) \end{pmatrix} \circ \begin{pmatrix} dW_{1,t} \\ dW_{2,t} \\ dW_{3,t} \end{pmatrix} \tag{20}$$

We illustrate the usage of the `snssde3d` function to simulate the solution of a Itô (19) or Stratonovich (20) SDE's three dimensional, by three applications.

2.3.1 Basic example 2

Assume that we want to describe the following SDE (3d) in Itô form:

$$\begin{cases} dX_t = 4(-1 - X_t)Y_t dt + 0.2 dW_{1,t} \\ dY_t = 4(1 - Y_t)X_t dt + 0.2 dW_{2,t} \\ dZ_t = 4(1 - Z_t)Y_t dt + 0.2 dW_{3,t} \end{cases} \tag{21}$$

for (21), we simulate a flow of 50 trajectories, with integration stepsize $t = 0.001$, and using stochastic Runge-Kutta methods 2-stage,

```
> fx <- expression(4*(-1-x)*y)
> gx <- expression(0.2)
> fy <- expression(4*(1-y)*x)
> gy <- expression(0.2)
> fz <- expression(4*(1-z)*y)
> gz <- expression(0.2)
> mod3d <- snssde3d(x0=2,y0=-2,z0=-2,driftx=fx,diffx=gx,drifty=fy,diffy=gy,
+                driftz=fz,diffz=gz,N=1000,M=50,method="rk2")
> mod3d

Ito Sde 3D:
        | dX(t) = 4 * (-1 - X(t)) * Y(t) * dt + 0.2 * dW1(t)
        | dY(t) = 4 * (1 - Y(t)) * X(t) * dt + 0.2 * dW2(t)
        | dZ(t) = 4 * (1 - Z(t)) * Y(t) * dt + 0.2 * dW3(t)
Method:
        | Runge-Kutta method of order 2
Summary:
        | Size of process       | N  = 1000.
        | Number of simulation  | M  = 50.
        | Initial values        | (x0,y0,z0) = (2,-2,-2).
        | Time of process       | t in [0,1].
        | Discretization        | Dt = 0.001.

> summary(mod3d)

  Monte-Carlo Statistics for (X(t),Y(t),Z(t)) at final time T = 1

                           X        Y        Z
Mean                 -0.770600 0.896747 0.790397
Variance              0.008100 0.088119 0.008337
Median               -0.762318 0.853475 0.793482
First quartile       -0.813898 0.668774 0.745630
Third quartile       -0.711699 1.154523 0.821978
```

```
Skewness             -0.536928 0.316727 0.377908
Kurtosis              2.742914 2.846316 3.383140
Moment of order 2     0.007938 0.086357 0.008170
Moment of order 3    -0.000391 0.008285 0.000288
Moment of order 4     0.000180 0.022102 0.000235
Moment of order 5    -0.000021 0.008133 0.000021
Bound conf Inf (95%) -0.974326 0.381735 0.629168
Bound conf Sup (95%) -0.628643 1.327309 1.005249
```

for plotted (with time) using the command `plot`, and in the plane (O, X, Y, Z) using the command `plot3D`. The result is shown in Figure 8,

```
> plot(mod3d,union = TRUE,pos=2)    ## with time
> plot3D(mod3d,display="persp")     ## in space (O,X,Y,Z)
```

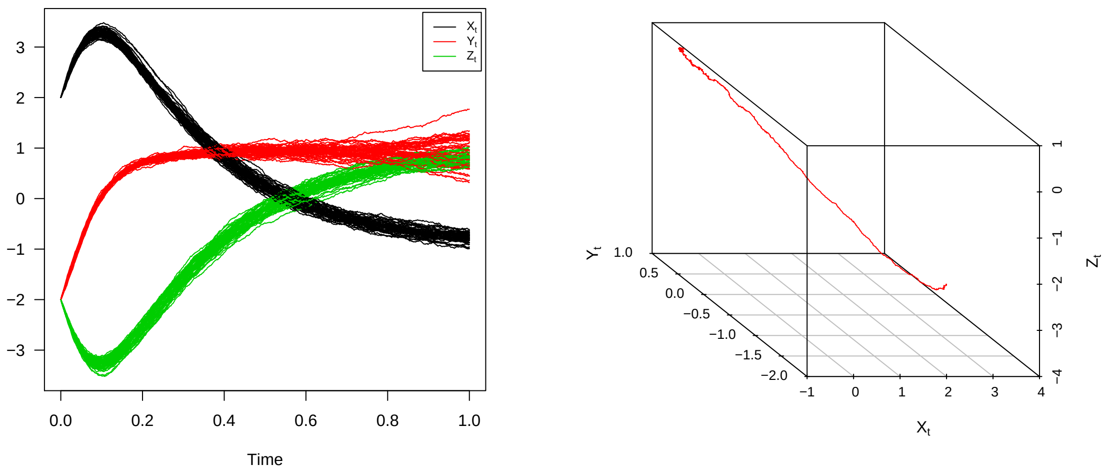

Figure 8: Simulation 50 trajectories of system (21) (Left), representation one path of (21) in a plane (O, X, Y, Z) (Right).

2.3.2 Attractive model for multidimensional diffusion processes

If we assume that $U_w(x, y, z, t)$, $V_w(x, y, z, t)$ and $S_w(x, y, z, t)$ are neglected and the dispersion coefficient $D(x, y, z)$ $\left(= 0.5\sigma^2\right)$ is constant. A system (9) becomes (see Boukhetala [1996]):

$$\begin{aligned} dX_t &= \left(\frac{-KX_t}{X_t^2 + Y_t^2 + Z_t^2}\right) dt + \sigma dW_{1,t} \\ dY_t &= \left(\frac{-KY_t}{X_t^2 + Y_t^2 + Z_t^2}\right) dt + \sigma dW_{2,t} \\ dZ_t &= \left(\frac{-KZ_t}{X_t^2 + Y_t^2 + Z_t^2}\right) dt + \sigma dW_{3,t} \end{aligned} \tag{22}$$

with initial conditions $(X_0, Y_0, Z_0) = (1, 1, 1)$, by specifying the drift and diffusion coefficients of three process X_t, Y_t and Z_t as plain R expressions passed as strings which depends on the three state variables `(x,y,z)` and time variable t, with integration stepsizeand $\Delta t = 0.0001$ and numerical method used by default `"euler"`. Which can easily be implemented (22) in R as follows:

```
> K = 4; s = 1; sigma = 0.2
> fx <- expression( (-K*x/sqrt(x^2+y^2+z^2)) )
> gx <- expression(sigma)
> fy <- expression( (-K*y/sqrt(x^2+y^2+z^2)) )
> gy <- expression(sigma)
> fz <- expression( (-K*z/sqrt(x^2+y^2+z^2)) )
```

```
> gz <- expression(sigma)
> mod3d <- snssde3d(driftx=fx,diffx=gx,drifty=fy,diffy=gy,driftz=fz,diffz=gz,
+                   N=10000,x0=1,y0=1,z0=1)
> mod3d

Ito Sde 3D:
        | dX(t) = (-K * X(t)/sqrt(X(t)^2 + Y(t)^2 + Z(t)^2)) * dt + sigma * dW1(t)
        | dY(t) = (-K * Y(t)/sqrt(X(t)^2 + Y(t)^2 + Z(t)^2)) * dt + sigma * dW2(t)
        | dZ(t) = (-K * Z(t)/sqrt(X(t)^2 + Y(t)^2 + Z(t)^2)) * dt + sigma * dW3(t)
Method:
        | Euler scheme of order 0.5
Summary:
        | Size of process        | N  = 10000.
        | Number of simulation   | M  = 1.
        | Initial values         | (x0,y0,z0) = (1,1,1).
        | Time of process        | t in [0,1].
        | Discretization         | Dt = 1e-04.
```

for plotted (with time) using the command `plot`, and in the space (O, X, Y, Z) using `plot3D` with two display types `("rgl","persp")`, the first with `rgl` package [Daniel and Duncan, 2015] and the second display with `scatterplot3d` package [Uwe et all, 2015]. The result is shown in Figure 9,

```
> plot3D(mod3d,display="persp",col="blue")  ## in space
> plot(mod3d,union=TRUE,pos=2)              ## with time
```

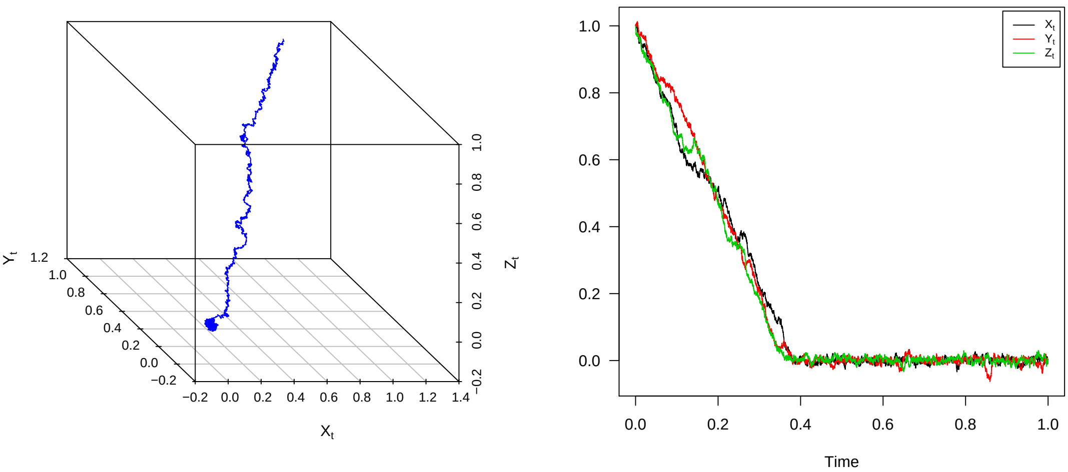

Figure 9: 3-dimensional attractive model $\mathcal{M}(K = 4, s = 1, \sigma = 0.2)$.

2.3.3 Stochastic Lotka-Volterra three-species

In the 1920s, the Italian mathematician Vito Volterra [Volterra, 1926] proposed a differential equation model to describe the population dynamics of two interacting species, a predator and its prey. Independently, in the United States, the very equations studied by Volterra were derived by Alfred Lotka [Lotka, 1925] to describe a hypothetical chemical reaction in which the chemical concentrations oscillate. The Lotka-Volterra model consists of the following system of (2D) differential equations:

$$\begin{cases} \dot{X} = aX - bXY \\ \dot{Y} = -cY + dXY \end{cases} \tag{23}$$

where Y_t and X_t represent, respectively, the predator population and the prey population as functions of time (for more details see, e.g., [Hofbauer and So, 1994],[Klebaner, 2005, p. 366]). The following model is proposed by Erica et all [2002] as:

$$\begin{cases} \dot{X} = aX - bXY \\ \dot{Y} = -cY + dXY - eYZ \\ \dot{Z} = -fZ + gYZ \end{cases} \tag{24}$$

The parameters $a, b, c, d, e, f > 0$, for the description of this model see Erica et all [2002]. We express mathematically the system (24) by Stratonovitch equations,

$$\begin{cases} dX_t = (aX_t - bX_tY_t)dt + \sigma \circ dW_{1,t} \\ dY_t = (-cY_t + dX_tY_t - eY_tZ_t)dt + \sigma \circ dW_{2,t} \\ dZ_t = (-fZ_t + gY_tZ_t)dt + \sigma \circ dW_{3,t} \end{cases} \tag{25}$$

simulate this system in space (O, X, Y, Z) using the function `snssde3d`, with parameters $a = b = c = d = e = f = 1$, $\sigma = 0.03$, $(X_0, Y_0, Z_0) = (0.5, 1, 2)$ and final time $T = 50$.

```
> fx <- expression((x - x*y))
> gx <- expression(0.03)
> fy <- expression(( -y + x*y-y*z ))
> gy <- expression(0.03)
> fz <- expression(( -z+ y*z ))
> gz <- expression(0.03)
> mod3d <- snssde3d(driftx=fx,diffx=gx,drifty=fy,diffy=gy,driftz=fz,diffz=gz,
+                   N=10000,T=20,x0=0.5,y0=1,z0=2,type="str")
> mod3d

Stratonovich Sde 3D:
        | dX(t) = (X(t) - X(t) * Y(t)) * dt + 0.03 o dW1(t)
        | dY(t) = (-Y(t) + X(t) * Y(t) - Y(t) * Z(t)) * dt + 0.03 o dW2(t)
        | dZ(t) = (-Z(t) + Y(t) * Z(t)) * dt + 0.03 o dW3(t)
Method:
        | Euler scheme of order 0.5
Summary:
        | Size of process       | N  = 10000.
        | Number of simulation  | M  = 1.
        | Initial values        | (x0,y0,z0) = (0.5,1,2).
        | Time of process       | t in [0,20].
        | Discretization        | Dt = 0.002.
```

The result is shown in Figure 10,

```
> plot3D(mod3d,"persp",col="blue")  ## in space
> plot(mod3d,union=TRUE)            ## with time
```

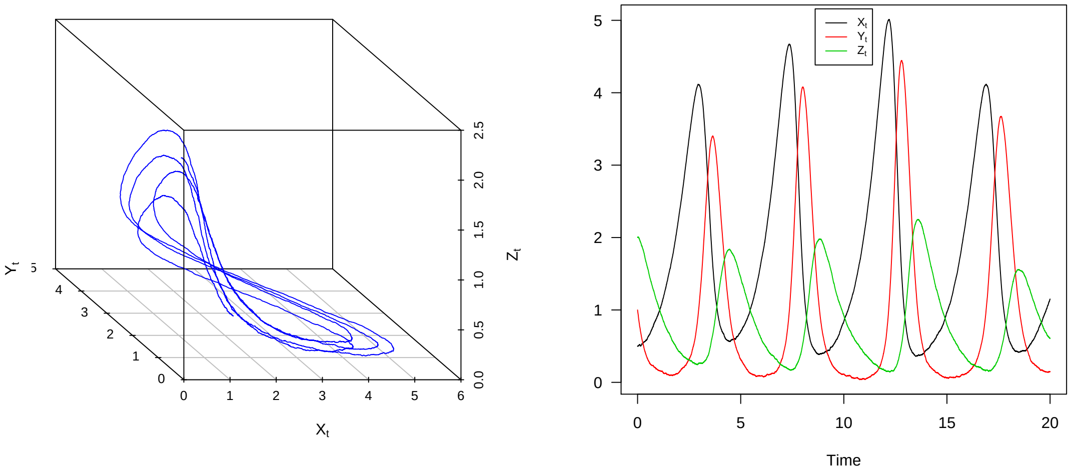

Figure 10: A trajectory in xyz-space (Left). A solution (X_t, Y_t, Z_t) with initial conditions (0.5,1,2) (Right). (The case: $a = b = c = d = e = f = 1$ and $\sigma = 0.03$)

2.3.4 Transformation of a SDE one dimensional

Next is an example of one dimensional SDE driven by three independent Brownian motions ($W_{1,t}$,$W_{2,t}$,$W_{3,t}$), as follows:

$$dX_t = \mu W_{1,t}dt + \sigma W_{2,t}dW_{3,t} \tag{26}$$

To simulate the solution of the equation (26) we make a transformation to a system of three equations as follows:

$$\begin{aligned} dX_t &= \mu Y_t dt + \sigma Z_t dW_{3,t} \\ dY_t &= dW_{1,t} \\ dZ_t &= dW_{2,t} \end{aligned} \tag{27}$$

run by calling the function `"snssde3d"` to produce a simulation of the solution of (26), with $\mu = 2$ and $\sigma = 0.2$:

```
> fx <- expression(2*y)
> gx <- expression(0.2*z)
> fy <- expression(0)
> gy <- expression(1)
> fz <- expression(0)
> gz <- expression(1)
> modtra <- snssde3d(driftx=fx,diffx=gx,drifty=fy,diffy=gy,driftz=fz,diffz=gz)
> modtra
```

```
Ito Sde 3D:
        | dX(t) = 2 * Y(t) * dt + 0.2 * Z(t) * dW1(t)
        | dY(t) = 0 * dt + 1 * dW2(t)
        | dZ(t) = 0 * dt + 1 * dW3(t)
Method:
        | Euler scheme of order 0.5
Summary:
        | Size of process        | N  = 1000.
        | Number of simulation   | M  = 1.
        | Initial values         | (x0,y0,z0) = (0,0,0).
        | Time of process        | t in [0,1].
        | Discretization         | Dt = 0.001.
```

the following code produces the result in the Figure 11,

```
> plot(modtra$X,plot.type="single",ylab="X")
```

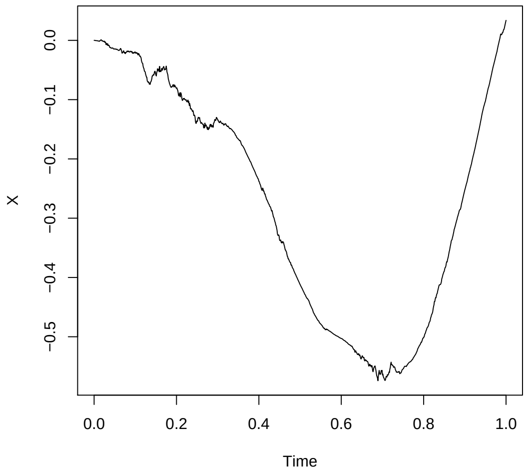

Figure 11: Simulate path of: $dX_t = 2W_{1,t}dt + 0.2W_{2,t}dW_{3,t}$ used `snssde3d` function.

3 Itô vs Stratonovich: How to choose?

- White noise is an idealisation; real fluctuating forcing has finite amplitude and timescale.
- If white noise is approximation to continuously fluctuating noise with finite memory (much shorter than dynamical timescales), appropriate representation is Stratonovich.
- If white noise approximates set of discrete pulses with finite separation to which system responds, or SDE continuous approximation to discrete system, then Itô representation appropriate.
- Because in an atmosphere/ocean/climate context "driving noise" a representation of "fast" part of continuous fluid dynamical system, Stratonovich SDEs usually most natural. For example, consider $2D$ SDEs:

$$\begin{aligned}\frac{dX_t}{dt} &= a(t, X_t) + b(t, X_t)\eta \\ \frac{dX_t}{dt} &= -\frac{1}{\tau}\eta + \frac{\sigma}{\tau}\dot{W}\end{aligned}$$

  as $\tau \to 0$, $\eta \to \dot{W}$ and X_t satisfies the Stratonovich SDE.
- Operationally: Stratonovich SDE's easier to solve analytically, but Itô SDE's more natural starting point for numerical schemes.
- Chief usages:
  - Stratonovich SDEs: Physics and engineering.
  - Itô SDEs: Mathematical analysis and financial mathematics.

4 Summary

This work is about ready to be used `Sim.DiffProc` package for simulation of stochastic differential equations and some related estimation methods based on discrete sampled observations from such models. We hope that the package presented here and the updated survey on the subject might be of help for practitioners, postgraduate and PhD students, and researchers in the field who might want to implement new methods and ideas using R as a statistical environment. The simulation studies implemented in R language seem very preferment and efficient, because it is a statistical environment, which permits to realize, to visualize and validate the simulations.

References

Friedman, A. (1975). *Stochastic differential equations and applications.* Vol.1, Academic Press.

Soong, T.T. (1973). *Random differential equations in science and engineering.* Academic Press, New York. Lc Number: QA274.23.S58.

Henderson, D. and Plaschko,P. (2006). *Stochastic differential equations in science and engineering.* World Scientific.

Itô, K. (1944). Stochastic integral. Tokyo, *Proc. Jap. Acad*, 20, pp. 519–529.

Allen, E. (2007). *Modeling with Itô stochastic differential equations.* Springer-Verlag.

Klebaner, F.C. (2005). *Introduction to Stochastic Calculus with Applications.* Imperial College Press

Kasdin, J. (1995). Runge-Kutta algorithm for the numerical integration of stochastic differential equations. *Journal of Guidance, Control, and Dynamics.* Vol.18, No.1, 114–120.

Andreas, R. (2003a). Embedded stochastic Runge-Kutta methods. *Proc. Appl. Math. Mech.*, 2(1), 461–462.

Andreas, R. (2003b). Coefficients of Runge-Kutta schemes for Itô stochastic differential equations. *Proc. Appl. Math. Mech.*, 3(1), 571–572.

Andreas, R. (2004). Runge-Kutta methods for Stratonovich stochastic differential equation systems with commutative noise. *J. Comput. Appl. Math.*, 164–165, 613–627.

Andreas, R. (2007). Second order Runge-Kutta methods for Stratonovich stochastic differential equations. *BIT Numerical Mathematics.* Vol.47, No.3, 657–680.

Andreas, R. (2010). Runge-Kutta methods for the strong approximation of solutions of stochastic differential equations. *SIAM J. Numer. Anal.*, Vol.48, No.3, 922–952.

Racicot, F.É, and Théoret, R. (2006). *Finance Computationnelle et Gestion des Risques.* Presses de l'Université du Québec.

Boukhetala, K. (1996). Modelling and simulation of a dispersion pollutant with attractive centre. ed by Computational Mechanics Publications, Southampton ,U.K and Computational Mechanics Inc, Boston, USA, 245–252.

Jedrzejewski, F. (2009). *Modeles aleatoires et physique probabiliste.* Springer-Verlag.

Stefano, M.I. (2008). *Simulation and inference for stochastic differential equations: with R examples.* Springer-Verlag, New York.

Stefano, M.I. (2015). `sde`: Simulation and Inference for Stochastic Differential Equations. *R package version 2.0.14*. `http://CRAN.R-project.org/package=sde`

Stefano, M.I. (2011). *Option Pricing and Estimation of Financial Models with R.* John Wiley & Sons Ltd.

Alexandre Brouste, Masaaki Fukasawa, Hideitsu Hino, Stefano M. Iacus, Kengo Kamatani, Yuta Koike, Hiroki Masuda, Ryosuke Nomura, Teppei Ogihara, Yasutaka Shimuzu, Masayuki Uchida, Nakahiro Yoshida. (2014). The `yuima` Project: A Computational Framework for Simulation and Inference of Stochastic Differential Equations. *Journal of Statistical Software*, 57(4).1–51. `http://www.jstatsoft.org/v57/i04/`.

Guidoum, A. C. and Boukhetala, K. (2015). `Sim.DiffProc`: Simulation of Diffusion Processes. R package version 3.1. `http://CRAN.R-project.org/package=Sim.DiffProc`

Guidoum, A. C. and Boukhetala, K. (2014). A new R package for Itô and Stratonovich stochastic calculus. In submission (Computational Statistics, Springer).

Kloeden, P.E, and Platen, E. (1989). A survey of numerical methods for stochastic differential equations. *Stochastic Hydrology and Hydraulics*, 3, 155–178.

Kloeden, P.E, and Platen, E. (1991a). Relations between multiple Itô and stratonovich integrals. *Stochastic Analysis and Applications*, 9(3), 311–321.

Kloeden, P.E, and Platen, E. (1991b). Stratonovich and Itô stochastic taylor expansions. *Mathematische Nachrichten*, 151, 33–50.

Kloeden, P.E, and Platen, E. (1995). *Numerical Solution of Stochastic Differential Equations.* Springer-Verlag, New York.

Kloeden, P.E, Platen, E. and Schurz, H. (1994). *Numerical solution of SDE through computer experiments.* Springer-Verlag, New York.

Daniel, A. and Duncan, M. (2015). rgl: 3D visualization device system (OpenGL). *R package version 0.95.1367.* http://CRAN.R-project.org/package=rgl

Uwe, L., Martin, M. and Sarah, S. (2015). scatterplot3d: 3D Scatter Plot. *R package version 0.3-36.* http://CRAN.R-project.org/package=scatterplot3d

Øksendal, B. (2000). *Stochastic Differential Equations: An Introduction with Applications.* 5th edn. Springer-Verlag, Berlin.

Rolski T, Schmidli H, Schmidt V and Teugels J. (1998). *Stochastic Processes for Insurance and Finance.* John Wiley & Sons.

Heinz, S. (2011). *Mathematical Modeling.* Stochastic Evolution, pp. 295–334, Springer-Verlag, Berlin Heidelberg.

Platen, E. (1980). Weak convergence of approximations of Itô integral equations. *Z Angew Math Mech.* 60, 609–614.

Platen, E. and Bruti-Liberati, N. (2010). *Numerical Solution of Stochastic Differential Equations with Jumps in Finance.* Springer-Verlag, New York

Saito, Y, and Mitsui, T. (1993). Simulation of Stochastic Differential Equations. *The Annals of the Institute of Statistical Mathematics*, 3, 419–432.

Hofbauer, J. and J. W. H. So. (1994). Multiple limit cycles for three dimensional Lotka-Volterra equations. *Applied Mathematics Letters.* 7(6).59–63.

R Development Core Team (2015). *R: A Language and Environment for Statistical Computing.* Vienna, Austria. http://www.R-project.org/

Hadeler, K. de Mottoni, P. and Schumacher, K. (1980). Dynamic Models for Animal Orientation. *The Journal of Mathematical Biology*, 10, 307–332.

Helland, S. (1983). Diffusion models for the dispersal of insects near an attractive center. *The Journal of Mathematical Biology*, 18, 103–122.

Volterra, V. (1926). Variazioni e fluttuazioni del numero d'individui in specie animali conviventi. *Mem. R. Accad. Naz.dei Lincei,* Ser. V, vol. 2.

Lotka, A. J. (1925). Elements of Physical Biology. *Williams & Wilkins Co., Baltimore.*

Heemink, A. (1990). Stochastic Modelling of Dispersion in Shallow Water. *Stochastic Hydrology and Hydraulics*, 4, 161–174.

Erica E., Joseph E.P., Joseph, P.P. and Zac W. (2002). A Lotka Volterra Three Species Food Chain. *Matematics Magazine*, 75(4), 243–255. http://math.bd.psu.edu/faculty/jprevite/mathmag243-255.pdf

Leung, H. K. (1995). Stochastic transient of a noisy van der Pol oscillator. *Physica A: Statistical Mechanics and its Applications*, 221(1–3), 340–347.

Van der Pol, B. (1922). *Philosophical Magazine.* 43–177.

Naess, A. and Hegstad, B. K. (1994). Response Statistics of Van der Pol Oscillators Excited by White Noise. *Nonlinear Dynamies.* 5, 287–297.

Zhujun, J., Zhiyan, Y. and Tao, J. (2006). Complex dynamics in Duffing–Van der Pol equation. *Chaos, Solitons and Fractals.* 27, 722–747.