A stochastic SIR model

Nicola Dal Cin
dalcin.nicola@spes.uniud.it

February 2, 2023

Abstract

In this project we consider a stochastic SIR model proposed in [1] to study a pandemic evolution throughout a system of SDEs. This model was later studied by [2], who introduced a suitable reproduction number $\tilde{R}_0$ and found conditions to determine the behaviour of a solution with certain initial values. In particular, we are interested in conditions who leads to the *extinction* or the *persistence* of the plague.

The goal of the present project was to implement a program to solve numerically the above system of SDEs, and reproduce the different scenarios that may occur, depending on system's parameters, with numerical simulations. This was done mainly using the function `snssde2d()` of the library `Sim.DiffProc` implemented in R.

1 Introduction

Modelling the evolution of a biological phenomenon, like the spread of a pandemic injection, is a theme of deep study and great interest, in particular in these years. Many attempts and different choices can be made to tackle the problem, main examples are the traditional deterministic ODEs models [3], but there are also implementations accounting various delays that occurs in such a system, and therefore use DDEs [4].

Another approach can be the formulation of a non–deterministic model, there are few examples of statistical and probabilistic ones, using for examples Markov Chains [5]. In this small project we try to follow this path, and address the problem of modelling the evolution of a pandemic throughout a system of Stochastic Differential Equations (SDEs). The choice of such a system is not the most common for this kind of problems, and definitely more adopted in other fields (e.g. financial mathematics [6], [7]).

2 Model and Simulated Examples

The model we are going to consider is a stochastic SIR, proposed in [1] as a reformulation of the following deterministic one

$$\begin{cases} \dot{S}(t) = \Lambda - \mu S(t) - \beta S(t) I(t), \\ \dot{I}(t) = \beta S(t) I(t) - (\mu + \gamma + \epsilon) I(t), \\ \dot{R}(t) = \gamma I(t) - \mu R(t); \end{cases} \tag{1}$$

which, we recall[1], assumes the population divided into the following partition: the susceptible S, the infective I, and recovered R. The other parameters mentioned are: the birth rate Λ, the average number of contacts per infective per day β, the recovery rate γ, and the death rate of the infectives caused by the disease, while the death rate μ due to other causes.

In the study of this system dynamics, we are mainly interested in determining whether the solution reaches an equilibrium (a zero of the vector field) and, in that case, to determine the behaviour of solutions in its neighborhood (e.g. if it is stable or unstable).

[1] We remand to [3] for a precise review of these models.

In (1) there is the disease–free equilibrium $E_0 := (\Lambda/\mu, 0, 0)$, this equilibrium is the most relevant, or at least desirable, and one of the most useful parameter to classify it is the *basic reproduction number*, which in the case of (1) is $R_0 = \frac{\beta\Lambda}{\mu(\mu+\gamma+\epsilon)}$.

Theorem 2.1. In system (1) the disease–free equilibrium $E_0 := (\Lambda/\mu, 0, 0)$ is globally asymptotically stable if $R_0 \leq 1$. Whereas when $R_0 > 1$, E_0 is unstable and there is an endemic equilibrium

$$E^* := (S^*, I^*, R^*), \text{ with } S^* := \frac{\Lambda}{\mu R_0}, \ I^* = \frac{\mu}{\beta}(R_0 - 1), \ R^* = \frac{\gamma}{\beta}(R_0 - 1) \tag{2}$$

which is a global attractor.

Proof. See [8]. □

As we will see later, determining an analogue of R_0 in the context of SDEs systems is a much more difficult task than in the deterministic case.

2.1 System equations and results

The SDEs system we will consider is obtained adding in (1) a diffusion term (depending on a variance σ) as follows

$$\begin{cases} dS(t) = [\Lambda - \mu S(t) - \beta S(t)I(t)]\,dt - \sigma S(t)I(t)\,dW(t), \\ dI(t) = [\beta S(t)I(t) - (\mu + \gamma + \epsilon)I(t)]\,dt + \sigma S(t)I(t)\,dW(t), \\ dR(t) = [\gamma I(t) - \mu R(t)]\,dt. \end{cases} \tag{3}$$

We remark that the dynamic of R has no effects on the transmission dynamics, and the third equation can be therefore ignored only considering

$$\begin{cases} dS(t) = [\Lambda - \mu S(t) - \beta S(t)I(t)]\,dt - \sigma S(t)I(t)\,dW(t), \\ dI(t) = [\beta S(t)I(t) - (\mu + \gamma + \epsilon)I(t)]\,dt + \sigma S(t)I(t)\,dW(t), \end{cases} \tag{4}$$

For the above, as showed in [29], there is the existence of a positive solution $(S(t), I(t))$ for $t \geq 0$, and for any initial value $(S(0), I(0)) \in \mathbb{R}^2_{>0}$. Moreover, they showed that

$$\Gamma := \{(S(t), I(t)) \in \mathbb{R}^2_{>0} \,|\, S(t) + I(t) \leq \Lambda/\mu, \text{ a.e. } t \geq 0\} \tag{5}$$

is an invariant set.

In the next sections we enunciate some definitions and results on the stability of this model, that the reader can find in [2]. Then we will focus on the main part: description of the implementation in R and of its results.

2.2 Extinction

As seen earlier, the definition of a suitable reproduction parameter helps in determine whether the disease–free equilibrium $(\Lambda/\mu, 0, 0)$ can be reached and maintained stably. Such a definition, in a stochastic model as ours, is usually hard to find. In [2] the parameter $\tilde{R}_0$ is chosen as

$$\tilde{R}_0 = R_0 - \frac{\sigma^2\Lambda^2}{2\mu^2(\mu + \gamma + \epsilon)}, \tag{6}$$

since they prove that if $\tilde{R}_0 < 1$ and the noise is not too large then the disease will die out; the same situation occurs even if the noise. In particular holds

Theorem 2.2. Let $(S(t), I(t))$ be the solution of (4), with initial value $(S(0), I(0)) \in \Gamma$. If one of the following holds:

$$\tilde{R}_0 < 1, \text{ and } \sigma^2 \leq \frac{\mu\beta}{\Lambda}, \tag{7}$$

or

$$\sigma^2 > \max\left\{\frac{\mu\beta}{\Lambda}, \frac{\beta^2}{2(\mu+\gamma+\epsilon)}\right\}, \tag{8}$$

then the disease will die out with probability one, in fact

$$\lim_{t\to+\infty} S(t) = \frac{\Lambda}{\mu} \quad \text{and} \quad \lim_{t\to\infty} I(t) = 0 \;\; a.s. \tag{9}$$

Proof. Theorem 2.2 in [2]. □

Example 2.1. It is interesting to simulate the system solving numerically system (4). Here is one of my attempts (see section 3 for the implementation) for the model, in the extintion scenario. The parameters are chosen: $\Lambda = 1, \mu = 1, \beta = 2, \gamma = 0.8, \epsilon = 0.2$, and the evolution time is considered from $t_0 = 0$ to $T = 10$.

Solutions have initial value $(S(0), I(0)) := (0.8, 0.19)$ which is in Γ since $\Lambda/\mu = 1$. Deviation standard is taken $\sigma = 0.5$, and therefore variance is $\sigma^2 = 0.25$. The reproduction parameter and the two discriminants are respectively

$$R_0 = 0.9375 < 1, \quad \delta_1 := \frac{\mu\beta}{\Lambda} = 2, \quad \delta_2 = \frac{\beta^2}{2(\mu+\gamma+\epsilon)} = 1, \tag{10}$$

and we are so in the hypotheses of theorem 2.2. This is coherent with the simulation as we show below:

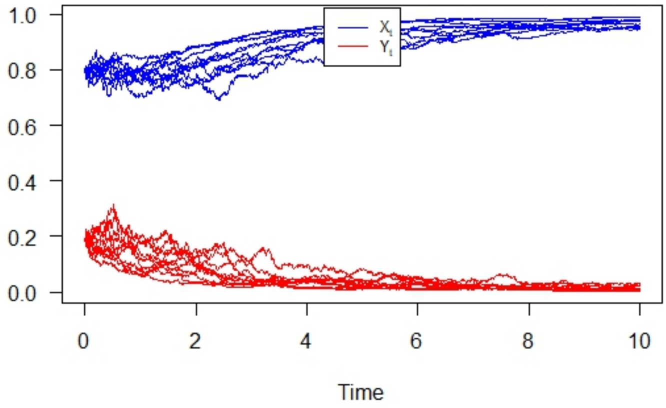


Figure 1: Plot of 10 trajectories generated with Monte–Carlo, and solved with Eulero–Maruyama method. Susceptible are plotted in blue while the Infective in red.

Figure 2.1 suggest how eventually will be reached a disease–free equilibrium. In order to better highlight this behaviour, we can run the program generating $M = 1000$ trajectories and analyzing the probability densities (following a Monte–Carlo approach). In particular, we show the snapshots of the data at time $t_1 := 2$ and $t_2 := 7$.

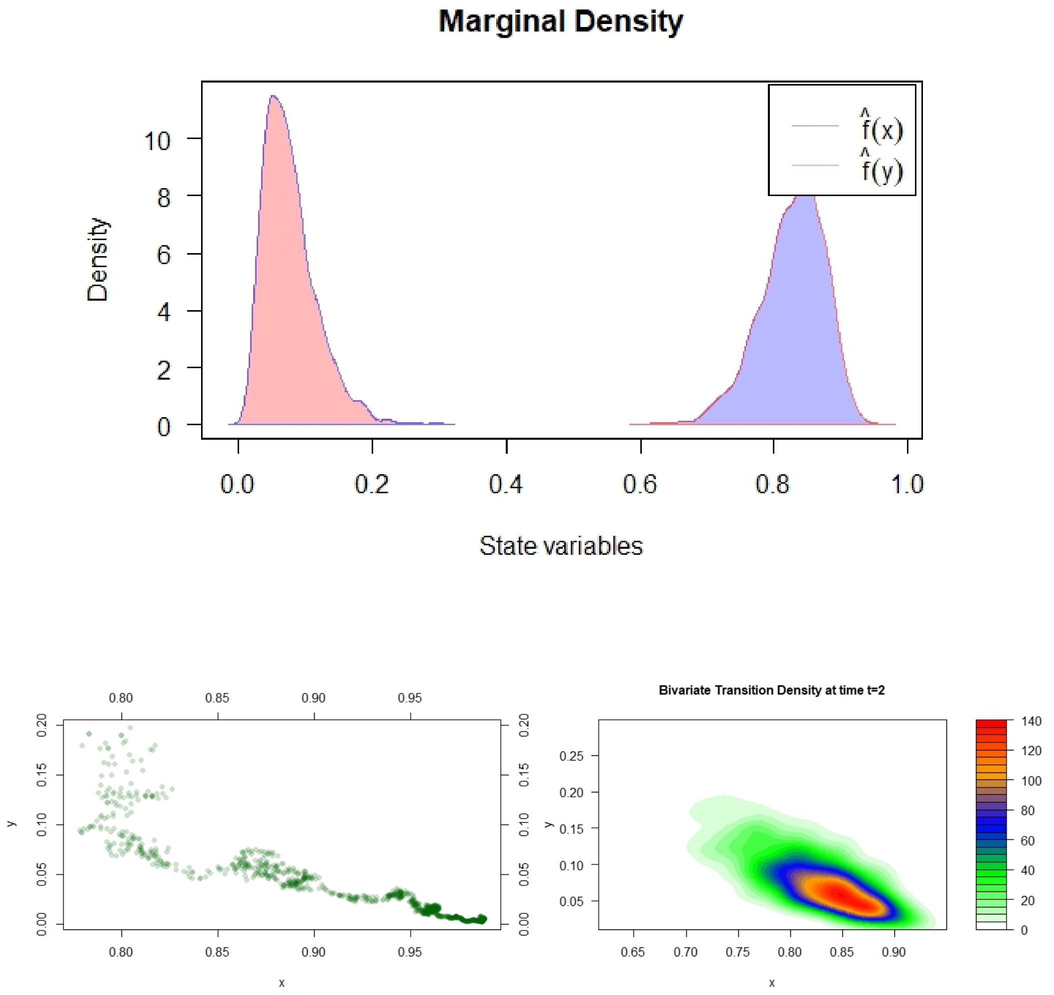


Figure 2: The marginal density, representing Susceptible and Infective probability distributions respectively in red and blue, reflects the behaviour predicted. Below, other representations of the simulated data (always in case $M = 1000$ and time $t_1 := 2$).

From figure 2.1 we see that the marginal densities are approaching the expected limit behaviour, even if the plague is still not extinct a.s. Looking at figure 2.1 and we expect this will eventually happen, which is actually the case as we can check in figure 2.1 by the plots made at $t_2 := 7$.

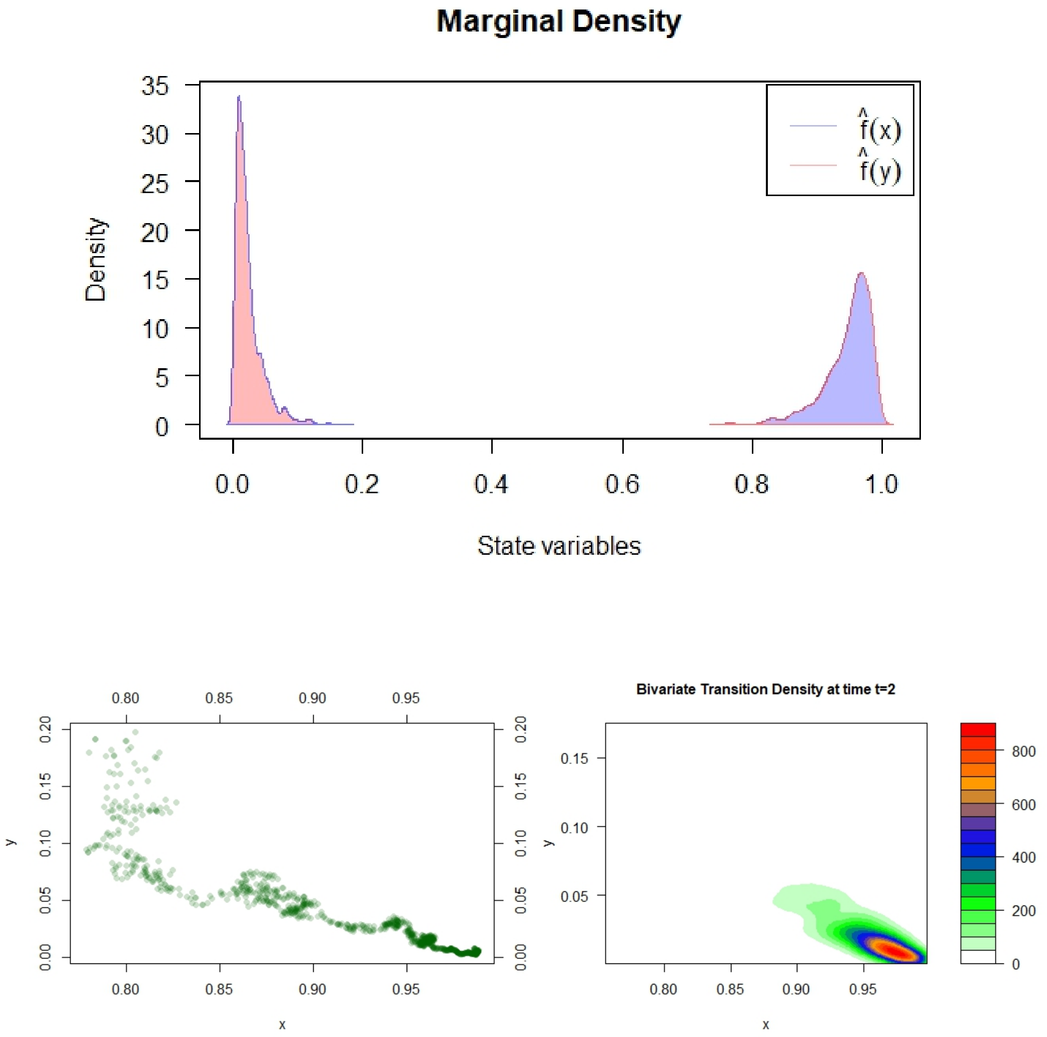


Figure 3: Same graphs of Figure 2.1, but at time $t_2 := 7$, we can see that how expected the marginal density of $I(t)$ has a higher and higher spike in zero as t grows, i.e. the disease will die out.

2.3 Persistence

It has also been proved that the case not considered in theorem 2.2, correspond to the scenario of having both infected and susceptible population persistent in time.

Theorem 2.3. Let $(S(t), I(t))$ be the solution of (4), with initial value $(S(0), I(0)) \in \Gamma$. If $\tilde{R}_0 > 1$ and $\sigma^2 < \mu\beta/\Lambda$ then

$$\frac{\mu(\tilde{R}_0 - 1)}{\beta} \leq \liminf_{t\to+\infty} \frac{1}{t}\int_0^t I(s)\,ds \leq \limsup_{t\to+\infty} \frac{1}{t}\int_0^t I(s)\,ds \leq \frac{\mu(\tilde{R}_0 - 1)}{\beta - \frac{\sigma^2\Lambda}{\mu}} \quad a.s. \tag{11}$$

and

$$\frac{\Lambda}{\mu} - \frac{(\mu+\gamma+\epsilon)(\tilde{R}_0 - 1)}{\beta - \frac{\sigma^3\Lambda}{\mu}} \leq \liminf_{t\to+\infty} \frac{1}{t}\int_0^t S(s)\,ds \leq \tag{12}$$

$$\leq \limsup_{t\to+\infty} \frac{1}{t}\int_0^t S(s)\,ds \leq \frac{\Lambda}{\mu} - \frac{(\mu+\gamma+\epsilon)(\tilde{R}_0 - 1)}{\beta} \quad a.s. \tag{13}$$

Proof. Theorem 3.1 in [2]. □

Example 2.2. Here is one of my simulations (see section 3 for the implementation) for the model in this scenario. The parameters are chosen: $\Lambda = 1, \mu = 1, \beta = 3, \gamma = 0.8, \epsilon = 0.2$, and the evolution time is considered from $t_0 = 0$ to $T = 10$.

Solutions have initial value $(S(0), I(0)) := (0.8, 0.19)$ which is in Γ since $\Lambda/\mu = 1$. Deviation standard is taken $\sigma = 0.5$, and therefore variance is $\sigma^2 = 0.25$. The reproduction parameter and the two discriminants are respectively

$$R_0 = 1.4375, \quad \delta_1 := \frac{\mu\beta}{\Lambda} = 3, \quad \delta_2 = \frac{\beta^2}{2(\mu+\gamma+\epsilon)} = 2.25, \tag{14}$$

and we are so in the hypotheses of theorem 2.3. This is coherent with the simulation as we show below:

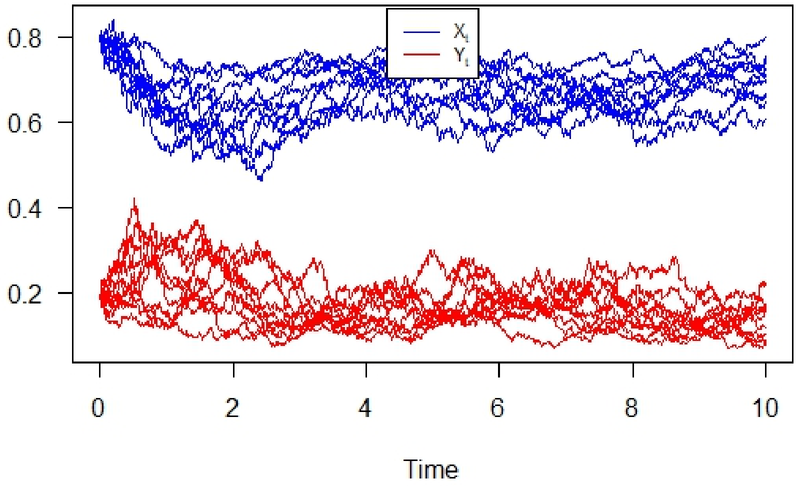


Figure 4: Plot of 10 trajectories generated with Monte–Carlo, and solved with Eulero–Maruyama method. Susceptible are plotted in blue while the Infective in red.

Even from this simple plot, we see how the disease seem to persist in time. To verify and see better this result, running the program with $M = 1000$ trajectories, gives the following distributions (here plotted at time $t_1 := 2$).

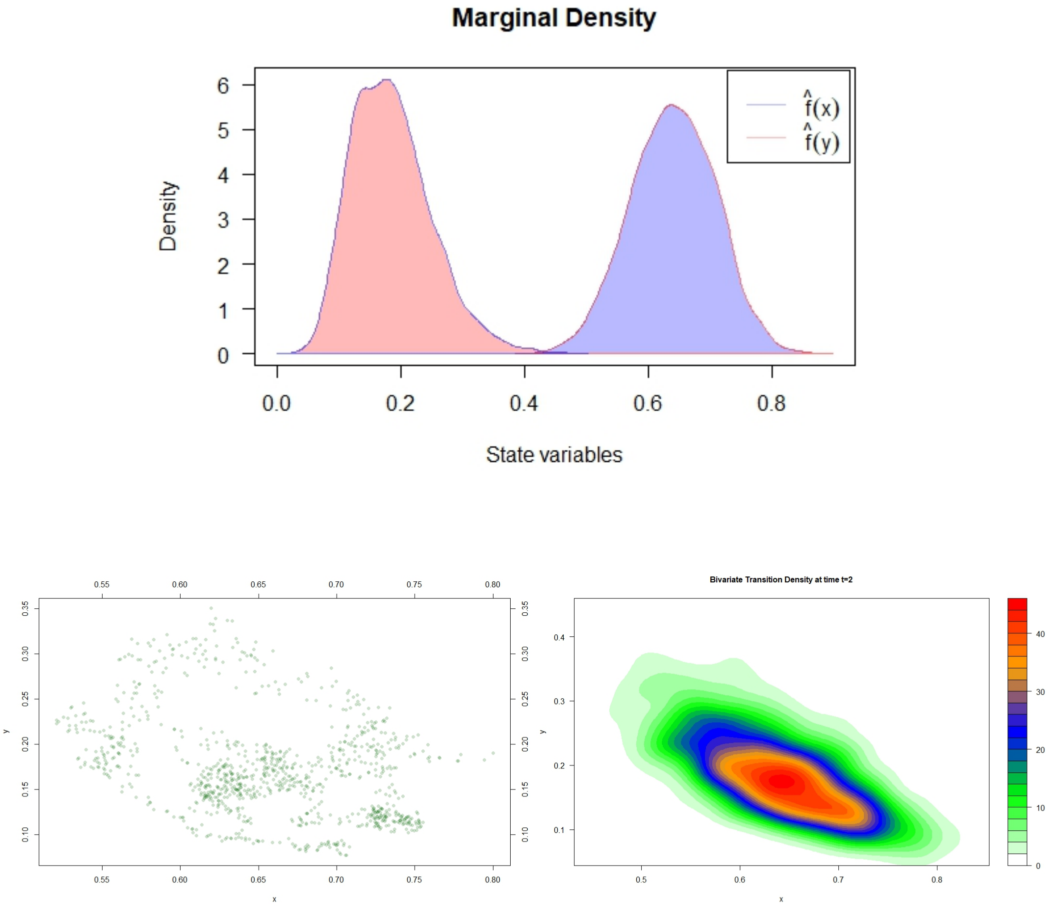


Figure 5: The marginal density, representing Susceptible and Infective probability distributions respectively in red and blue, reflects the behaviour predicted. Below, other representations of the simulated data at $t = t_1$ and in case $M = 1000$.

Remark 2.1. It is very interesting to notice how a simple change in the parameter β (indicating the average number of contacts per infective per day) from 2 to 3, leads to a completely different scenario. This highlights the importance of these studies to help taking strategic decisions, as we all experienced e.g. during COVID–19 pandemic. Another important topic of research can be therefore the analysis of different types of controls on the system parameters.

3 Implementation

In this section we will focus on some details regarding the implementation of both the stochastic model (4) and its deterministic counterpart (1). It is indeed interesting to compare the results of the two, also to estimate the (eventual) convergence of methods applied to solve (4).

3.1 Stochastic Model

As said before, to implement stochastic model (4) I used the software R. In particular, to solve system (4), the function `snssde2d()` of the library `Sim.DiffProc` was chosen. This function allows, indeed, to solve numerically SDEs of Itô or Stratonovich type (see appendix A) using different methods such as Euler–Maruyama (EM) or Milstein [9].

Remark 3.1. In the implementation, the variables $S(t)$ and $I(t)$ defined in the model are indicated respectively by $X(t)$, and $Y(t)$. This choice was made for the sake of consistency with the function's syntax, and not to confuse them with the deterministic solutions of (1).

Here is an example of the usage and the arguments of `snssde2d()` taken from `Sim.DiffProc`'s documentation [10].

```
snssde2d(N = 1000, M = 1, x0 = c(0,0),t0 = 0, T = 1, Dt, drift, diffusion,
corr = NULL, type = c("ito", "str"), alpha = 0.5, mu = 0.5, method = c("euler",
"milstein", "predcorr", "smilstein", "taylor", "heun", "rk1", "rk2","rk3"),...)
```

Referring for any deepening to the bibliography, we just point out that

- `N` ← number of simulation steps;
- `M` ← number of trajectories considered using a Monte–Carlo method;
- `x0` ← initial values $(X(0), Y(0))$ of the process;
- `t0, T` ← initial time and ending time;
- `Dt` ← number of simulation steps (default: $\Delta t = (T - t_0)/N$);
- `drift` ← drift coefficient, i.e. an expression (function) of three variables t, x, and y. Each component is the drift term of the correspondent SDE;
- `diffusion` ← diffusion coefficient, i.e. an expression (function) of three variables t, x, and y. Each component is the diffusion term of the correspondent SDE;
- `corr` ← correlation structure of two Brownian motions $W_1(t)$ and $W_2(t)$. It must be a real, symmetric, positive–definite, square matrix of dimension two[2];
- `method` ← numerical methods of simulation, default is `method = "euler"`.

```r
# Solving 2D SIR stochastic differential equation
library(Sim.DiffProc)

set.seed(1234, kind = "L’Ecuyer-CMRG")

# EXAMPLE 1: Disease-free equilibrium
# mu=1; sigma=0.5; gam=0.8; bet=2; lambda=1 ; epsilon=0.2; t_1=2; t_2=7

# EXAMPLE 2: Persistence equilibrium
mu=1; sigma=0.5; gam=0.8; bet=3; lambda=1; epsilon=0.2; t_1=2; t_2=7

x0 = 0.8; y0 = 0.19 # Initial values

### STOCHASTIC MODEL ###
f <- expression((lambda +(-bet)*x*y-mu*x),((bet)*x*y-(mu+gam+epsilon)*y))
g <- expression((-sigma*x*y),(sigma*x*y))
mod2d <- snssde2d(N= 1000, M=100, x0=c(x0,y0), t0=0, T=10, drift=f, diffusion=g,
                method="euler")

## Parameters defined in the paper and mentioned in stability theorems
R0 = bet*lambda/(mu*(mu+gam+epsilon))
stoc_R0 = R0-(sigma*lambda)**2/(2*mu**2*(mu+gam+epsilon))
var = sigma**2
discrim1 = mu*bet/lambda
discrim2 = bet**2/(2*(mu+gam+epsilon))
```

[2] I tried to consider different correlation matrices; however, since no significant difference was observed in the results, I decided to stick with the default identity matrix.

3.2 Deterministic Model

To solve system (1) was used the function `ode()` of the library `deSolve`. An example of its usage is the following:

```
ode(y, times, func, parms, method = "euler")
```

where

- `y` ← initial value of the solution;
- `times` ← time sequence for which output is wanted; its first must be the initial time.
- `func` ← either an R-function that computes the values of the derivatives in the ODE system (i.e, the model vector field f) at time t, or a character string giving the name of a compiled function in a dynamically loaded shared library.
- `params` ← parameters passed to `func`.
- `method` the integrator to use, either a function that performs integration, or a list of class `rkMethod`, or a string.

The method used in the implementation was the default one, i.e. `"Isoda"` while, to be consistent with the above method it was used the same initial values and mesh defined by $t_0 = 0$, $T = 10$, and[3] $\delta t = 1/1000$.

```
library(tidyverse)
library(deSolve)
### Deterministic SIR model ###
ode.sir.model <- function (t, x, parms) {
  ## first extract the state variables
  S <- x[1]
  I <- x[2]
  R <- x[3]
  mu <- parms[1]
  gam <- parms[2]
  bet <- parms[3]
  lambda <- parms[4]
  epsilon <- parms[5]
  ## now code the model equations
  dSdt <- lambda - mu*S - bet*S*I
  dIdt <- bet*S*I - (mu+gam+epsilon)*I
  dRdt <- gam*I - mu*R
  ## combine results into a single vector
  dxdt <- c(dSdt,dIdt,dRdt)
  ## return result as a list
  list(dxdt)
}

## ode in time
times <- seq(from=0,to=10,by=0.01) # mesh
xstart <- c(S=x0,I=y0,R=(1-x0-y0)) # initial values
parms <- c(mu, gam, bet, lambda, epsilon) # parameters

ode(func=ode.sir.model, y=xstart, times=times,
    parms = parms) %>% as.data.frame() -> sir_ode
```

[3] This precision in the ode time step is probably not even needed but, since the time complexity of the code is relatively low and the method converges, we chose it anyway. Other numerical experiments were done with $\delta t = 1/10, 1/100$ and the strong errors (see below) were very similar (since mainly due to the stochastic model).

4 Comparison with deterministic model and stability

To determine the convergence of the method adopted to solve a SDE, one can consider (see [11]) the *strong error*

$$e_{\Delta t}^{\text{strong}}(X) := \sup_{t_n \in \mathcal{I}_{\Delta t}} \mathbb{E}[|X_n - X(t_n)|], \tag{15}$$

where $X(t_n)$ is the exact solution at time $t_n := n\Delta t$ and X_n is the approximated solution of a one dimensional SDE.

Not being able to compute (15) analytically, I performed a Monte–Carlo estimation considering in place of $X(t_n)$ the approximated solution of the deterministic model (1) $\tilde{X}(t_n)$ (computed as described in the above section). In particular, as we see in the code below, I computed the following quantity

$$\tilde{e}_{\Delta t}(X) = M^{-1} \max_{t_n \in \mathcal{I}_{\Delta t}} \sum_{m=1}^{M} |X_n^m - \tilde{X}(t_n)|, \tag{16}$$

where X_n^m is the m-th random path generated by the method for $1 \leq m \leq M$.

The value (16) can be interpreted as the *discrepancy* between the prediction made by the two models (assuming the convergence of the numerical method used).

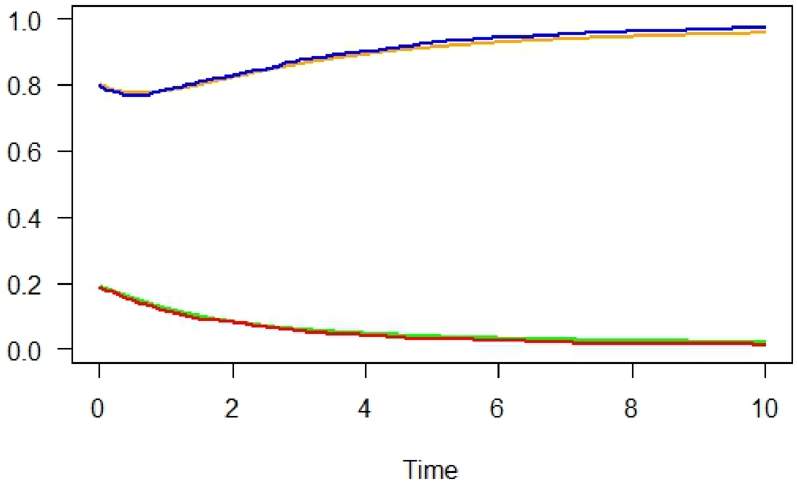
```r
### Convergence analysis ###
# "Discrepancy" on S(t)
S_mean_time_t <- vector(mode = "numeric", length = (N+1) )
RandS <- mod2d$X

for (n in 1:N+1) {
  errors <- vector(mode = "numeric", length = M )
  for (m in 0:M-1) {
    errors[m+1] <- abs(RandS[m*(N+1)+n] - sir_ode$S[n])
  }
  S_mean_time_t[n] <- mean(errors)
}
max(S_mean_time_t)

# "Discrepancy" on I(t)
I_mean_time_t <- vector(mode = "numeric", length = (N+1) )
RandI <- mod2d$Y

for (n in 1:N+1) {
  errors <- vector(mode = "numeric", length = M )
  for (m in 0:M-1) {
    errors[m+1] <- abs(RandI[m*(N+1)+n] - sir_ode$I[n])
  }
  I_mean_time_t[n] <- mean(errors)
}
max(I_mean_time_t)
```

4.1 Disease–free equilibrium example

Let us consider again example 2.1. Figure 6 suggests that the deterministic and stochastic models provide similar results in this case, as expected knowing that

$$R_0 = 1 \quad \text{and} \quad \tilde{R}_0 = 0.9375 \leq 1 \tag{17}$$

in this case.



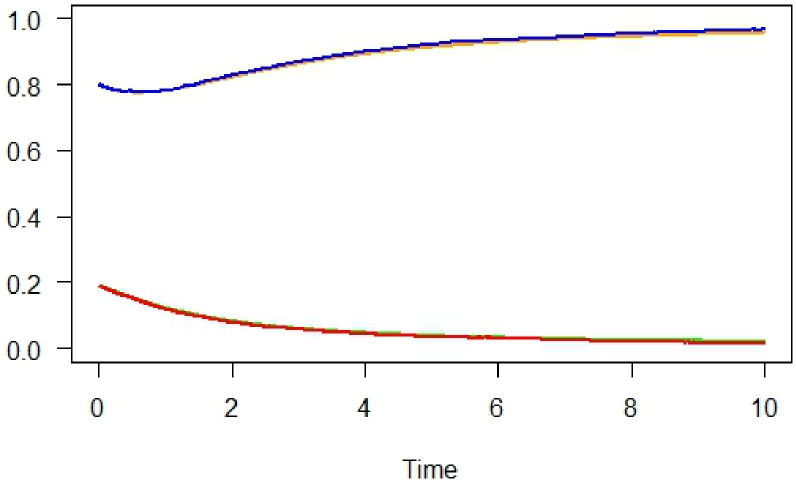


Figure 6: A comparison between ODE solutions and the mean of the M solutions for the SDEs' system (Euler–Maruyama was used). In blue and red we have respectively $E[X(t)]$ and $E[Y(t)]$, while in orange and green solutions $S(t)$ and $I(t)$ of (1). The top figure corresponds to the choice of $N = 100$, i.e. $\Delta t = 0.1$, and $M = 50$, while the plot below corresponds to $N = 1000$, and $M = 100$.

By the Figure 6, we can already (qualitatively) appreciate how the discrepancy between the two solutions is mainly due to the accumulation of the errors that leads to higher difference as t grows. In order to study the changing in the errors depending on different values of M and N, we performed some numerical simulations. As expected from *Central Limit Theorem* (CLT), values of M greater than 100 do not lead to remarkable differences (see Figure 7), and therefore later on we focus on the case $M = 100$, which is also convenient from a computational point of view.

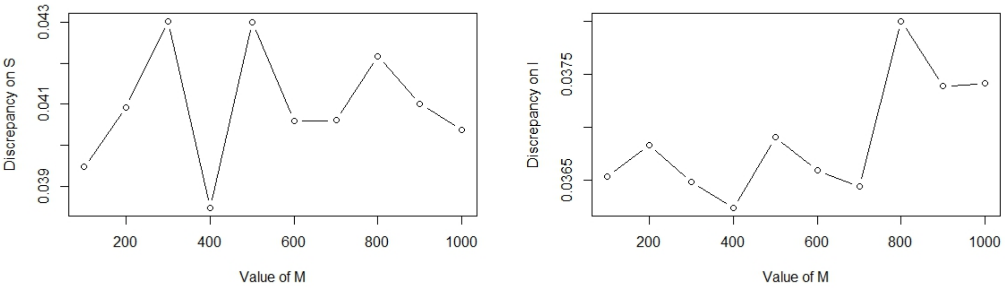


Figure 7: Susceptible and infected discrepancy $\tilde{e}_{0.01}$ computed using Euler–Maruyama and different values of M (and $N = 100$).

What is more interesting to analyze is the dimension of the mesh. As we recall in the appendix, EM has order of strong convergence 1/2; however, plots in figure 8 show how the error's magnitude does not decrease going from $N = 100$ to $N = 10000$, i.e. from $\Delta t = 0.1$ to $\Delta t = 0.001$. This fact suggests that the problem is likely ill–conditioned, and if this is the case, there is not much sense in talking about algorithms' stability.

To confirm this hypothesis, it can be useful to consider a better algorithm, i.e. which has an higher order of convergence.

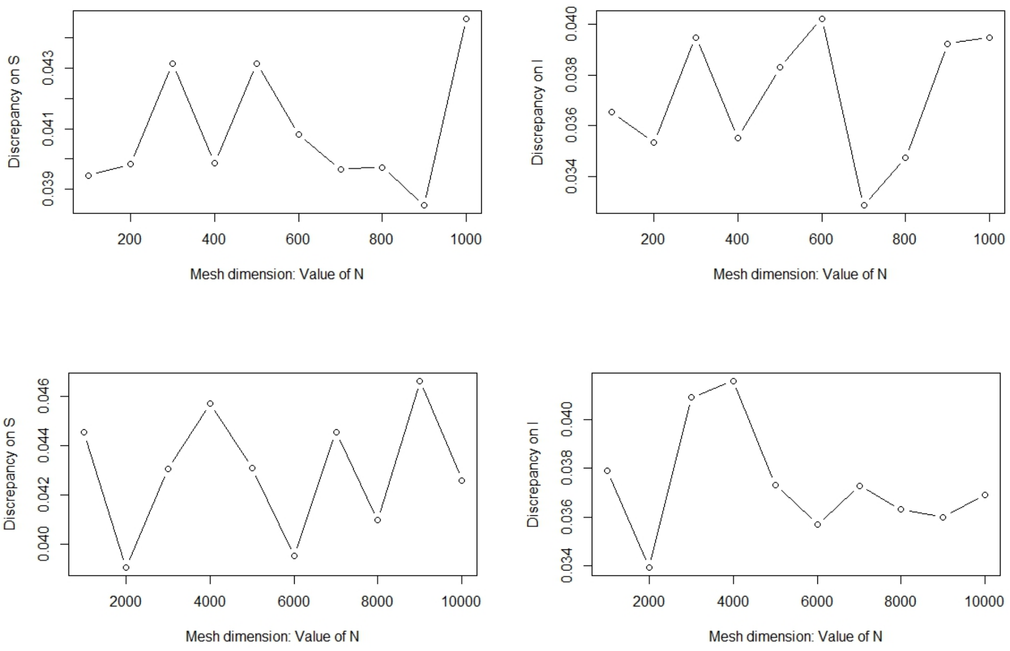


Figure 8: Discrepancy $\tilde{e}_{\Delta t}$ computed with EM respect to time steps $\Delta t = T/N = 10/N$.

As mentioned in the appendix, there are higher order methods such as Milstein's, which has an higher strong order of converge (equal to 1). Therefore, it is interesting to perform the previous numerical examples using Milstein method.

We obtain that the two algorithms (EM and Milstein) lead to really close results, even for high values of N (corresponding to $\Delta t \approx 0.001$). This can be due either to a large constant, as far as Milstein is concerned, bounding the strong error in equation (35), or (more likely) to an ill–conditioning of the problem.

Since the difference between the above two method is such small, we avoided plotting a figure as above, and we left the discrepancies $\tilde{e}_{\Delta t}(S)$ and $\tilde{e}_{\Delta t}(I)$ in the following table.

$\tilde{e}_{\Delta t}(S)$		
$N(10^3)$	EM	Milstein
1	0.04451	0.04440
2	0.03904	0.03909
3	0.04305	0.04300
4	0.04569	0.04565
5	0.04308	0.04304
6	0.03951	0.03954
7	0.04453	0.04447
8	0.04096	0.04100
9	0.04662	0.04658
10	0.04259	0.04264

$\tilde{e}_{\Delta t}(I)$		
$N(10^3)$	EM	Milstein
1	0.03789	0.03787
2	0.03392	0.03391
3	0.04092	0.04093
4	0.04160	0.04156
5	0.03732	0.03727
6	0.03570	0.03573
7	0.03730	0.03721
8	0.03631	0.03633
9	0.03600	0.03601
10	0.03691	0.03693

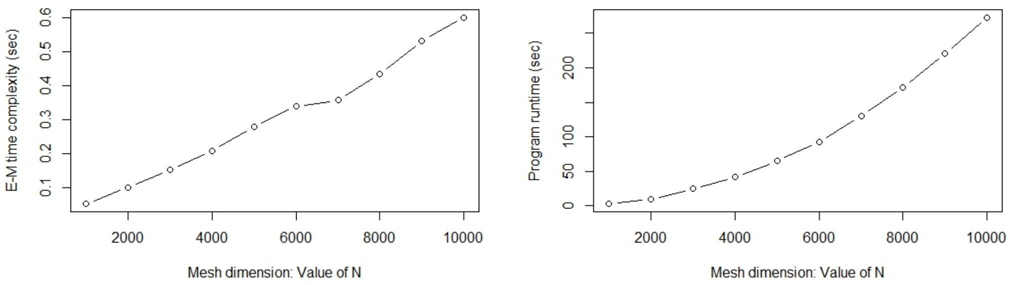


Figure 9: On the left hand side, the time elapsed to compute the numerical solution with EM; while on the right the total program run–time[5].

Remark 4.1. Considering a perturbation of this scenario, letting $\beta = 2.1$ (for which $R_0 = 1.05 > 1$ and $R_0 = 0.9875 < 1$) leads to higher discrepancies in the solution (one can check this using our code with e.g. $N = 100, \ldots, 1000$ and $M = 100$). Therefore, it can be interesting to compare solutions provided by ODE and SDE models using real data–set.

4.2 Persistence example

The same analysis and remarks can be performed for the other example 2.2. In this case, as we can see from figure 11, the discrepancies $\tilde{e}_{\Delta t}$ are even larger.

[5] Clearly the algorithm to compute the strong error I proposed in section 4 is not optimized for R, even if I pre–computed everything I could. In these high level programming languages is usually better to use in–built functions (e.g. the function `mean` of the class `"snssde2d"` was used to plot figure 6), but in this case I found none.

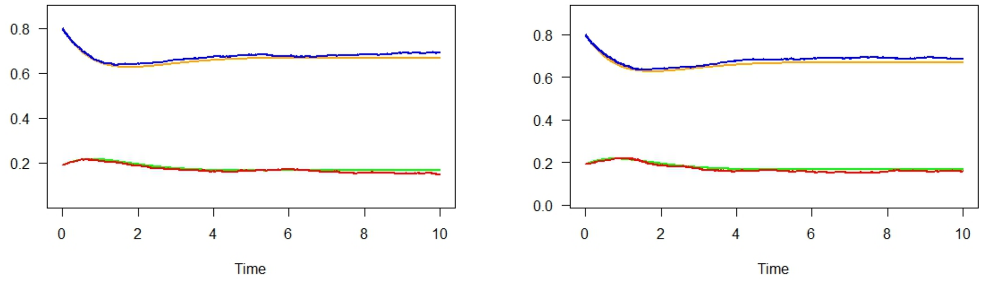


Figure 10: A comparison between ODE solutions and the mean of the M solutions for the SDEs' system (Euler–Maruyama was used). In blue and red we have respectively $E[X(t)]$ and $E[Y(t)]$, while in orange and green solutions $S(t)$ and $I(t)$ of (1). The figure on the left corresponds to the choice of $N = 1000$ and $M = 100$; while the plot on the right corresponds to $N = 10000$, and $M = 100$.

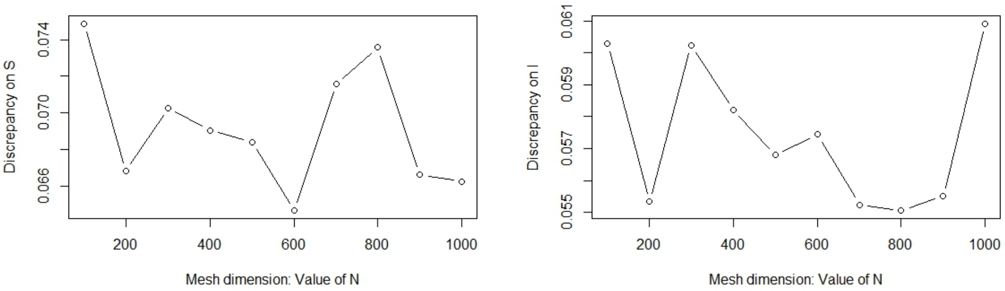


Figure 11: Discrepancy $\tilde{e}_{\Delta t}$ computed with EM respect to time steps $\Delta t = T/N = 10/N$.

5 Conclusions

We considered the stochastic SIR model proposed in [1], and implemented it in R mainly using the function `snssde2d()` of the library `Sim.DiffProc`. Simulations were performed to show the consistency of the numerical solutions with theoretical results proved in [2]; finally, results were compared with the numerical solutions of the deterministic SIR counterpart of (4) (these also to check empirically the method convergence).

In section 4 we showed empirically that the problem is likely to be ill–conditioned, and that there are quite significant discrepancy in results predicted by the two methods, in particular considering scenarios corresponding to parameters choices that are close to a bifurcation for a system (e.g. if $R_0 > 1$ and $\tilde{R}_0 < 1$). This remark maybe suggest how stochastic models can be studied to consider perturbations in the deterministic counterpart.

A Numerical methods of SDEs

In this appendix we briefly recall some definitions about stochastic differential equations and some relative numerical methods. For any detail we remand to the bibliography (e.g. [9]).

A.1 Basic introduction of SDEs

A stochastic differential equation is a differential equation in which one (or more) of the terms is a stochastic process, it follows that also its solution is therefore a random variable $X(t)$.

More intuitively, if we want to add a random forcing term to an ODE defined by a vector field $f : \mathbb{R}^d \to \mathbb{R}^d$ like

$$\begin{cases} \dot{x}(t) = f(x(t)), \quad t \in [0, T] \\ x(0) = x_0, \end{cases} \tag{18}$$

we can consider

$$\begin{cases} dX(t) = f(X(t))dt + g(X(t))dW(t) \quad t \in [0, T] \\ X(0) = X_0, \end{cases} \tag{19}$$

where f is called *drift*, while $g : \mathbb{R}^d \to \mathbb{R}^{d\times m}$ *diffusion* term, and $W(t)$ is an m–dimensional standard Wiener process (i.e. $W_0 = 0$, it is continuous with respect to t, and has independent Gaussian increments $W_{t+s} - W_t \sim \mathcal{N}(0, s)$).

Example A.1. The case we considered in (4) is a bit more articulated since we deal with a coupled system of two equations in which the drift and diffusion terms depend on both variables. If $f, g : \mathbb{R}^d \to \mathbb{R}^{d\times 2}$ we have a system of the form

$$\begin{cases} dX(t) = f_1(X(t), Y(t))\, dt + g_1(X(t), Y(t))\, dW(t) \\ dY(t) = f_2(X(t), Y(t))\, dt + g_2(X(t), Y(t))\, dW(t) \\ (X(0), Y(0)) = (X_0, Y_0). \end{cases} \tag{20}$$

It is known that under certain conditions (Cauchy–Lipschitz) of regularity (18) has existence and uniqueness of solutions, which is

$$x(t) = x_0 + \int_0^t f(x(s))ds, \quad t \in [0, T]. \tag{21}$$

The stochastic counterpart of the above is the integral equation

$$X(t) = X(0) + \int_0^t f(X(s)) + \int_0^t g(X(s))\, dW(s), \quad t \in [0, T], \tag{22}$$

where the last term is called *Itô integral.*

Numerically solving (19) is therefore equivalent to well compute (approximate) integrals in (22). In the next section we give some ideas on how this can be done.

A.2 Numerical integration

To compute Itô integrals we need to estimate the probability distribution induced by the Wiener process $W(t)$. To do so we can of course use the condition on the increments proceeding as follows:

(i) Partition $[0, T]$ into N sub–intervals of equal length $\Delta t = T/N$.

(ii) Knowing that $W(0) = 0$ and letting $t_n = n\Delta t$ we have

$$\Delta W_n := W(t_n) - W(t_{n-1}) \sim \mathcal{N}(0, \Delta t); \tag{23}$$

we recursively sample N values

$$W_n = W_{n-1} + \Delta W_n \quad n = 1, \ldots, N. \tag{24}$$

Now, for a given deterministic function $v : \mathbb{R} \to \mathbb{R}$ we can compute its integral in $dW(s)$ in different ways. The two main examples use the either the left–hand endpoint of the mesh or the midpoint.

A.2.1 Itô SDEs

Having a discretization mesh $0 = t_0 < t_1 < \cdots < t_n < \cdots < t_N = T$ we can choose the left–hand endpoint of each interval and define, for a deterministic $v : [0, T] \subset \mathbb{R} \to \mathbb{R}$

$$\int_0^T v(s)dW(s) = \lim_{N\to\infty} \sum_{n=0}^{N} v(t_n)(W(t_{n+1}) - W(t_n)), \tag{25}$$

this construction is in fact due to Itô. This version is widely adopted because of its *martingale property*, especially in mathematical finance where the *market hypothesis* fits the requisite evaluating v in the left endpoint (see [12] for more).

A downside in using the above Itô formulation is its peculiar chain rule; in fact the following expression (*Itô's formula*) holds

$$dv(X(t)) = \left(v'(X(t))f(X(t)) + \frac{1}{2}v''(X(t))g^2(X(t))\right)dt + v'(X(t))g(X(t))dW(t). \tag{26}$$

A.2.2 Stratonovich SDEs

Not to renounce to the usual chain rule, one can follow Stratonovich recipe and evaluating v in the midpoints of the mesh $0 = t_0 < t_1 < \cdots < t_N = T$. In this way one defines

$$\int_0^T v(s)dW(s) = \lim_{N\to\infty} \sum_{n=0}^{N} v\left(\frac{t_n + t_{n+1}}{2}\right)(W(t_{n+1}) - W(t_n)). \tag{27}$$

The above formula is usually referred to as *Stratonovich integral* and, beyond obeying to the usual chain rule, it has others good properties such as the preservation of certain systems' Hamiltonian.

There is no uniform consensus about which version between (25) and (27) is more relevant or useful; they are both very used in several fields. For the details of what was briefly recalled above we leave to e.g. [13].

A.3 Euler–Maruyama method

The sampled values (24) and (26) allows to approximate the expression

$$X(t) = X(t_n) + \int_{t_n}^{t} f(X(s))ds + \int_{t_n}^{t} g(X(s))dW(s), \quad t \geq t_n. \tag{28}$$

By Itô's formula we obtain the Itô–Taylor expansions

$$\begin{aligned} f(X(s)) = f(X(t_n)) &+ \int_{t_n}^{s} \left(f'(X(u))f(X(u)) + \frac{1}{2}f''(X(u))g^2(X(u))\right)du \\ &+ \int_{t_n}^{s} f'(X(u))g(X(u))dW(u) \end{aligned} \tag{29}$$

and

$$\begin{aligned} g(X(s)) = g(X(t_n)) &+ \int_{t_n}^{s} \left(g'(X(u))f(X(u)) + \frac{1}{2}g''(X(u))g^2(X(u))\right)du \\ &+ \int_{t_n}^{s} g'(X(u))g(X(u))dW(u) \end{aligned} \tag{30}$$

Approximating to the first order

$$f(X(s)) \approx f(X(t_n)), \quad \text{and} \quad g(X(s)) \approx g(X(t_n)), \tag{31}$$

and substituting into (28) with $t = t_{n+1}$, we get

$$X(t_{n+1}) \approx X(t_n) + f(X(t_n))\Delta t + g(X(t_n))(W_{n+1} - W_n). \tag{32}$$

Finally, if we define the numerical solution as $X_n :\approx X(t_n)$, we have

$$X_{n+1} \approx X_n + f(X_n)\Delta t + g(X_n)\Delta W_{n+1}. \tag{33}$$

A.3.1 Strong convergence

It is useful to consider the difference between the exact solution and the approximated one, at each point t_n. The idea is in fact to estimate the expectation value of $|X(t_n) - X_n|$.

We recall that the *strong error* is defined as

$$e_{\Delta t}^{\text{strong}} = \sup_{n=1,\dots,N} \mathbb{E}[|X_n - X(t_n)|], \tag{34}$$

and that a method is said *strongly convergent with order* p if exist $C, \Delta t^\star > 0$ such that

$$e_{\Delta t}^{\text{strong}} \leq C\Delta t^p, \quad \text{for all } 0 < \Delta t < \Delta t^\star. \tag{35}$$

We also recall (see [9]) that the Euler-Maruyama method previously described is strongly convergent with $p = 1/2$.

A.4 Milstein method

One can try to use an higher order approximation than (31) in order to have a better convergence. Keeping for example terms up to the second order in the Itô–Taylor expansion leads to the approximation

$$X_{n+1} \approx X_n + f(X_n)\Delta t + g(X_n)\Delta W_{n+1} + \frac{1}{2} g'(X_n)g(X_n)(\Delta W_{n+1}^2 - \Delta t). \tag{36}$$

The method using the above approximation step is named *Milstein* [9], and can be showed to have order of strong convergence equal to 1.

References

[1] E. Tornatore, S. Maria Buccellato, and P. Vetro, “Stability of a stochastic sir system,” *Physica A: Statistical Mechanics and its Applications*, vol. 354, pp. 111–126, 2005.

[2] C. Ji and D. Jiang, “Threshold behaviour of a stochastic sir model,” *Applied Mathematical Modelling*, vol. 38, no. 21, pp. 5067–5079, 2014.

[3] M. Iannelli and F. Milner, “The basic approach to age-structured population dynamics,” *Lecture Notes on Mathematical Modelling in the Life Sciences. Springer, Dordrecht*, 2017.

[4] T. Zhang, X. Meng, T. Zhang, and Y. Song, “Global dynamics for a new high-dimensional sir model with distributed delay,” *Applied Mathematics and Computation*, vol. 218, no. 24, pp. 11806–11819, 2012.

[5] A. Vivanco-Lira, “Predicting covid-19 distribution in mexico through a discrete and time-dependent markov chain and an sir-like model,” *arXiv preprint arXiv:2003.06758*, 2020.

[6] P. Glasserman, *Monte Carlo methods in financial engineering*, vol. 53. Springer, 2004.

[7] Y.-K. Kwok, *Mathematical models of financial derivatives.* Springer, 2008.

[8] M. Iannelli, “Mathematical problems in the description of age structured populations,” in *Mathematics in Biology and Medicine: Proceedings of an International Conference held in Bari, Italy, July 18–22, 1983*, pp. 19–32, Springer, 1985.

[9] E. Platen, “An introduction to numerical methods for stochastic differential equations,” *Acta numerica*, vol. 8, pp. 197–246, 1999.

[10] K. Boukhetala, A. Guidoum, and M. A. Guidoum, “Package ‘sim. diffproc’,” 2012.

[11] M. J. Panik, “Stochastic differential equations: An introduction with applications in population dynamics modeling,” 2017.

[12] P. Wilmott, S. Howson, S. Howison, J. Dewynne, *et al.*, *The mathematics of financial derivatives: a student introduction.* Cambridge university press, 1995.

[13] D. F. Kuznetsov, “Multiple Itô and Stratonovich stochastic integrals: approximations, properties, formulas,” 2013.