DTU Compute

02425
October 12, 2022
UHT/uht

Exercises for session 6: Itô calculus and SDE's

Using Itō's lemma to verify of solution

Question 1: Verify that $Y_t = \sinh B_t$ satisfies the Itô SDE

$$dY_t = \frac{1}{2} Y_t \, dt + \sqrt{1 + Y_t^2} \, dB_t$$

Solution: The verification consists of the following steps: We write $Y_t = h(B_t)$ and we use Itō's lemma to write $\{Y_t\}$ as an Itō process. Then we eliminate B_t from the drift and noise intensity. The following piece of `Maple` code does the computations:

```
h := sinh:

## Identify the inverse
hi := unapply(solve(y=h(b),b),y);

## Determine the increment of Y using Ito's lemma
dY := diff(h(B),B) * dB + 1/2 * diff(h(B),B,B) * dt ;

## Eliminate B
dY := simplify( subs( B = hi(Y), dY ) );
```

The code produces the following output:

```
                  hi := y -> arcsinh(y)

            dY := cosh(B) dB + 1/2 sinh(B) dt

                        2     1/2        Y dt
              dY := (Y  + 1)     dB + ----
                                          2
```

which confirms that $\{Y_t\}$ satisfied the SDE, as claimed.

Stochastic logistic growth

Consider the Itō stochastic differential equation governing $\{X_t\}$, the abundance of bacteria in a population:

$$dX_t = X_t(1 - X_t) \, dt + \sigma X_t \, dB_t$$

Question 2 Using Itō's lemma to perform a coordinate transformation: Identify a Lamperti transform h, i.e. find a transformed coordinate $Y_t = h(X_t)$ such that the Itō equation for $\{Y_t\}$ has additive noise. Write up this Itō equation.

Solution: According to the notes (equation (7.5) on page 155), we can find the transform by integrating $1/g(x)$ where g is the noise intensity. Once the transformation has been found, we apply Itō's lemma to write the transformed coordinate $\{Y_t\}$ as an Itō process. Finally we eliminate the original coordinate X_t using the inverse transformation, so that the equation for Y_t refers only to Y_t and not to X_t. The following piece of `Maple` code does the calculations:

```
f := x -> x*(1-x) :
g := x -> sigma * x:
gp := D(g):

## Find the Laperti transform as the antiderivative of 1/g
h := unapply(int(1/g(x),x),x);

## Find the inverse of the Lamperti transform
hi := unapply(solve(h(x) = y,x),y);

## Pose the SDE expressed in Y=h(X)
dY := ( f(X) / g(X) - 1/2 * gp(X) ) * dt + dB;

## ... alternatively, using Ito's lemma directly:
dY := ( f(X) * D(h)(X) + 1/2*g(X)^2*D(D(h))(X) ) * dt + D(h)(X)*g(X)*dB;

## Eliminate X and write in out usual form with a dt-term and a dB-term
dY := collect(simplify(subs( X = hi(Y), dY)),dt);
```

The code produces the following output:

```
                    ln(x)
        h := x -> ------
                    sigma

        hi := y -> exp(sigma y)

              /1 - X     sigma\
        dY := |-----  -  -----| dt + dB
              \sigma       2  /

              /1 - X     sigma\
        dY := |-----  -  -----| dt + dB
              \sigma       2  /

                        2
                 (-sigma  - 2 exp(sigma Y) + 2) dt
        dY := 1/2 --------------------------------- + dB
                              sigma
```

Now, `Maple`'s idea of simplified output does not completely agree with ours. So we rewrite this manually:

$$dY_t = \left[\frac{1}{\sigma} - \frac{\sigma}{2} - \frac{1}{\sigma}e^{\sigma Y_t}\right] dt + dB_t$$

Note that it has the desired form, i.e. the noise intensity that multiplies dB_t in this coordinate system is constant and equal to 1. Note also that when the state Y_t is very negative, i.e. X_t is near 0, then the drift in Y_t is approximately constant. As the state Y_t grows, the term $\exp(\sigma Y_t)$ kicks in an prevents unlimited growth.

Numerical analysis with the Euler-Maruyama method

Question 3: Simulate a sample path of $\{X_t\}$ with the Euler-Maruyama discretization method on the time interval $[0, 10]$. Take $\sigma = 0.2$ and $X_0 = 0.1$. Choose a sufficiently (but not excessively) small time step. Plot the solution.

Solution: See the answer to the next question; the code there solves both this question and the next.

Question 4: Extend the Euler-Maruyama simulation so that it solves the Itō equation for $\{X_t\}$ *and* the Itō equation for $\{Y_t\}$ in the same loop. Finally, determine numerically the solution to

$$dZ_t = h'(X_t)\ dX_t + \frac{1}{2}h''(X_t)\ d[X]_t$$

Plot in the same window X_t, $h^{-1}(Y_t)$, and $h^{-1}(Z_t)$ versus time.

Solution: We first set up the function in the SDE and in the transform

```
## Basic SDE
f <- function(x) x*(1-x)
g <- function(x) sigma*x

## Transform, its derivatives, and its inverse
h <- function(x) log(x)/sigma
dhdx <- function(x) 1/x/sigma
dh2dx2 <- function(x) -1/x^2/sigma

hi <- function(y) exp(y*sigma)
```

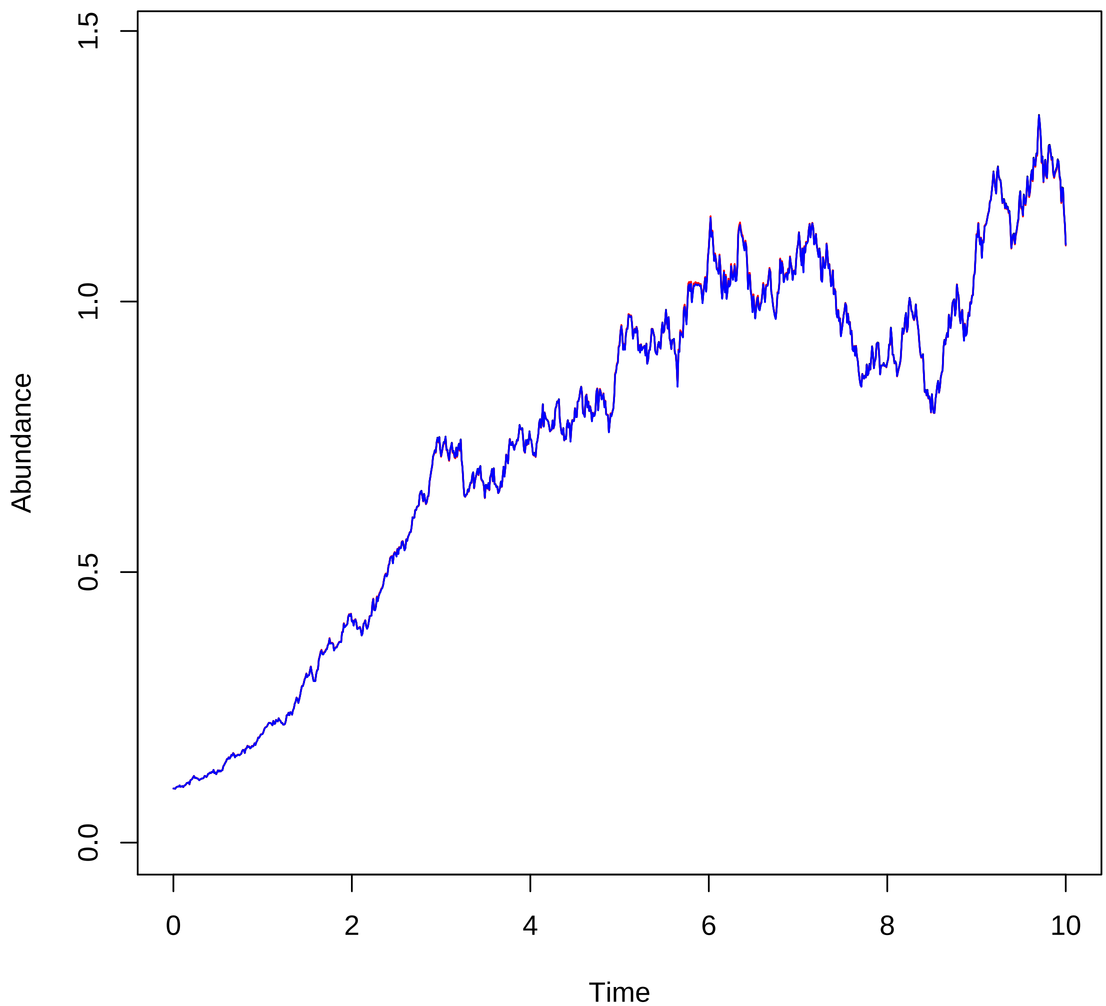
The following is the core function for simulating the process. It simulates both $\{X_t\}$ and $\{Y_t\}$ using the Euler-Maruyama methods, and afterwards computes a $\{Z_t\}$ by numerical Itō integration.

```r
## Use package SDEtools for rBM, rvBM, stochint, and covariation
require(SDEtools)
```

```
## Loading required package:  SDEtools
```

```r
## Helper for simulating X Y and Z, given an initial condition, a time grid, and
## possibly a sample path of the Brownian motion
simXYZ <- function(X0,tv,B=NULL)
{
    ## If no sample path of B is given, simulate one
    if(is.null(B)) B <- rBM(tv)

    nt <- length(tv)
    dt <- diff(tv)

    dB <- diff(B)

    Y <- X <- numeric(nt)
    X[1] <- x0

    Y[1] <- h(X[1])

    for(i in 1:(nt-1))
    {
        dX <- f(X[i])*dt[i] + g(X[i]) * dB[i]
        X[i+1] <- X[i] + dX

        ## Use the general expression:
        ##
        ## Y[i+1] <- Y[i] + dhdx(X[i])*dX + 0.5*dh2dx2(X[i])*g(X[i])*g(X[i])*dt[i]
        ##
        ## or the specific one:

        Y[i+1] <- Y[i] + ((1-exp(Y[i]*sigma))/sigma  - sigma/2 ) * dt[i] + dB[i]
    }

    ## Compute Z by stochastic integration
    Z <- Y[1] +  stochint(dhdx(X),X) + 0.5*stochint(dh2dx2(X),covariation(X,X))

    return(list(X=X,Y=Y,Z=Z))
}
```

```
## Compare different algorithms
sigma <- 0.2
x0 <- 0.1

tv <- seq(0,10,0.01)

sim <- simXYZ(X0,tv)

plot(tv,sim$X,type="l",
     ylim=c(0,max(c(sim$X,sim$Y,sim$Z))),
     xlab="Time",
     ylab="Abundance")
lines(tv,hi(sim$Y),col="red")
lines(tv,hi(sim$Z),col="blue")
```



It is quite difficult to tell the lines apart. So therefore we compute the difference:

```
print(max(abs(sim$X-hi(sim$Y))))
```

```
## [1] 0.004108396
```

```
print(max(abs(sim$X-hi(sim$Z))))
```

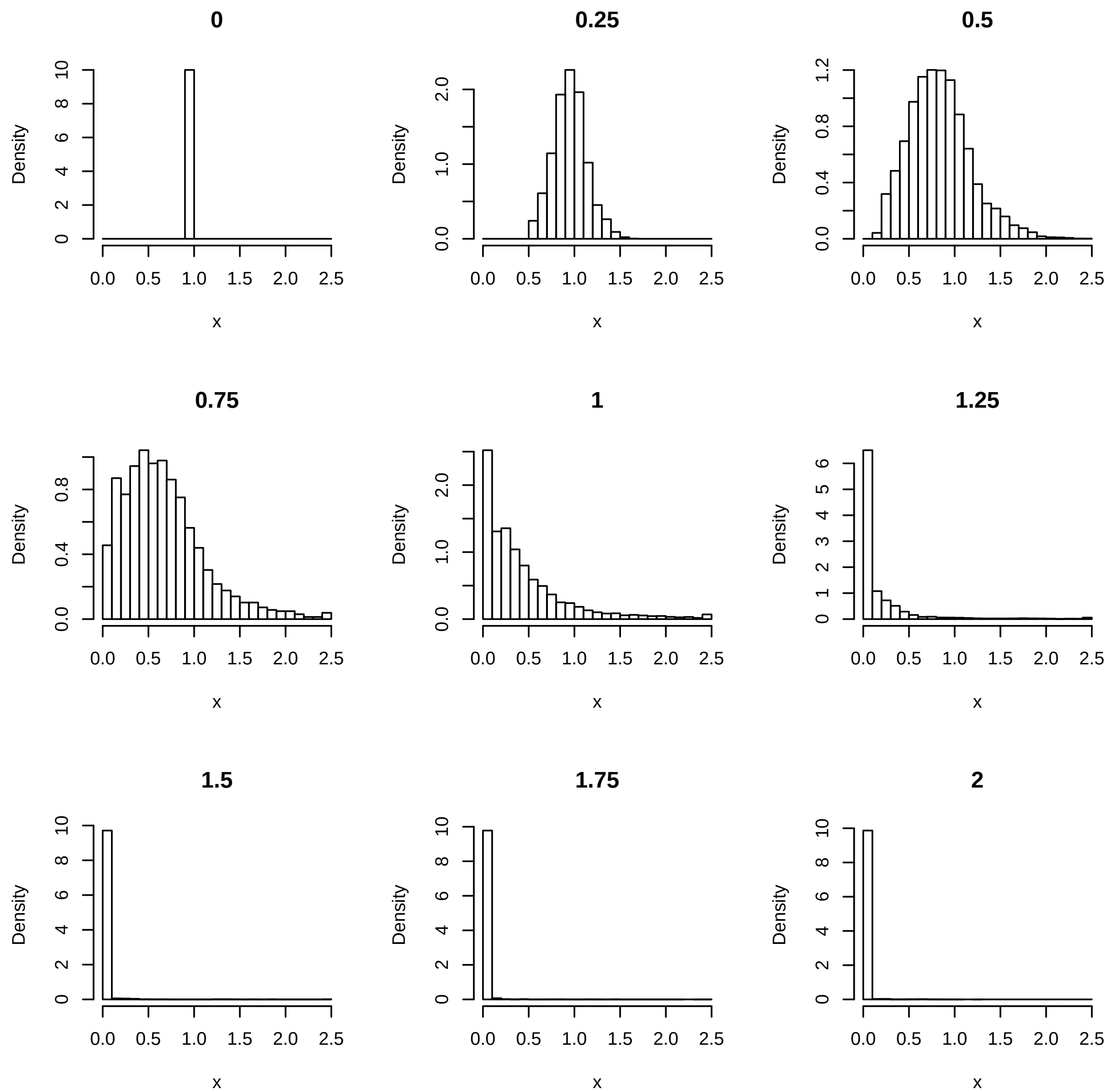
```
## [1] 0.001227433
```

As expected, the three solutions are reasonably close to each other - given that the Euler-Maruyama method is quite coarse.

Question 5: Extend the simulation of $\{X_t\}$ (or $\{Y_t\}$) for sufficiently long that the process appears to reach a steady state. Plot a histogram of the estimated stationary distribution of $\{X_t\}$, and estimate the mean and variance in steady state. Repeat with stronger noise, e.g. $\sigma = 0.5$, $\sigma = 1$ and $\sigma = 2$. What is the effect of increasing the noise?

Solution: Here, we compute the solution for 9 different values of σ.

```
## The effect of sigma on the stationary distribution
sigmas <- seq(0,2,0.25)
vars <- means <- numeric(length(sigmas))

## Long simulation
tv <- seq(0,100,0.01)

## Use the same Brownian motion
B = rBM(tv)

par(mfrow=c(3,3))

for(j in 1:length(sigmas))
{
    sigma <- sigmas[j]

    sim <- simXYZ(X0,tv,B)

    ## Base statistics only on the tail
    Xtail <- tail(sim$X,round(0.9*length(tv)))
    means[j] <- mean(Xtail)
    vars[j] <- var(Xtail)

    ## Plot the histogram
    ## Fix the break points so it is easier to compare between different values of sigma
    Xmax <- 2.5
    hist(pmin(Xtail,Xmax),breaks=seq(0,Xmax,0.1),main=sigma,xlab="x",freq=FALSE)
}
```

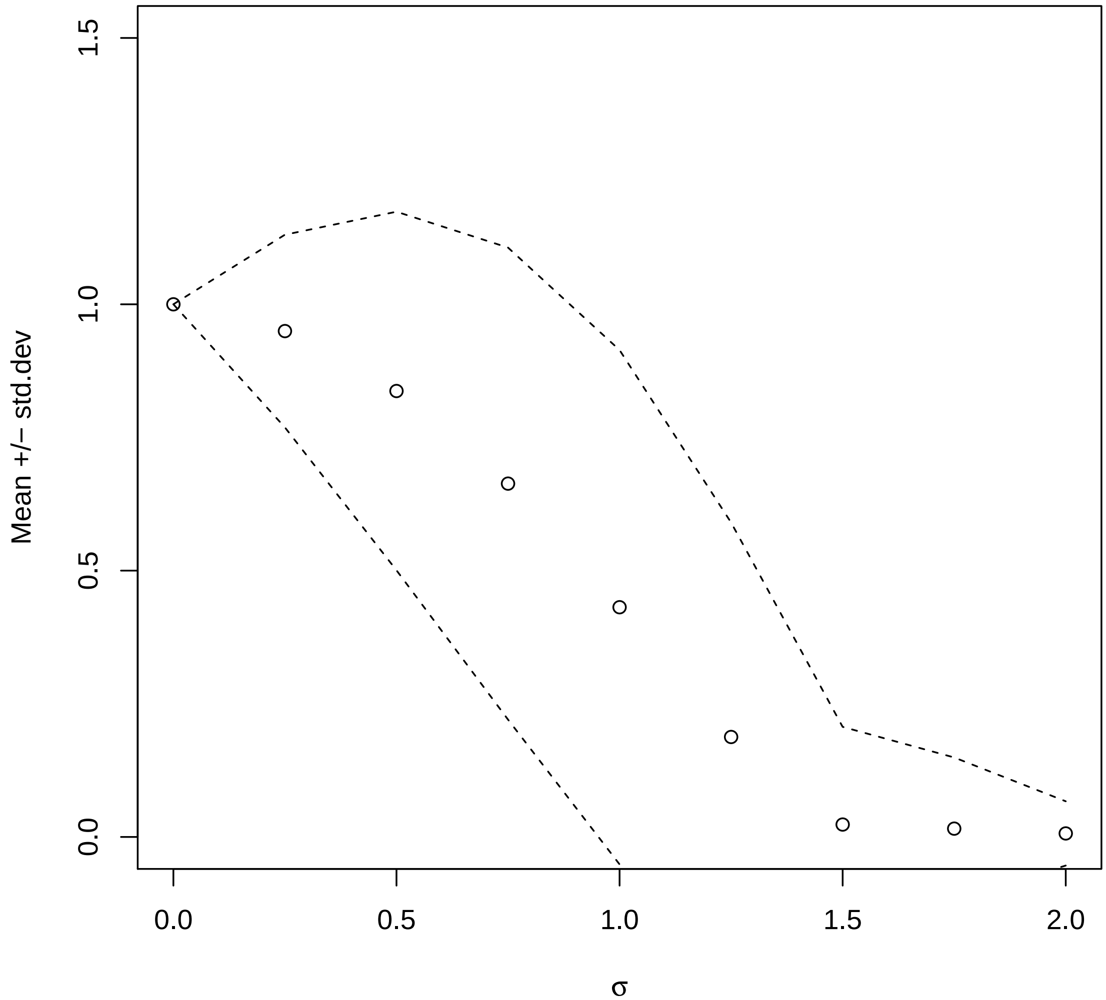


We see that without noise, the long-term behavior of the process is to be at rest at the equilibrium, $X_t = 1$. With weak noise, we see a stationary distribution which is - more or less - a Gaussian which is centered at the mean value $X_t = 1$ (a closer inspection reveals that the stationary expectation is only 1 in the limit $\sigma \to 0$; we will return to this later in the course when discussing transition probabilities and stationary distributions).

As the noise is increased, the stationary distribution not just has bigger variance but is increasingly skewed. At $\sigma = 1.5$, the solution appears to collapse to a Dirac delta at the origin. Here, the population has gone extinct. We will return to this *bifurcation* later in the course, when discussing stochastic stability.

We can highlight the effect of the noise level σ on the stationary mean and standard deviation:

```
plot(sigmas,means,ylim=c(0,1.5),xlab=expression(sigma),ylab="Mean +/- std.dev")
lines(sigmas,means+sqrt(vars),lty="dashed")
lines(sigmas,means-sqrt(vars),lty="dashed")
```



Note that noise affects the mean negatively. In the beginning, the standard deviation grows with the noise intensity, but as the process approaches collapse, the size of fluctuations are reduced again. We will soon obtain analytical results for these statistics (for this specific model).

Mean and variance in a the noisy harmonic oscillator

Consider the SDE (compare also exercise **??**)

$$dX_t = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} X_t \, dt + \sigma \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \, dB_t$$

with the initial condition $X_0 = x$. Here $X_t \in \mathbf{R}^2$ and $\{B_t : t \geq 0\}$ is two-dimensional Brownian motion.

Question 6: Verify that

$$\mathbf{E}X_t = \begin{bmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{bmatrix} x$$

for all t.

Solution: We must verify that the postulated expectation $\mathbf{E}X_t$ satisfies the equation

$$\frac{d}{dt}\mathbf{E}X_t = A\mathbf{E}X_t.$$

This can be done easily by hand; the `Maple` code in the answer to the next question also performs the computation.

Question 7: Write $S_t = \|X_t\|^2$ as an Itô process and find $\mathbf{E}S_t$ as a function of t.

Solution: We use the multivariate form of Itō's lemma to find

$$dS_t = (2X_t^\top AX_t + \mathrm{tr}(G^\top G))\ dt + 2X^\top G\ dB_t.$$

The following `Maple` code does the matrix multiplication and the simplification:

```
with(LinearAlgebra):

A := Matrix([ [ 0,1 ], [-1,0] ]):
EXt := Matrix([ [ cos(t) , sin(t) ] , [ -sin(t) , cos(t) ] ]):

## Verify the expectation
map(diff,EXt,t) - A . EXt ;

## Find the Ito equation for S
X := Vector(2,i -> x[i]):
dB := Vector(2,i -> db[i]):

G := sigma * IdentityMatrix(2):

dS := (2*Transpose(X).A.X + Trace(Transpose(G).G))*dt + 2*Transpose(X).G.dB ;

## Find the expectation of St
ESt = int( coeff(dS,dt),t=0..t);
```

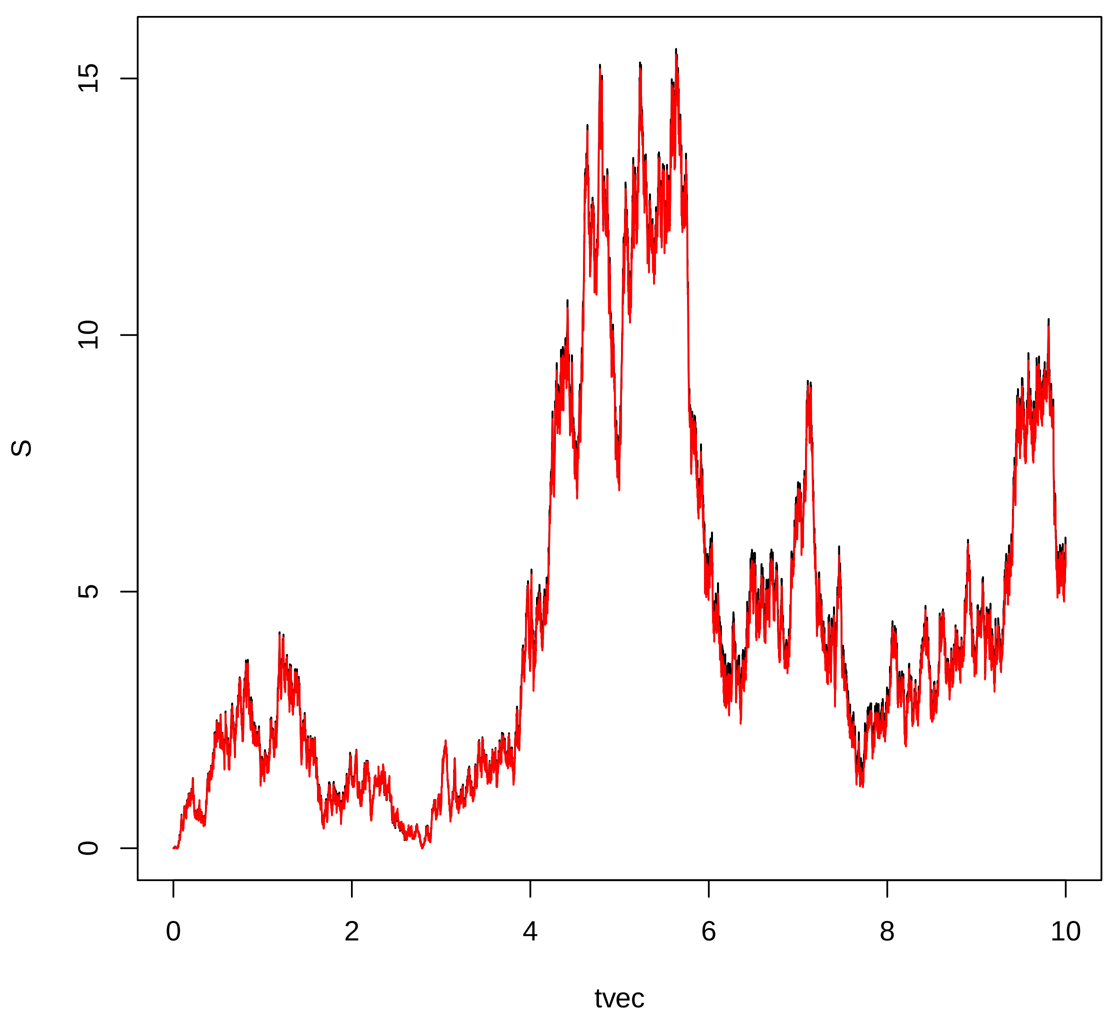
and produces the following output:

```
                    [0    0]
                    [      ]
                    [0    0]

                  2
dS := 2 dt sigma   + 2 sigma db[1] x[1] + 2 sigma db[2] x[2]

                              2
                  ESt = 2 sigma  t
```

Note that the expectation $\mathbf{E}S_t$ is easy to find, because the drift term is so simple. The fluctuations in S_t would be somewhat more demanding to describe.

The following `R` code verifies the results using simulation (this is not asked for in the exercise).

```
A <- array(c(0,-1,1,0),c(2,2))
sigma <- 1
G <- sigma*diag(c(1,1))

tvec <- seq(0,10,0.001)

B <- rvBM(tvec,n=2)
sim <- euler(f=function(x) A %*% x,g=function(x) G,times=tvec,x0=numeric(2),B=B)

S <- apply(sim$X^2,1,sum)
S1 <- 2*stochint(sigma*sim$X[,1],B[,1]) + 2*stochint(sigma*sim$X[,2],B[,2]) + 2*sigma^2*tvec

plot(tvec,S,type="l")

lines(tvec,S1,col="red")
```



Question 8: Pose and solve the differential Lyapunov equation governing the variance-covariance matrix of X_t. *Hint:* To solve the equation, first guess that the variance-covariance matrix is diagonal.

Solution: The differential Lyapunov equation is

$$\dot{\Sigma}_t = A\Sigma_t + \Sigma_t A^\top + GG^\top$$

We use the hint that the matrix is diagonal:

$$\Sigma_t = \begin{bmatrix} \Sigma_t^{11} & 0 \\ 0 & \Sigma_t^{22} \end{bmatrix}$$

to obtain equations for the diagonal elements:

$$\dot{\Sigma}_t^{11} = \sigma^2, \quad \dot{\Sigma}_t^{22} = \sigma^2.$$

From this it is easy to state the solution:

$$\Sigma_t = \sigma^2 t \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Inserting, we can verify that the hint is correct, i.e. with 0 in the off-diagonals, this matrix solves the differential Lyapunov equation. Note that the oscillatory behavior of the process is not reflected in the variance (but would be in the autocovariance function).