DTU Compute

02425
December 4, 2024
UHT/uht

Exercise 9: The Kolmogorov equations II

Question 1 Random directions and the von Mises distribution: (Compare exercise 9.7 in the notes) Consider the Itō stochastic differential equation

$$dX_t = -\sin X_t \; dt + \sigma \; dB_t,$$

which can be viewed as a random walk on the circle which is biased towards $X_t = 2n\pi$ for $n \in \mathbf{N}$. This is a popular model for random reorientations, when there is a preferred direction.

1. Write up the forward Kolmogorov equation and show that a stationary solution is the so-called *von Mises* (or Tikhonov) distribution $\rho(x) = Z^{-1} \exp(\kappa \cos x)$, where $\kappa = 2/\sigma^2$. *Note:* This ρ integrates to 1 over an interval of length 2π, so we consider the state X_t an angle which is only well defined up to adding a multiple of 2π.
2. Take $\sigma = 1$. Simulate the process starting at $x = 0$ over the time interval $t \in [0, 100]$. Plot the trajectory and the histogram of the state. Compare the histogram with the stationary distribution.
3. Discretize the generator on $x \in [-\pi, \pi]$ using periodic boundary conditions. Determine the stationary distribution numerically from the generator and compare it, graphically, with the results from the previous question. *Note:* Unless the spatial grid is very fine, some numerical diffusion stemming from the discretization will affect the numerical solution.
4. Estimate the autocovariance function of $\{\sin X_t\}$ from the time series (using a built-in routine in your favorite software environment) and plot it. Use a sufficient large number of lags until you can see how long it takes for the process to decorrelate.
5. Compute the autocovariance function numerically from the following formula:

   $$\mathbf{E}[(h(X_0) - \mu)h(X_t)] = \int_{\mathbf{X}} \rho(x)(h(x) - \mu)[e^{Lt}h](x) \; dx.$$

   Here, we take $h = \sin$. We have $\mu = \mathbf{E}h(X_0) = \int_{\mathbf{X}} \rho(x)h(x) \; dx$. Add this autocovariance to the empirical plot from the previous.
6. Compute the slowest 3 eigenmodes of L from the numerical discretization. Add to the plot of the autocovariance an exponentially decaying function $e^{-\lambda t}\mathbf{V} \sin X_0$ where λ is the largest non-zero eigenvalue of L. Comment on the agreement. Then, in a different plot, plot the slowest 3 eigenfunctions of L as well as of L^* and describe their role.

Solution: The forward Kolmogorov equation is

$$\dot{\rho} = -(\rho \sin x)' + \frac{1}{2}\sigma^2 \rho''.$$

The stationary distribution is the canonical distribution

$$\rho(x) = \frac{1}{Z} \exp(-U(x)/D)$$

where, as always, $D = \sigma^2/2$ and the potential U is an antiderivative to $-f(x) = \sin x$, i.e. $U(x) = -\cos x$. So

$$\rho(x) = \frac{1}{Z} \exp(D^{-1} \cos x)$$

as claimed, with $\kappa = 1/D = 2/\sigma^2$.

The question does not ask us to find Z, but we can: We normalize the distribution over $[0, 2\pi)$, obtaining

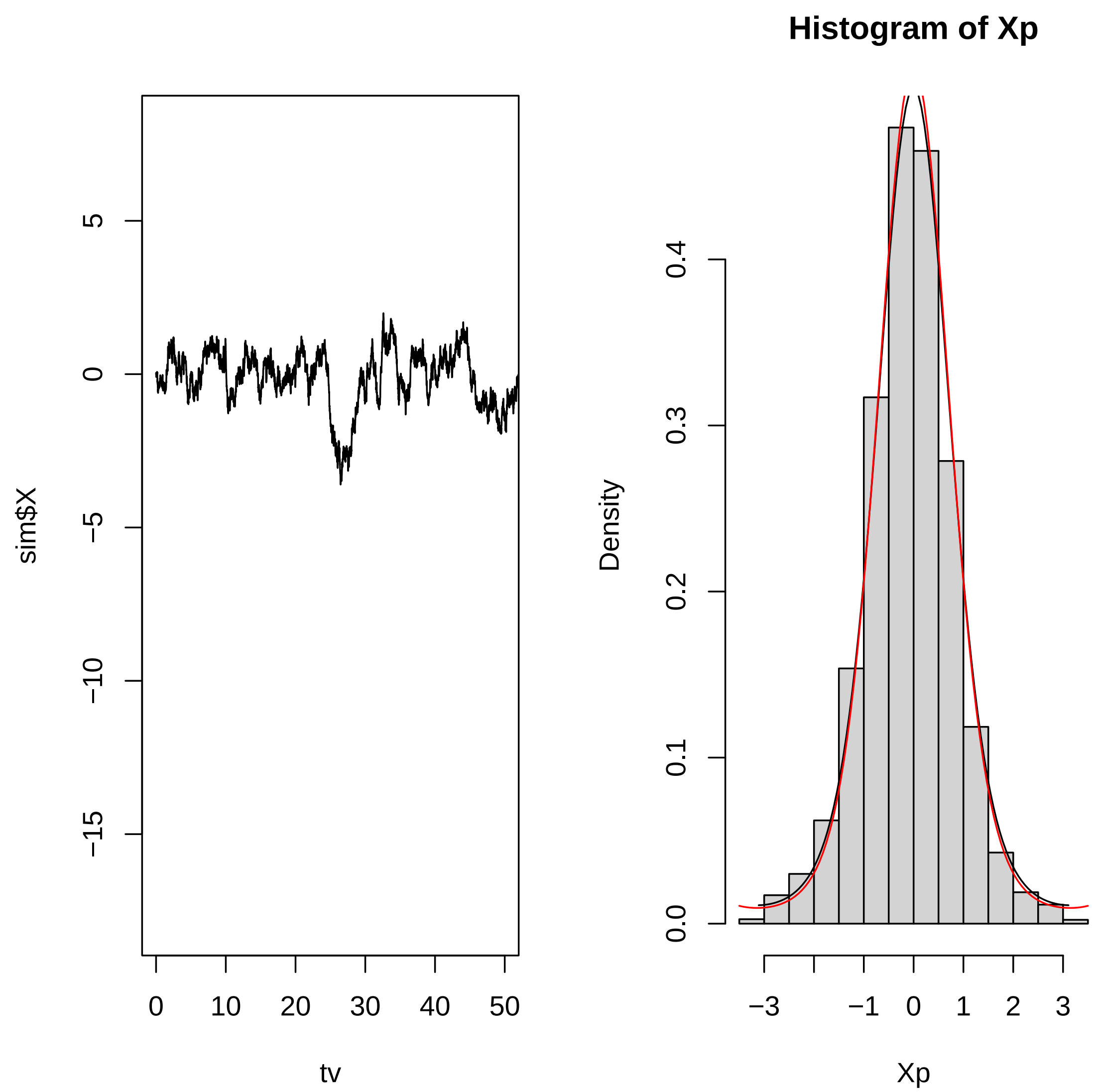
$$Z = \int_0^{2\pi} \exp(-U(x)/D)\ dx = 2\pi I_0(2/\sigma^2)$$

where $I_0(\cdot)$ is the modified Bessel function of the first kind of order 0.

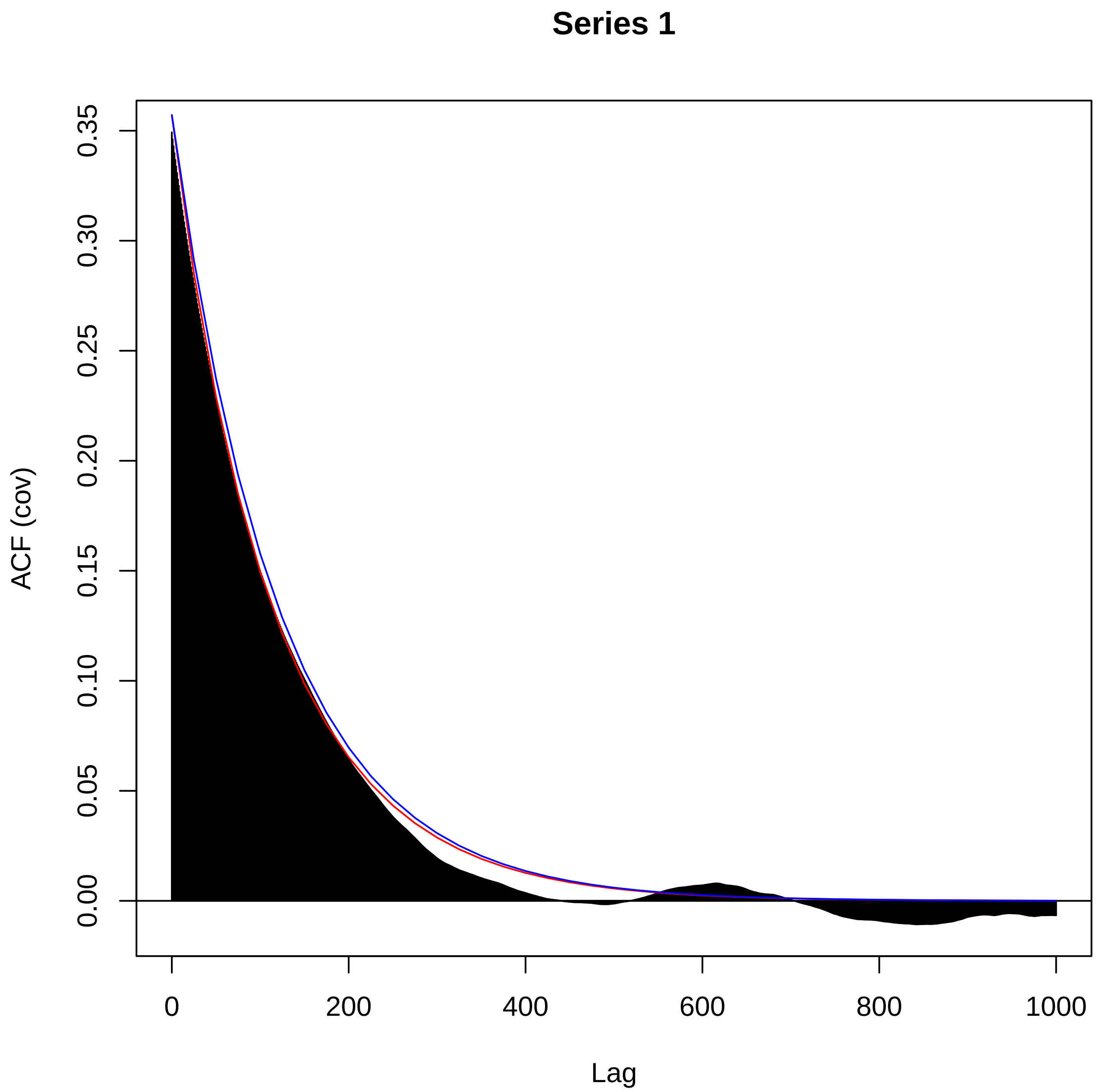
For the numerical part of the exercise:

```
rm(list=ls())
graphics.off()

require(SDEtools)


## Loading required package: SDEtools


xi <- seq(-pi,pi,length=101)
dx <- diff(xi)
xc <- 0.5*(tail(xi,-1)+head(xi,-1))

sigma <- 1

f <- function(x) -sin(x)
g <- function(x) sigma

u <- function(x) f(x)
D <- function(x) 0.5*sigma^2

G <- fvade(u,D,xi,'p')


## Loading required package: Matrix


rho <- StationaryDistribution(G) / diff(xi)

## Simulation

dt <- 0.01
tv <- seq(0,2000,dt)
sim <- heun(f,g,tv,0)
```

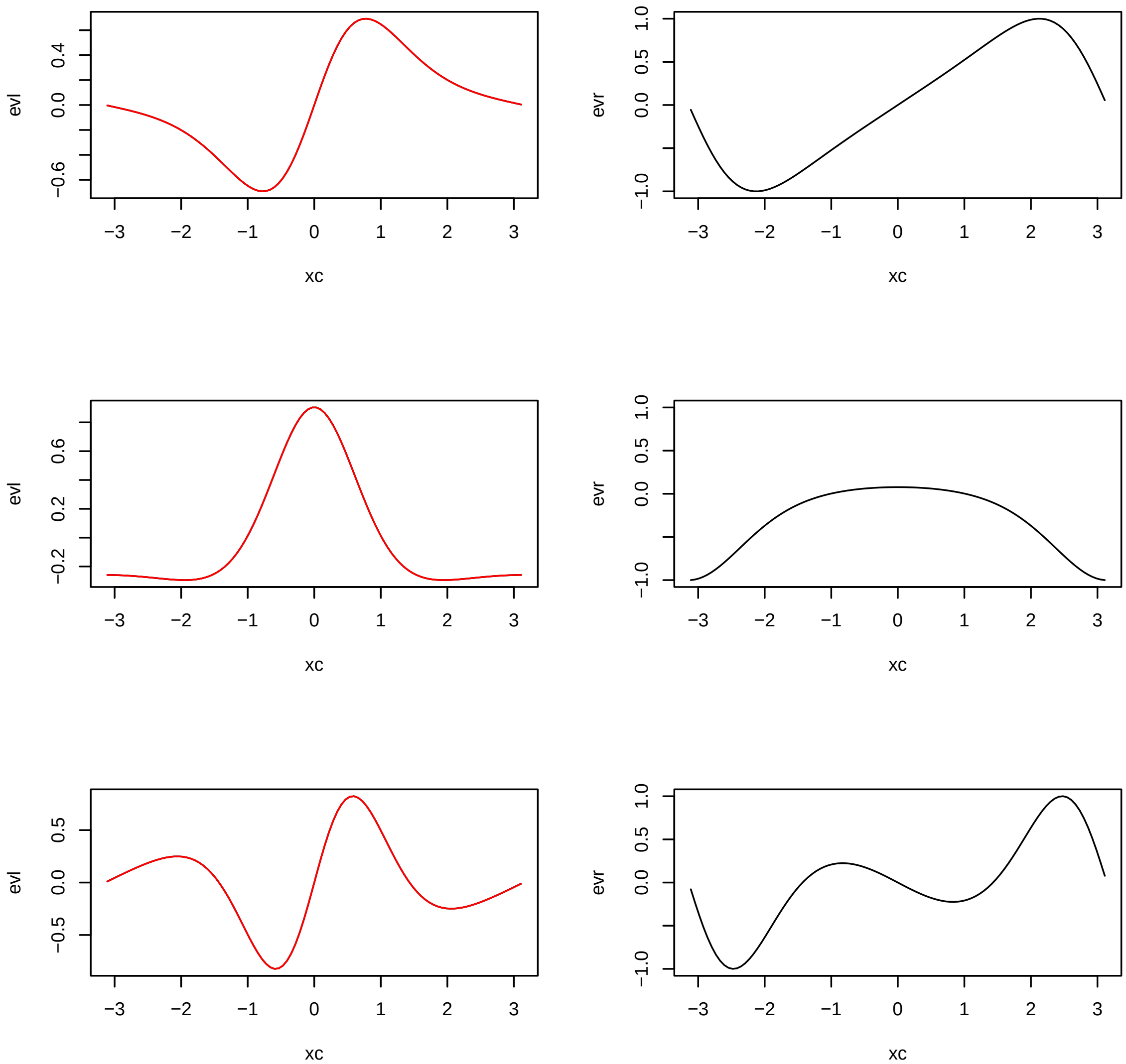
```r
par(mfrow=c(1,2))
plot(tv,sim$X,type="l",xlim=c(0,50))

## Project X on the interval [-pi,pi]
Xp <- (sim$X + pi) %% (2*pi) - pi
hist(Xp,freq=FALSE)


lines(xc,rho)

Z <- 2*pi*besselI(2/sigma^2,0)
curve(1/Z*exp(cos(x)*2/sigma^2),add=TRUE,col="red")
```



```
acf(sin(sim$X),type="cov",lag=1000)

h <- sin(xc)
mu <- sum(rho*h*diff(xi))

acf.theory <- Vectorize(function(ti)
    sum(rho*(h-mu)*(expm(G*ti)%*% h)*diff(xi)))

tv <- seq(0,10,0.25)
lines(tv/dt,acf.theory(tv),col="red")

## Evs
require(RSpectra)


## Loading required package: RSpectra


er <- eigs(G,k=4,sigma=1e-8)
el <- eigs(t(G),k=4,sigma=1e-8)

er$values <- Re(er$values)
el$values <- Re(el$values)
er$vectors <- Re(er$vectors)
el$vectors <- Re(el$vectors)

n <- nrow(G)

lambda <- -sort(abs(er$values))[2]
lines(tv/dt,acf.theory(0) * exp(lambda * tv),col="blue")
```


Series 1
ACF (cov)
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0
200
400
600
800
1000
Lag

```r
par(mfrow=c(3,2))
for(i in 1:3)
{
    ## Extract left eigenvector
    evl <- el$vectors[,4-i]
    ## Normalize to 2-norm = 1
    evl <- evl / sqrt(sum(evl^2/dx)) / dx

    ## Extract right eigenvector
    evr <- er$vectors[,4-i]
    evr <- evr / max(abs(evr))

    plot(xc,evl,type="l")
    evl2 <- evr*rho
    evl2 <- evl2 * sum(evl^2)/sum(evl2*evl)
    lines(xc,evl2,col="red")
    plot(xc,evr,type="l",ylim=c(-1,1))
}
```


evl
evr
xc
−3 −2 −1 0 1 2 3
−0.6 0.0 0.4
−1.0 0.0 0.5 1.0
−0.2 0.2 0.6
−0.5 0.0 0.5