

Problem Set 12

Ian McGroarty

26APRIL2020

```
blackScholes = function(T,N, lambda, mu) {

  ##   Loop over time steps

  timeIncrements = seq(0, T,T/N);
  W_vec = rep(0, N+1);
  X0 = 1;
  # Euler-Maruyama (Solution)
  EM_soln = rep(0,N+1);
  EM_soln[1] = X0;

  for (j in 0:N) {

    if (j == 0) {
      W = 0;
      EM = X0;
    }
    if (j > 0) {
      dW = rnorm(1,0, sqrt(T/N));
      W = W_vec[j] + dW
      # (eqn. 4.5)
      EM = EM_soln[j] + lambda*EM_soln[j]*T/N + mu*EM_soln[j]*dW
    }
    W_vec[j+1] = W
    EM_soln[j+1] = EM

  }
  # (eqn. 4.6)
  Xt = X0*exp((lambda-mu^2/2)*timeIncrements + mu*W_vec)
  #par(mfrow = c(1,2))
  plot(timeIncrements, Xt)
  plot(timeIncrements, EM_soln)

}

blackScholes(1,1000,lambda=2,mu=1)
```

Problem 1:

In class I simulated a solution to the Black-Scholes model (using the exact solution) and compared it to the solution that I got using the Euler-Maruyama approximation. Do the same thing for the Ornstein-Uhlenbrek process. Remember that stochastic differential equation was written as

$$dX(t) = \theta(\mu - X(t))dt + \sigma dW(t)$$

and the analytical solution is

$$X(t) = X(0)e^{-\theta t} + \mu(1 - e^{-\theta t}) + \int_0^t \sigma e^{\theta(s-t)} dW(s)$$

To solve this we are going to have to take a closer look at the solution. There is a pretty obvious problem here which is that it contains a stochastic integral. So what we need to do is break this out. Conveniently enough, we can do this using the Euler-Maruyama approximation.

$$\begin{aligned} X(t) &= X(0)e^{-\theta t} + \mu(1 - e^{-\theta t}) + \int_0^t \sigma e^{\theta(s-t)} dW(s) \\ &= X(0)e^{-\theta t} + \mu(1 - e^{-\theta t}) + \sigma e^{-\theta t} \underbrace{\int_0^t e^{\theta s} dW(s)} \end{aligned}$$

We can estimate the underlined term using the Euler-Maruyama approximation. This is illustrated below.

```
blackScholes.OUP = function(T,N, theta, mu, sigma) {

  ##   Loop over time steps

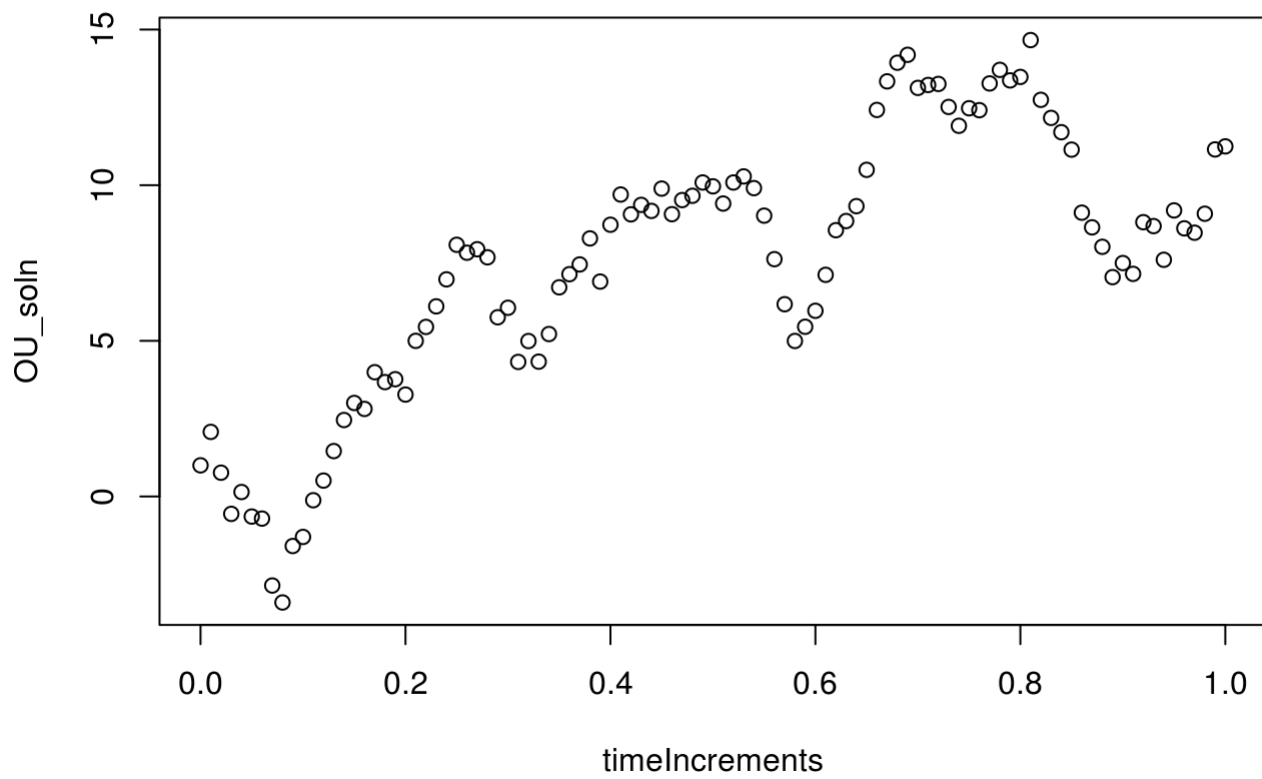
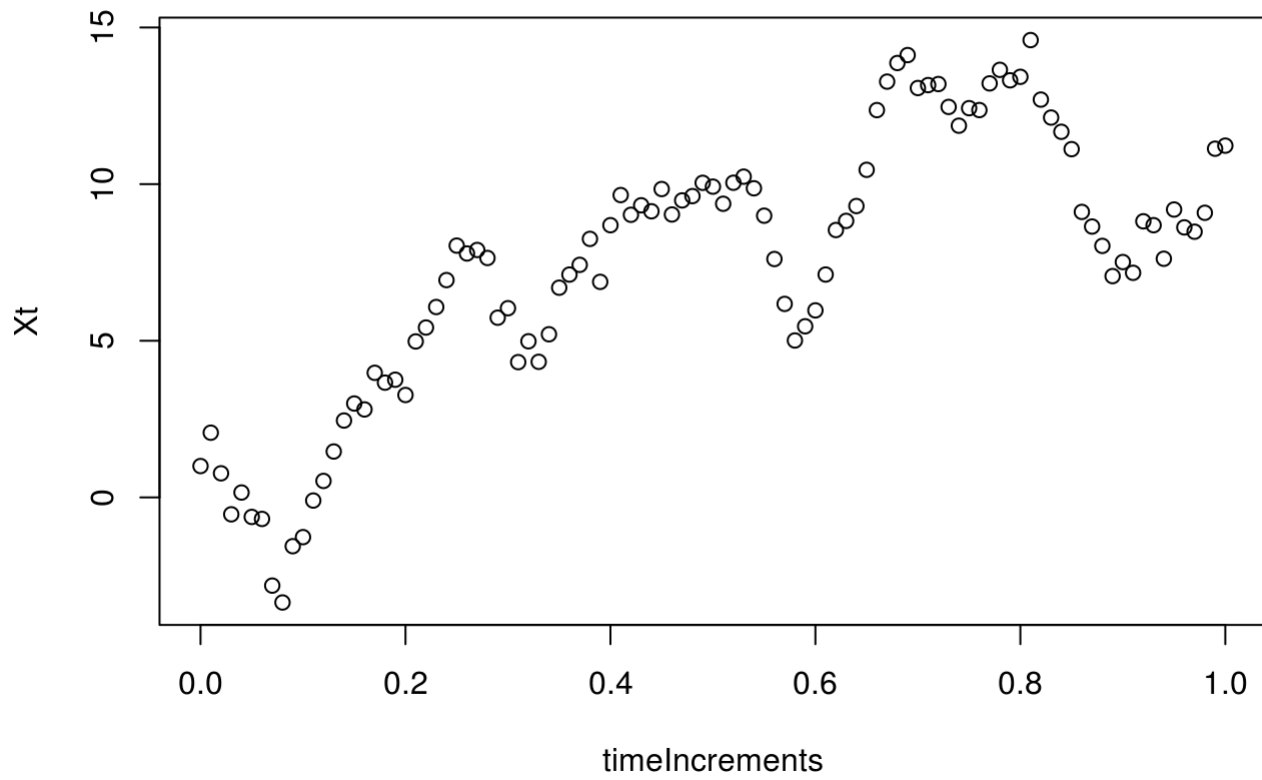
  timeIncrements = seq(0, T,T/N)
  W_vec = rep(0, N+1)
  X0 = 1;
  # Ornstein-Uhlenbrek (Solution)
  OU_soln = rep(0,N+1)
  OU_soln[1] = X0

  for (j in 0:N) {

    if (j == 0) {
      W = 0
      OU = X0
    }
    if (j > 0) {
      dW = rnorm(1,0, sqrt(T/N))
      # This is a cumulative sum so it evaluates like an integratal.
      W = W_vec[j] + exp(theta*timeIncrements[j])*dW
      OU = OU_soln[j] + (theta*(mu-OU_soln[j]))*(T/N) + sigma*dW
    }
    W_vec[j+1] = W
    OU_soln[j+1] = OU
  }

  ex <- exp(-1*theta*timeIncrements)
  Xt = X0*ex + mu*(1-ex) + sigma*ex*W_vec
  #+ sum(sigma*exp(theta*N)*W_vec[j+1])
  #par(mfrow = c(1,2))
  plot(timeIncrements, Xt)
  plot(timeIncrements, OU_soln)
}

blackScholes.OUP(1,100,theta=1,mu=20,sigma=10)
```



Problem 2

Illustrate through a simulated R example the difference between the Ito integral and the Stratonovich integral. That is, find some value of $h(t)$ and simulated process $W(t)$ so that calculating the integral $\int_0^T h(t)dW(t)$ is considerable different for the two calculations.

For this I use the example presented in the notes. I realize this is a cop out but it seems very straightforward and simple.

```
T <- 1
N <- 500
dt <- T/N
dW <- rnorm(n=N, mean=0, sd=sqrt(dt))
W <- cumsum(dW)

ito <- sum(c(0, W[1:(N-1)]) * dW)
strat <- sum((0.5 * (c(0, W[1:(N-1)]) + W) + 0.5 * dW) * dW)
print(paste0("The ito estimation is ", ito, ". The Stratonovich estimation is ", strat))
```

```
## [1] "The ito estimation is -0.440519499435028. The Stratonovich estimation is 0.613046920373839"
```

Problem 3

Below I've conducted the blackScholesConvergence function for 5 steps of size $(10^{-(1-p)})$ with $p \in [1, 5]$. The red line shows the approximate line of best fit (based on log-log regression). While this is not exactly $(1/2)$ it is fairly close and demonstrates the approximate order of convergence to be $(1/2)$.

ref: <https://epubs.siam.org/doi/pdf/10.1137/S0036144500378302>

```
blackScholesConvergence = function(T,N, lambda, mu, nmbSimulations) {

  ##   Loop over time steps

  vectorOfFinalDifferences = rep(0, nmbSimulations)

  for (iSim in 1:nmbSimulations) {

    timeIncrements = seq(0, T,T/N);
    W_vec = rep(0, N+1);
    X0 = 1;
    EM_soln = rep(0,N+1);
    EM_soln[1] = X0;

    for (j in 0:N) {

      if (j == 0) {
        W = 0;
        EM = X0;
      }
      if (j > 0) {
        dW = rnorm(1,0, sqrt(T/N));
        W = W_vec[j] + dW;
        EM = EM_soln[j] + lambda*EM_soln[j]*T/N + mu*EM_soln[j]*dW;
      }
      W_vec[j+1] = W;
      EM_soln[j+1] = EM;

    }
    Xt = X0*exp((lambda-mu^2/2)*timeIncrements + mu*W_vec);

    vectorOfFinalDifferences[iSim] = Xt[N+1] - EM_soln[N+1]

  }

  averageFinalDifference = mean(vectorOfFinalDifferences)
  return(c(averageFinalDifference, T/N))

}

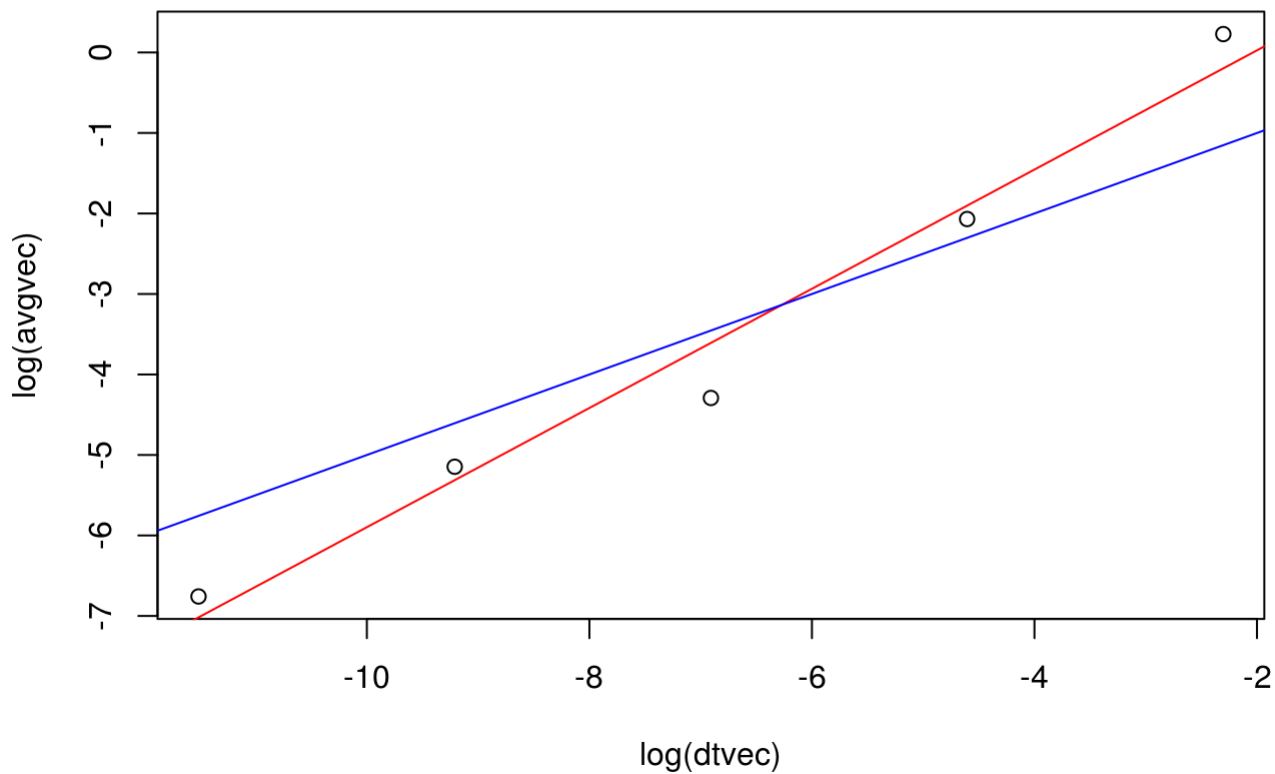
avgvec <- c()
dtvec <- c()
for (i in 1:5) {
  N <- 10^i

  a <- blackScholesConvergence(T=1,N,lambda=2,mu=1,nmbSimulations = 1000)
  avgvec[i] <- a[1]
  dtvec[i] <- a[2]
```

```

}
plot(log(dtvec),log(avgvec))
abline(lm(log(avgvec)~log(dtvec)),col='red')
abline(coef=c(0,.5),col='blue')

```



Problem 4

Consider the sequence of random variables $W_1 \cdots W_n$. Assume they are i.i.d. s.t.

$$W_j = \begin{cases} 1 & \text{with probability } 1/2 \\ -1 & \text{with probability } 1/2 \end{cases}$$

And consider the variables Z_1, \dots, Z_n s.t.

$$Z_j \sim N(0, \text{Var} = \frac{1}{n})$$

Below I've drawn 100 samples of W and Z and taken the difference using the strong and weak convergence. The red dots illustrate strong convergence which clearly move to 1. The blue dots illustrate weak convergence and clearly move to 0.

```

N <- 100
a <- rbinom(n=N,size= 1,prob=0.5)
W <- ifelse(a==1,a,-1)

Z <- c()
for (i in 1:N) {
  Z[i] <- rnorm(n=1,mean=0,sd=(sqrt(1/i)))
}

strong <- cumsum(abs(W - Z))/(1:N)
weak <- abs((cumsum(W)/(1:N)) - (cumsum(Z)/(1:N)))
plot(strong, col = 'red', ylim=c(-1,2), ylab="Convergence")
points(weak, col='blue')

```

