Université Nice-Sophia Antipolis
SMEMP302 - ECUE Probabilitic computational methods, 2022-2023
Sylvain Rubenthaler (https://math.unice.fr/~rubentha/)

Answers for the Home Project

(1) We prove it by recurrence.
- This is true for $k=0$.
- If this is true for k, then

$$\begin{aligned}
\mathbb{E}((Y_{k+1})^2) &= \mathbb{E}((Y_k)^2)(1+\mu\Delta)^2+\sigma^2\Delta \\
&= \frac{\sigma^2}{\mu}\times\frac{(1+\mu\Delta)^{2k}-1}{2+\mu\Delta}\times(1+\mu\Delta)^2+\sigma^2\Delta \\
&= \frac{\sigma^2}{\mu}\times\frac{(1+\mu\Delta)^{2k+2}}{2+\mu\Delta}+\frac{\sigma^2}{\mu}\left(\mu\Delta-\frac{(1+\mu\Delta)^2}{2+\mu\Delta}\right) \\
&= \frac{\sigma^2}{\mu}\times\frac{(1+\mu\Delta)^{2k+2}-1}{2+\mu\Delta}\,.
\end{aligned}$$

Conclusion: for all k, $\mathbb{E}((Y_k)^2)=\frac{\sigma^2}{\mu}\times\frac{(1+\mu\Delta)^{2k}-1}{2+\mu\Delta}$.

(2) Suppose X_{t_k} is fixed. Then $(X_{t-t_k})_{t\geq t_k}$ is solution of an Ornstein-Uhlenbeck equation with coefficients μ, σ and starting point X_{t_k}.Hence the formula.

(3) For all $a,b>0$:

$$\begin{aligned}
2ab &= 2a\sqrt{\Delta}\frac{b}{\sqrt{\Delta}} \\
&\leq a^2\Delta+\frac{b^2}{\Delta}
\end{aligned}$$

so

$$(a+b)^2 \leq (1+\Delta)a^2+(1+\frac{1}{\Delta})b^2\,. \tag{0.1}$$

We have (for all $k\geq 0$):

$$\begin{aligned}
Y_{k+1}-X_{t_{k+1}} &= (1+\mu\Delta)Y_k-e^{\mu\Delta}X_{t_k}+\sigma\int_{t_k}^{t_{k+1}}1-e^{\mu(t_{k+1}-s)}dW_s \\
&\quad\text{(as } (W_s)_{s\geq t_k} \text{ is independant of } X_{t_k},Y_k) \\
\mathbb{E}((Y_{k+1}-X_{t_{k+1}})^2) &= \mathbb{E}(((1+\mu\Delta)Y_k)-e^{\mu\Delta}Y_k+e^{\mu\Delta}Y_k-e^{\mu\Delta}X_{t_k})^2) \\
&\quad+\sigma^2\mathbb{E}\left(\left(\int_{t_k}^{t_{k+1}}1-e^{\mu(t_{k+1}-s)}dW_s\right)^2\right) \\
&\quad\text{(by Equation (0.1))} \\
&\leq (1+\Delta)e^{2\mu\Delta}\mathbb{E}((Y_k-X_{t_k})^2)+\left(1+\frac{1}{\Delta}\right)\mathbb{E}((Y_k)^2)(e^{\mu\Delta}-1-\mu\Delta)^2 \\
&\quad+\sigma^2\mathbb{E}\left(\int_{t_k}^{t_{k+1}}\left(1-e^{\mu(t_{k+1}-s)}\right)^2ds\right) \\
&= (1+\Delta)e^{2\mu\Delta}\mathbb{E}((Y_k-X_{t_k})^2)+\left(1+\frac{1}{\Delta}\right)\mathbb{E}((Y_k)^2)(e^{\mu\Delta}-1-\mu\Delta)^2 \\
&\quad+\sigma^2\int_0^{\Delta}(e^{\mu u}-1)^2du\,.
\end{aligned}$$

(4) We have, for all $k \in \{0,1,\dots,n\}$,

$$
\begin{aligned}
\mathbb{E}((Y_k)^2) &\leq \frac{\sigma^2}{\mu} \times \frac{(1+\mu\Delta)^{2k}}{2+\mu\Delta} \\
&\leq \frac{\sigma^2}{2\mu} \exp\left(2k\log(1+\mu\Delta)\right) \\
&\leq \frac{\sigma^2}{2\mu} \exp\left(2k\mu\Delta\right) \\
&\leq \frac{\sigma^2}{2\mu} e^{2\mu T}\,.
\end{aligned}
$$

(5) As $\Delta \leq 1$, there exists a constant C_1 and a constant C_2 such that

$$
|e^{\mu u} - 1| \leq C_1 u\,,\ \forall u \in [0,1]\,,
$$
$$
|e^{\mu u} - 1 - \mu u| \leq C_2 u^2\,,\ \forall u \in [0,1]
$$

(C_1 and C_2 depending on μ). So (by Question (3), Question (4)):

$$
\begin{aligned}
\mathbb{E}((Y_{k+1} - X_{t_{k+1}})^2) &\leq (1+\Delta_n)e^{2\mu\Delta_n}\mathbb{E}((Y_k - X_{t_k})^2) + 2\left(1 + \frac{1}{\Delta_n}\right) C_2^2 \Delta_n^4 \mathbb{E}(Y_k^2) \\
&\quad + \sigma^2 C_2^2 \frac{\Delta_n^3}{3} \\
&\leq (1+\Delta_n)e^{2\mu\Delta_n}\mathbb{E}((Y_k - X_{t_k})^2) + C\Delta_n^3\,.
\end{aligned}
$$

for some constant C depending on μ, T, σ. By recurrence, we get, for all $k \geq 1$ (as $Y_0 = X_0 = x$),

$$
\begin{aligned}
\mathbb{E}((Y_{k+1} - X_{t_{k+1}}) &\leq C\Delta_n^3 \sum_{i=0}^{k-1} ((1+\Delta_n)e^{2\mu\Delta_n})^i \\
&\leq C\Delta_n^3 \times \frac{1}{1-(1+\Delta_n)e^{2\mu\Delta_n}} \\
&\leq C\Delta_n^3 \times \frac{1}{1-(1+\Delta_n)(1+2\mu C_1\Delta_n)} \\
&\leq C\Delta_n^3 \times \frac{1}{(1+2\mu C_1)\Delta_n - 2\mu C_1 \Delta_n^2} \\
&\leq C\Delta_n^3 \times \frac{1}{\Delta_n} \\
&= C\Delta_n^2\,.
\end{aligned}
$$

(6) We write the following code

```
import numpy as np
import scipy.stats as sps
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
mu=-1 ; sigma=1 ; T=1; x=1

def euler(n):
        Y=1
        tab=np.array([])
        Y=x*Y
        Delta=T/n
        for i in range(n):
                Z=sps.norm.rvs(size=(M))
```

```
                Y=(1+mu*Delta)*Y+sigma*np.sqrt(Delta)*Z
                tab=np.append(tab,Y[0])
# we add the last value of the Euler scheme
        plt.plot(range(n+1),tab)

euler(1000)
```

and we get the graph in Figure 0.1.

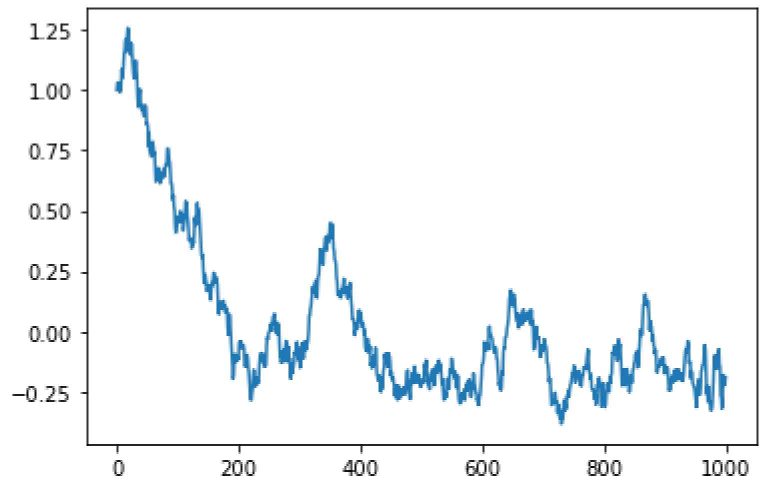

FIGURE 0.1. One trajectory of the Euler scheme

(7) We modify slightly tht above code.

```
def euler(n,M):
        Y=np.ones(M)
        tab=np.array(Y[0])
        Y=x*Y
        Delta=T/n
        for i in range(n):
                Z=sps.norm.rvs(size=(M))
                Y=(1+mu*Delta)*Y+sigma*np.sqrt(Delta)*Z
                tab=np.append(tab,Y[0])
# we add the last value of the Euler scheme
        return(Y)

# suppose that the function f is defined
print(np.mean(f(euler(1000,1000**2))))
```

(8) We know that X_T is Gaussian and centered in $xe^{\mu T}$. We could compute (useless with the chosen f)

$$
\begin{aligned}
\mathrm{Var}(X_1) &= \sigma^2 \int_0^T e^{2\mu(T-s)} ds \\
&= \sigma^2 \left[\frac{e^{2\mu(T-s)}}{-2\mu} \right]_0^T \\
&= \sigma^2 \left(\frac{1-e^{2\mu T}}{-2\mu} \right) .
\end{aligned}
$$

We write the following code.

```
def euler_mc(n):
        M=n**2
        return(np.abs(np.mean(euler(n,M))-x*np.exp(mu*T)))

euler_mc(1000)
```

(9) We write the following code

```
dessin=np.array([])
rr=range(3,12)
for i in rr:
        dessin=np.append(dessin,np.log(euler_mc(2**i))/np.log(2))
plt.plot(rr,dessin)
```

Here $n = 2^i$ with $i = 3, 4, \ldots, 11$. Si in `rr`, we have $\log_2(r)$. This is why we take the $\log_2$ of the values on the Y-axis. We get the graph in Figure 0.2.

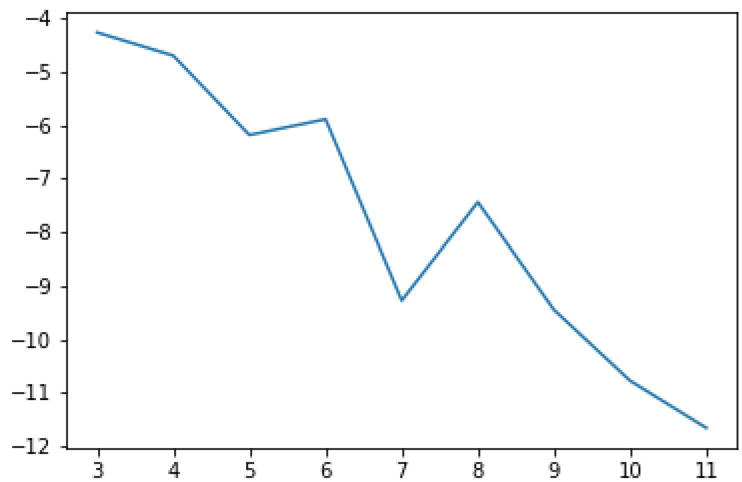

FIGURE 0.2. Log-log graph

(10) We write the following code

```
model = LinearRegression()
model.fit(np.array(rr).reshape((-1, 1)),dessin)
print(model.coef_)
```

and get the following answer

```
[-0.93066631]
```

So the true exponent might be -1.