Université Nice-Sophia Antipolis
SMEFA302 - ECUE Probabilitic computational methods, 2024-2025
Sylvain Rubenthaler (https://math.unice.fr/~rubentha/)

Exercices (2)

Exercise 1. Plot a Halton sequence in $[0,1]^2$ and a sequence of random numbers of uniform law in $[0,1]^2$ ($N = 1000$ terms). You should obtain something similar to Figure 0.1.

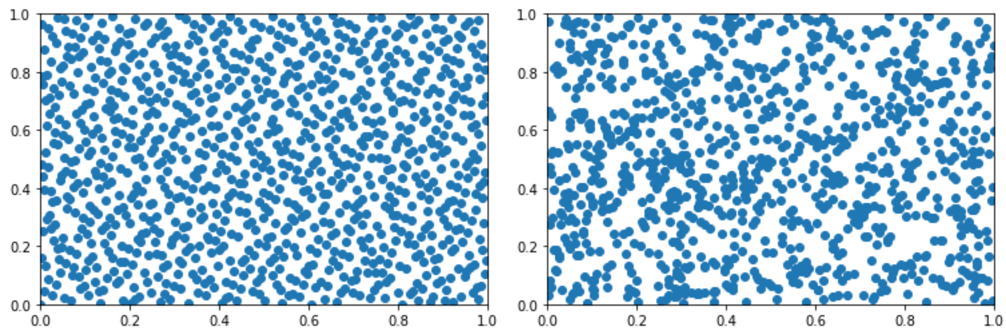

FIGURE 0.1. QMC (left) vs MC (right)

Exercise 2. We want to compute the price of the *Best of Call* option with strike K. This example is developped at the end of chapter 2 in the course. We recall the following formulas.

$$\begin{aligned} e^{-rT} h_T &\overset{\text{(law)}}{=} \phi(Z^1, Z^2) \\ &:= \left(\max \left(x_0^1 \exp\left(-\frac{\sigma_1^2}{2} T + \sigma_1 \sqrt{T} Z^1 \right), x_0^2 \exp\left(-\frac{\sigma_2^2}{2} T + \sigma_2 \sqrt{T} Z^2 \right) \right) - K e^{-rT} \right)_+ , \end{aligned}$$

where $Z = (Z^1, Z^2) \sim \mathcal{N}(0, I_2)$. We set

$$X_T^1 = x_0^1 \exp\left(rT - \frac{\sigma_1^2}{2} T + \sigma_1 \sqrt{T} Z^1 \right) , \; X_T^2 = x_0^2 \exp\left(rT - \frac{\sigma_2^2}{2} T + \sigma_2 Z^2 \right) .$$

We would like to compute

$$\mathbb{E}(h_T) = \mathbb{E}((\max(X_T^1, X_T^2) - K)_+) .$$

We set

$$x_0^1 = x_0^2 = 100 \, , \; T = 1 \, , \; r = 2\% \, , \; \sigma_1 = \sigma_2 = 30\% \, , \; K = 105 \, .$$

Implement a QMC-adapted Box-Muller[1] simulation method for $\mathcal{N}(0, I_2)$ (based on the Halton sequence) and organize a race MC vs QMC to compute the above call. You should obtain something similar to Figure 0.2.

PYTHON TIPS

Start by loading useful packages:

```
import numpy as np
from scipy.stats import qmc
import scipy.stats as sps
import matplotlib.pyplot as plt
import pandas as pd
```

[1] This is the method often used to simulate Gaussian variables using uniform variables. It is described in the course.

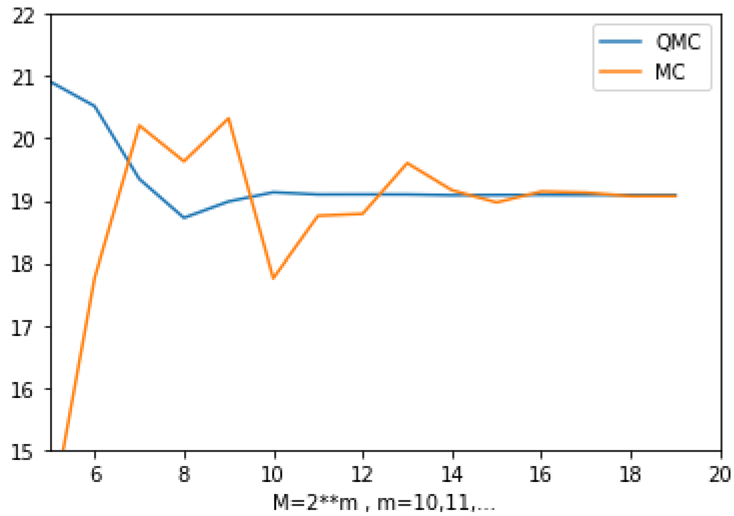

FIGURE 0.2. QMC vs MC

Sample a Halton sequence:

```
sampler=qmc.Halton(d=2,scramble=False)
                                #this is an object of class qmc.Halton,
                                #dimension=2
sample = sampler.random(n=N)
                        #we ask for the first N points in the sequence
df_sample=pd.DataFrame(sample)
                        #we convert into a data frame for further use
```

Scatter plot:

```
plt.scatter(df_sample[0],df_sample[1])
```

Add legend and label to a graph:

```
plt.legend(['QMC','MC'])
plt.xlabel('M=2**m , m=10,11,...')
```