```
## Exercice 1
## Loi uniforme

from random import random

def varUni(a=0,b=1):
    return a + (b-a) * random()


def simulUni(N,n=100, a=0, b=1):
    FreqX = [0] * n
    for k in range(N):
        x = varUni()
        i = int((x-a)/(b-a) * n)
        FreqX[i] += 1
    return [x/N for x in FreqX]

import numpy as np
import matplotlib.pyplot as plt
a, b = 1, 2.5
N = 10000
n = 100
I = np.arange(a,b,(b-a)/n)
FreqX = simulUni(N, n)
plt.clf()
plt.bar(I,FreqX,width = (b-a)/n)
plt.show()

Fx = [0] * n
for k in range(n-1):
    Fx[k+1] = Fx[k] + FreqX[k]
plt.figure(2)
plt.clf()
plt.title("Fonction de répartition")
plt.bar(I,Fx,width=(b-a)/n)
plt.show()

fx = [0] * n
for k in range(n):
    fx[k] = FreqX[k] * n/(b-a) # Attention !
plt.figure(3)
plt.clf()
plt.title("Densité de probabilité")
plt.axis([a,b,0,1.3])
plt.bar(I,fx,width=(b-a)/n)
plt.show()



## Exercice 2
## Loi Exponentielle
from random import random
import numpy as np
import matplotlib.pyplot as plt

# 1)
def varExp(param):
    u = random()
    return -np.log(1-u)/param

# 2)
Max = 15

def simulExp(N,n, param):
    FreqX = [0] * n
    Nbre = 0
    for k in range(N):
        x = varExp(param)
        if x < Max:
            i = int(x/Max * n)
            FreqX[i] += 1
            Nbre += 1
    return [x/Nbre for x in FreqX]

# 3)
N = 10000
n = 100
I = np.arange(0,Max,Max/n)
FreqX = simulExp(N, n, 0.5)
plt.figure(4)
```

```
plt.clf()
plt.title("Simulation loi exponentielle")
plt.bar(I,FreqX,width = Max/n)
plt.show()

# 4)
Fx = [0] * n
for k in range(n-1):
    Fx[k+1] = Fx[k] + FreqX[k]
plt.figure(5)
plt.clf()
plt.title("Fonction de répartition")
plt.bar(I,Fx,width=Max/n)
plt.show()


fx = [0] * n
for k in range(n):
    fx[k] = FreqX[k] * n/Max
plt.figure(6)
plt.clf()
plt.title("Densité de probabilité")
plt.bar(I,fx,width=Max/n)
plt.show()

# 6)
F = lambda x: 1 - np.exp(-0.5*x)
plt.figure(5)
plt.plot(I,F(I),'r')
plt.show()

f = lambda x: 0.5*np.exp(-0.5*x)
plt.figure(6)
plt.plot(I,f(I),'r')
plt.show()
```

Ce code produit les graphiques suivants :