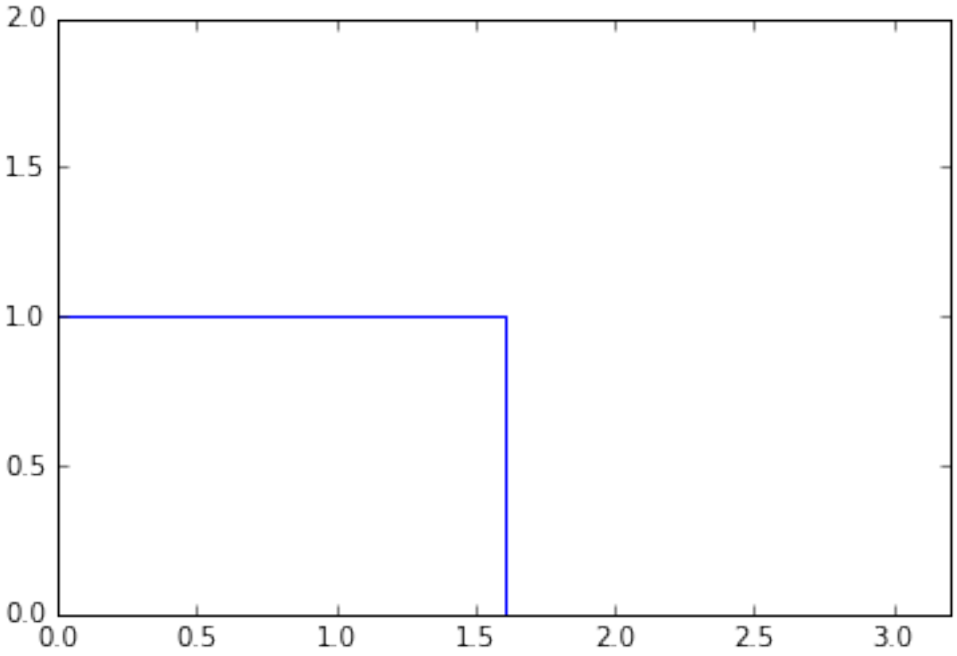
SV3 - Simulation de réactions chimiques

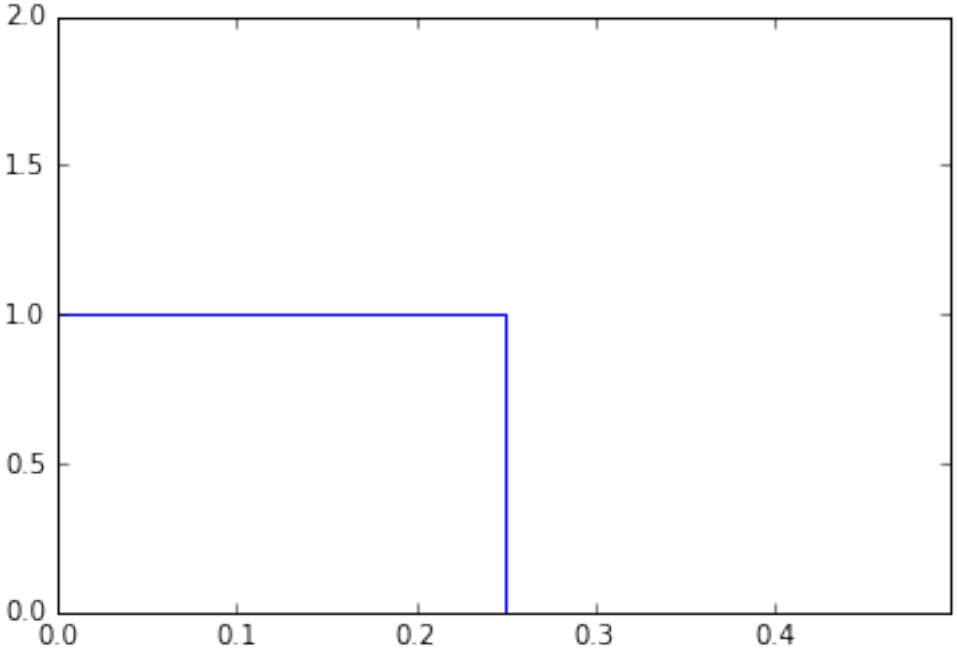
December 5, 2017

In [6]:
```
import numpy as np
import matplotlib.pyplot as plt
import time

%matplotlib inline

def f_time(c, r):
    return (1/c) * np.log(1/r)

def create_graph(t):
    plt.xlim((0, t*2))
    plt.ylim((0, 2))
    plt.plot([0, t, t, 2], [1, 1, 0, 0])
    plt.show()

# 3 graphes
for i in range(3):
    r = np.random.random()
    print("r: {}".format(r))
    create_graph(f_time(1, r))
```

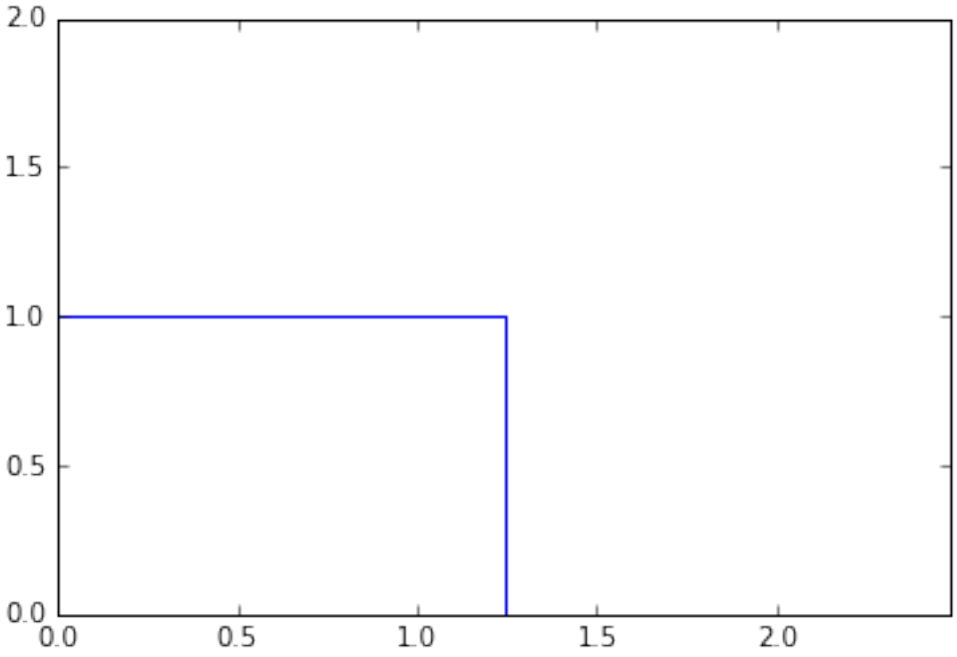
```
r: 0.2013692703144434
```




```
r: 0.7794901476290889
```



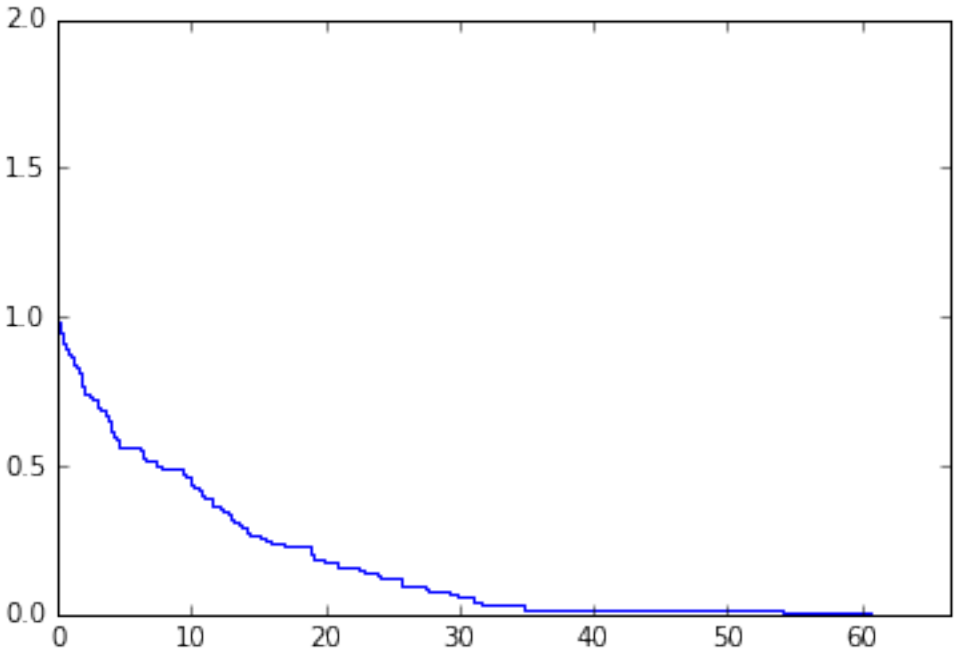
```
r: 0.2890000675271911
```



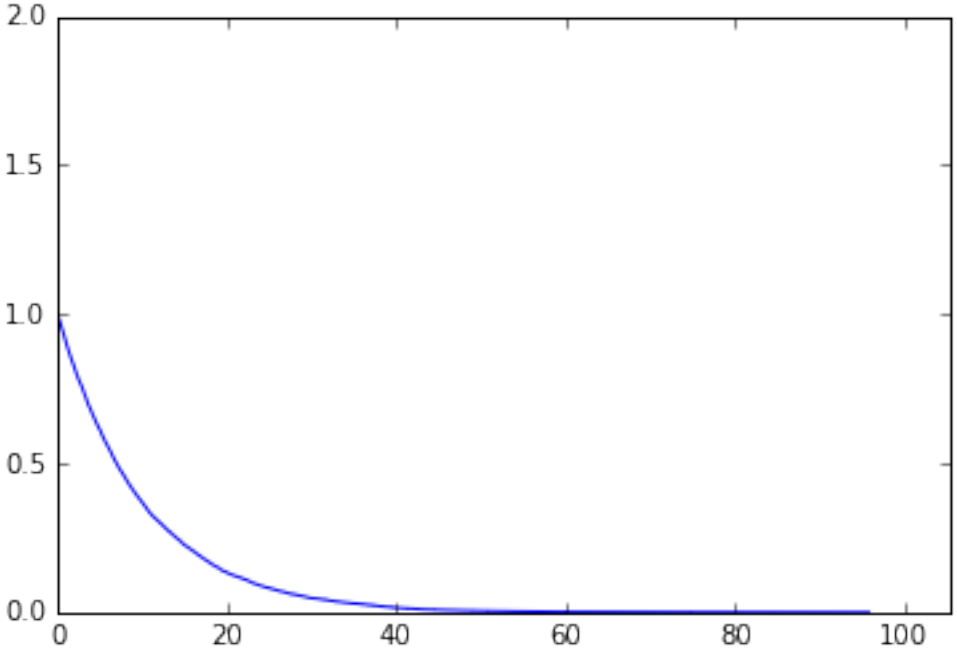

```
In [5]: def create_multi_graphs(g):
            x = g
            y = []
            for i in range(len(x)):
                y.append(1 - (i / x.size))

            plt.xlim((0, np.max(x)*1.1))
            plt.ylim((0, 2))

            # Ajout du premier point
            y = np.concatenate((np.array([1]), y))
            x = np.concatenate((np.array([0]), x))

            # Ajout du dernier point
            y = np.concatenate((y, (np.array([0]))))
            x = np.concatenate((x, (np.array([x[x.size-1]])))))

            plt.step(x, y)
            plt.show()

        # 12 graphes
        cs = [0.1, 1, 100]# 0.1, 1, 100
```

```
nb = [100, 10000, 1000000, 1]

for c in cs:
    for n in nb:
        res = []
        if n == 1: # 1 minute
            t1 = time.time()
            while (time.time() - t1 < 60):
                r = np.random.random()
                res.append(f_time(c, r))
            print("c: {} and nb: {}".format(c, len(res)))
        else:
            for i in range(n):
                r = np.random.random()
                res.append(f_time(c, r))
            print("c: {} and nb: {}".format(c, n))
        res = np.sort(res)
        create_multi_graphs(res)
```

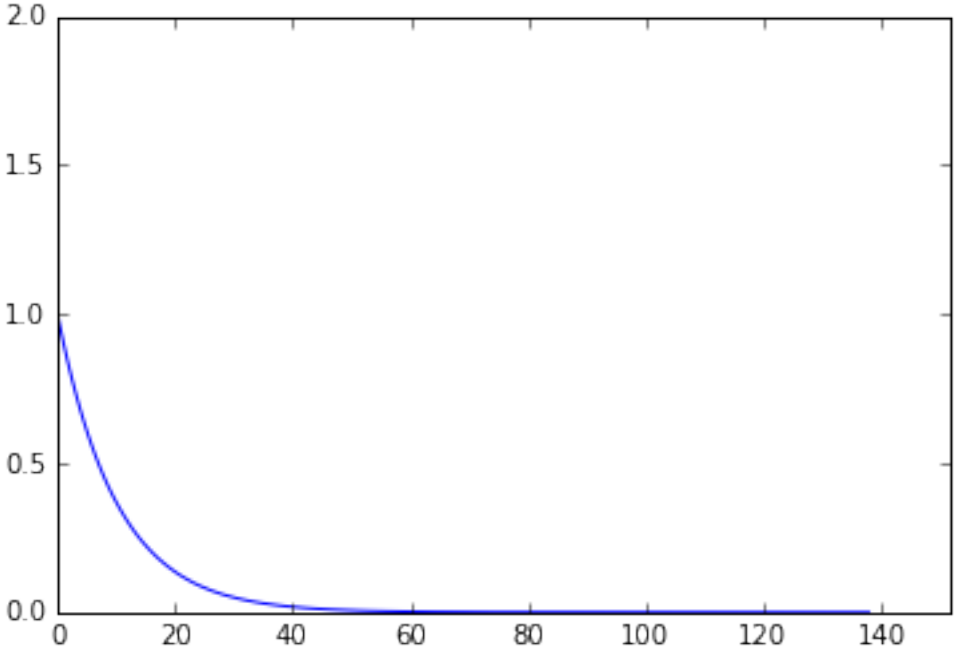
```
c: 0.1 and nb: 100
```



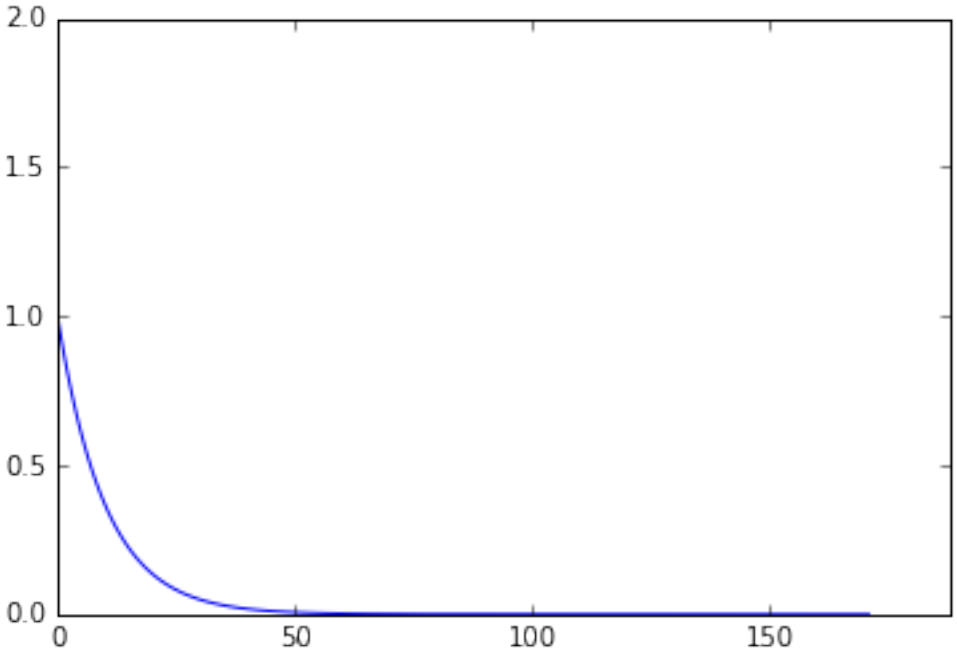

```
c: 0.1 and nb: 10000
```



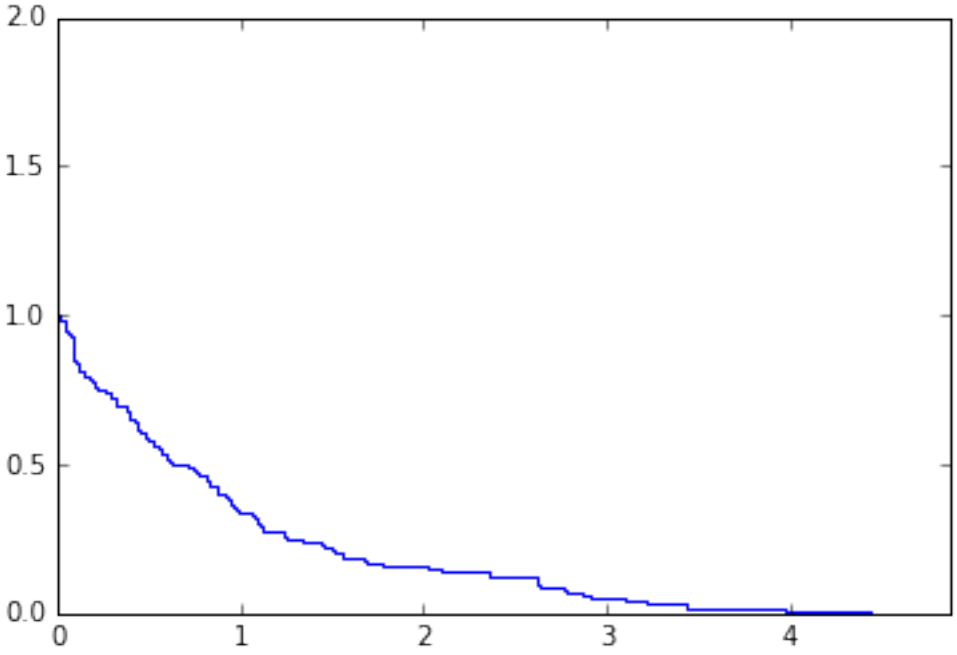

```
c: 0.1 and nb: 1000000
```



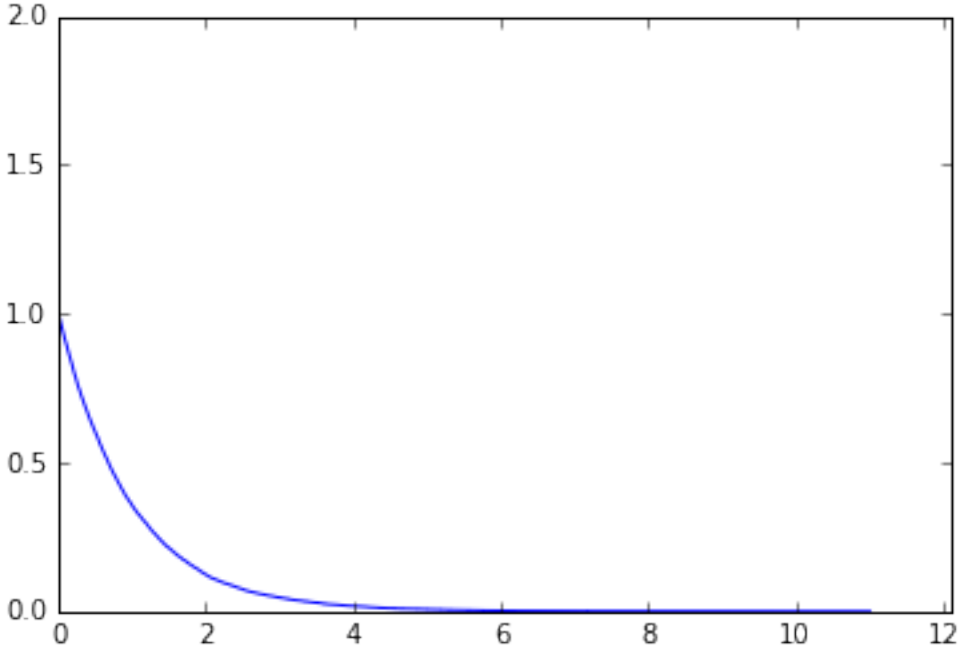
```
c: 0.1 and nb: 42394971
```



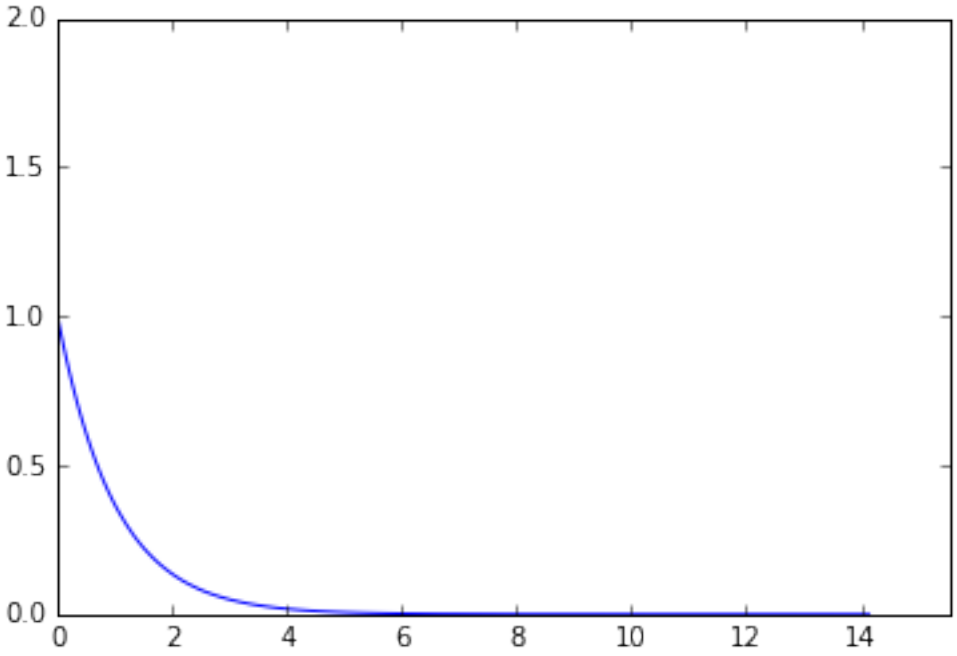

```
c: 1 and nb: 100
```



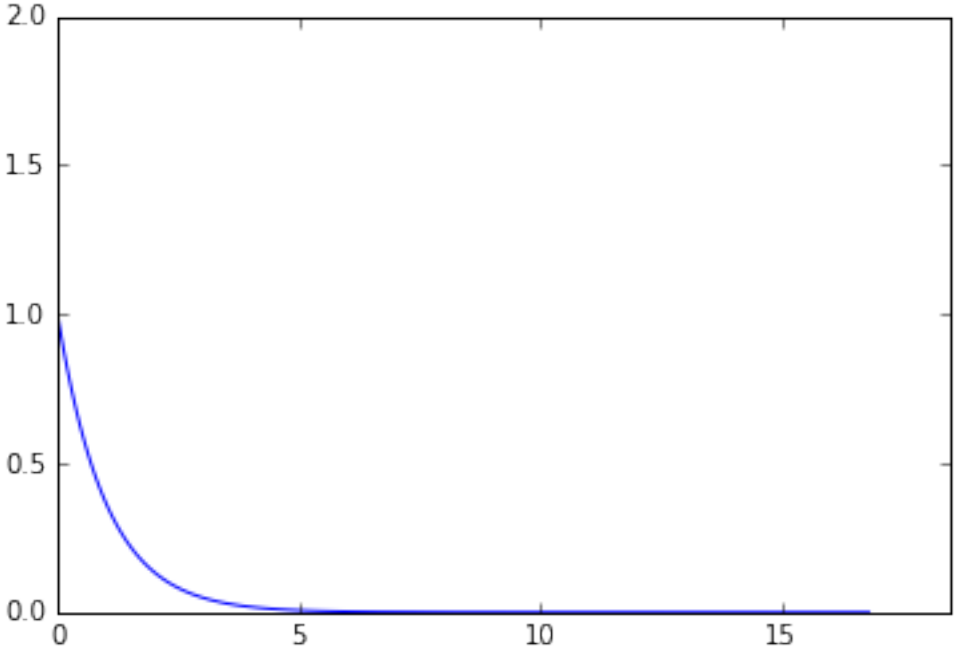
c: 1 and nb: 10000



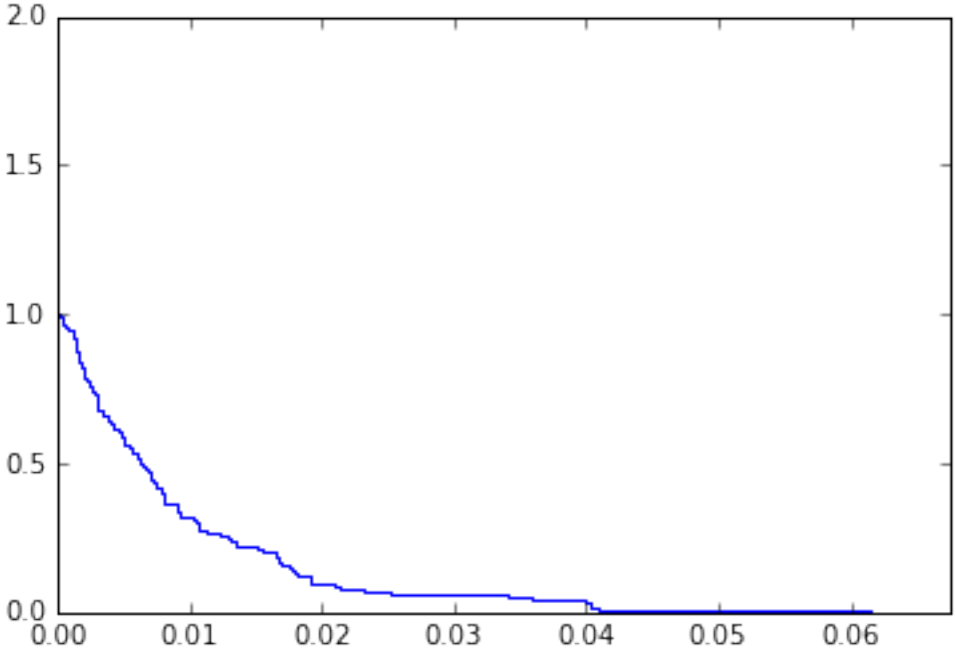
c: 1 and nb: 1000000



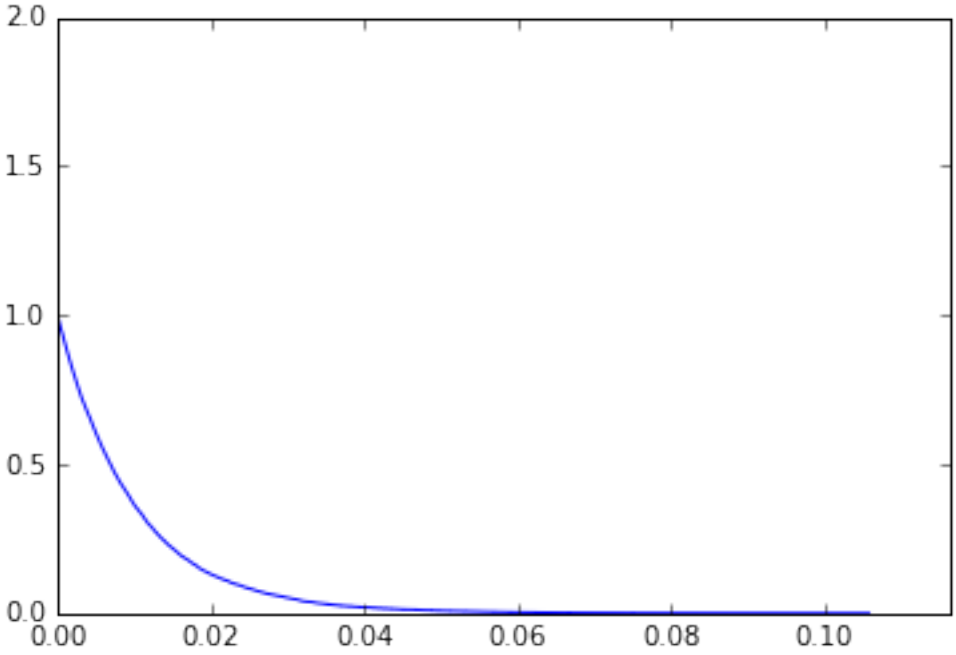

```
c: 1 and nb: 44894685
```



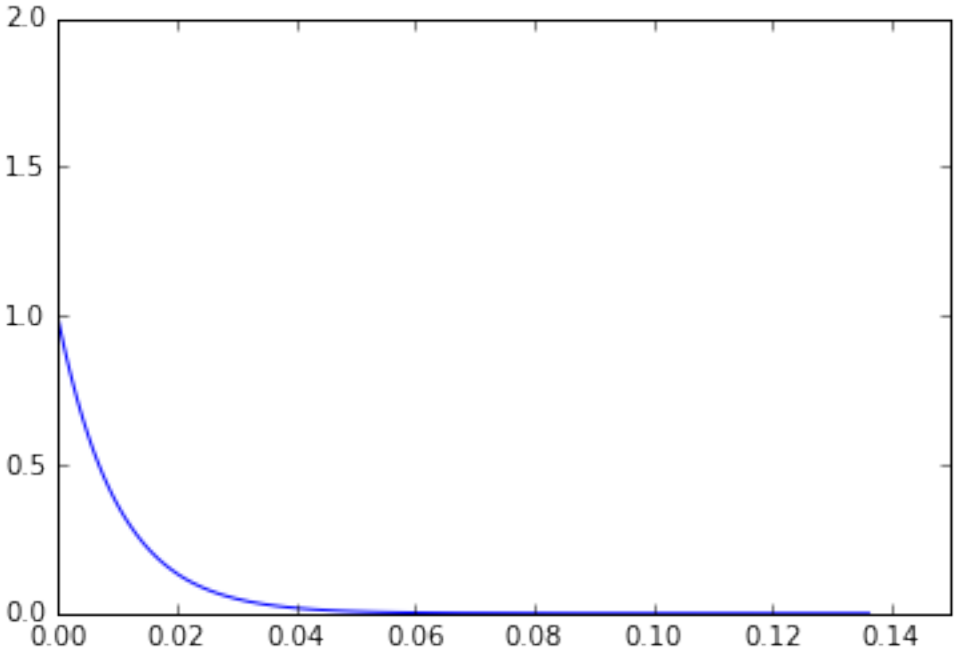
c: 100 and nb: 100



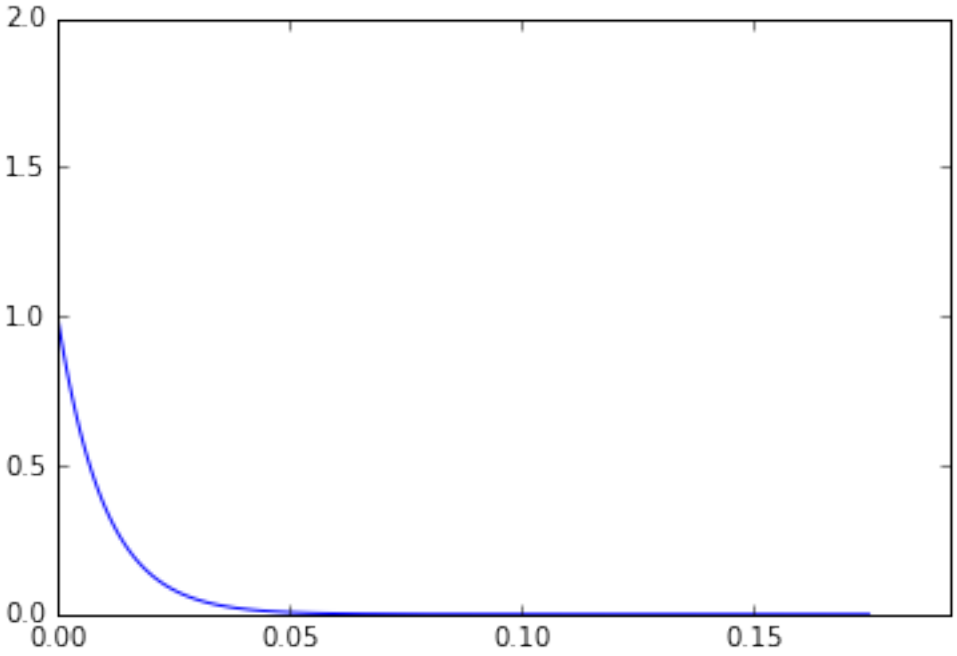
c: 100 and nb: 10000



```
c: 100 and nb: 1000000
```




```
c: 100 and nb: 44918019
```



Comme on peut le voir, plus le nombre d'essais est grand, plus on observe une courbe "lisse" qui converge vers 0.

En une minute, on arrive à faire environ 42 millions de calcul.

La variation du c fait varier simplement l'échelle de temps qu'on peut voir sur les graphes, avec un grand c, on voit que la réaction met beaucoup moins de temps à se déclencher en moyenne.