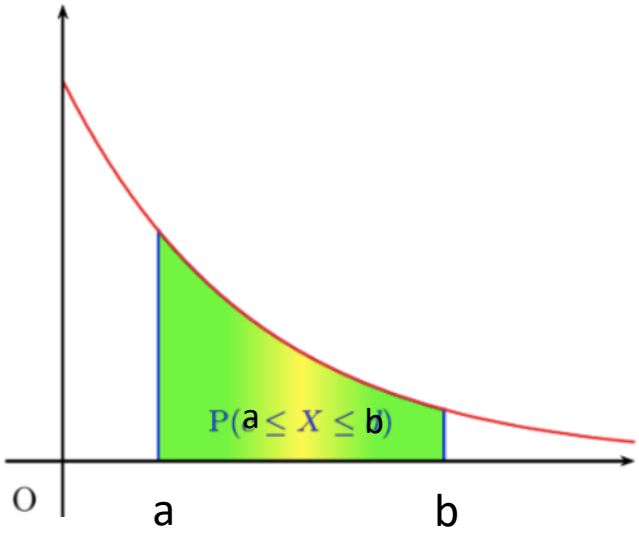
Sts-Ciel 2 Loi exponentielle - Python

On souhaite calculer la probabilité qu'un produit qui ne « vieillit pas » tombe en panne. Prenons ici l'exemple d'une lampe LED. Comme vu en cours, la fonction densité de la loi exponentielle est :

$$f(t) = \lambda e^{-\lambda t} dt$$

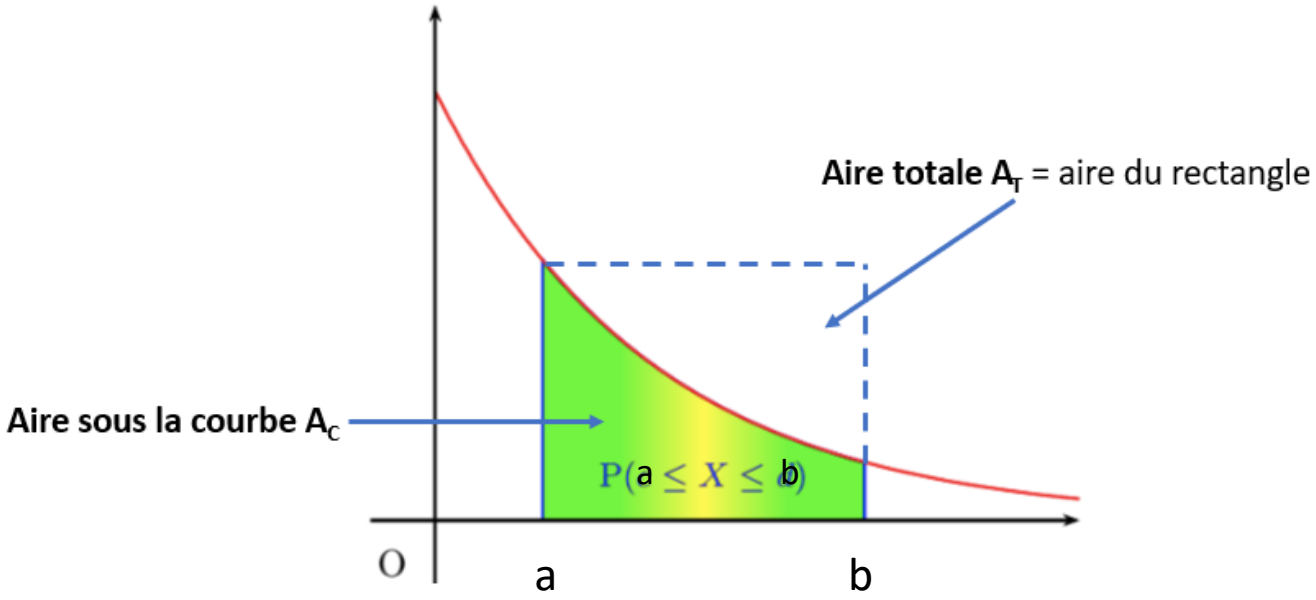
La probabilité que la durée de vie T de la lampe soit comprise entre a et b est : $P\ (a \leq T \leq b)$



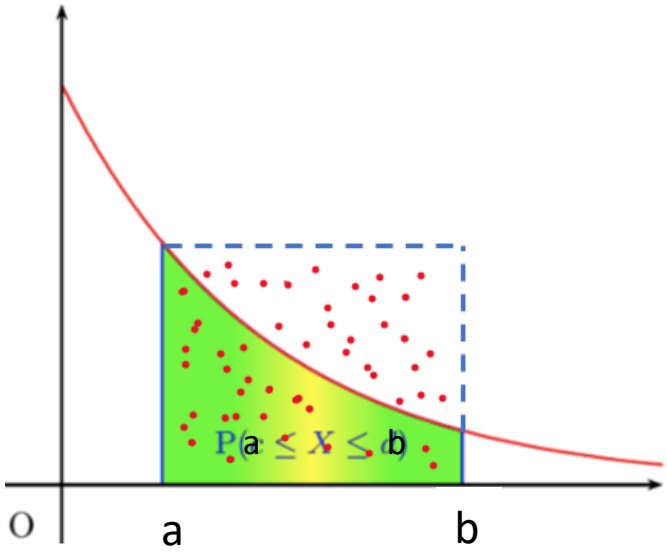
Dans ce TD, nous allons calculer cette probabilité pour T comprise entre 500 et 2000 heures. Pour cela nous utiliserons deux méthodes.

La première méthode est appelée Monte Carlo.

Le but de cette méthode est d'évaluer l'aire de sous la courbe de la loi exponentielle. Pour cela, nous considèrerons que l'aire sous la courbe (entre deux coordonnées a et b) divisée par l'aire totale du rectangle comme expliqué ci-après.



On considère qu'on génère de manière complètement aléatoire, un grand nombre de points, dans le rectangle défini ci-dessous pour $a < x < b$ et $0 < y < f(a)$:



Selon la méthode Monte-Carlo, la probabilité qu'un point généré aléatoirement dans ce rectangle, **soit dans l'aire sous la courbe A_C, est égale à** :

$$P = \frac{A_C}{A_T}$$

On rappelle que l'aire **A_C** sous la courbe correspond à $P(a \leq T \leq b)$ et donc à l'intégrale de la fonction densité $\int_a^b \lambda e^{-\lambda t} dt$. On a donc :

$$A_C = P \times A_T = \int_a^b \lambda e^{-\lambda t} dt$$

⇨ **Ouvrir un nouveau fichier s**ur Pyzo et l'enregistrer sous le nom *loiExponentielle_monNom.py* .

1. FONCTION fonction() :

La fonction *fonction()* donnée ci-dessous est incomplète. Elle prend deux arguments :

t : le temps

ℓ : paramètre de la loi exponentielle : $\ell = \lambda$

Elle renvoie l'image de t pour la fonction densité $f(t) = \lambda e^{-\lambda t} dt$

```
def fonction (t,l):

    return
```

⇨ Ne pas oublier de réaliser l'import du module math : import math. La fonction exponentielle se nomme math.exp()

2. FONCTION montecarlo() :

La fonction *montecarlo()* donnée ci-dessous est incomplète. Elle contient 4 paramètres :
n le nombre de points aléatoires générés, a, b qui sont les temps entre lesquelles nous souhaitons calculer $P(a \leq T \leq b)$ et λ qui sera noté ℓ.
Elle renvoie le produit $\frac{C}{n} \times aire\ totale\ du\ rectangle$, c'est-à-dire le calcul de $P(a \leq T \leq b)$.

```
def montecarlo (n,a,b,l):
    C=0
    ya = ....
    for i in range(n):
        x = ....
        y = ....
        if  y <= .... :
            C = C + 1
    ....
    ....
    return ....
```

x est un nombre aléatoire compris entre a et b. y est un nombre aléatoire compris entre 0 et $y_a = f(a)$
La variable C compte parmi les points générés aléatoirement ceux qui sont situés sous la courbe, c'est-à-dire, lorsque $y < f(x)$.

⇨ N'oubliez pas d'importer le module **random**.

Testez la fonction avec les valeurs suivantes :

```
>>> montecarlo (100000,500,2000,0.0004)
0.3692393823306391
```

3. FONCTION integrale() :

La fonction *integrale()* donnée ci-contre doit être complétée avec la fonction **quad** issue du module SciPy. Il s'agit d'un module de mathématiques.

```
def integrale ( ,  ,  ):

    return quad (   , , ,args=())
```

Quad permet de calculer l'intégrale d'une fonction
entre deux valeurs a et b et éventuellement des arguments supplémentaires :

quad(*func, a, b, args*=())

Cette fonction s'utilise après avoir importé le module **SciPy**

⇨ from scipy.integrate import quad

Dans l'exemple ci-dessus

func est le nom de votre fonction
a et *b* les valeurs entre lesquelles sera réalisée l'intégration de la fonction
args=(x) correspond à ou aux arguments supplémentaires utilisés par la fonction *func*.

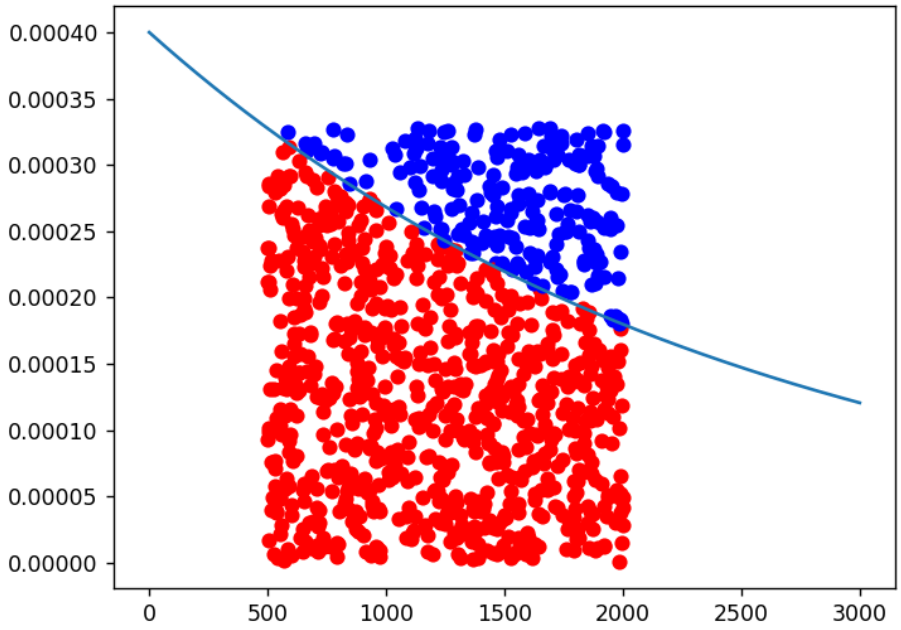
Complétez la fonction integrale et testez-la.

```
>>> integrale(500,2000,0.0004)
(0.3694017889607603, 4.101183714419591e-15)
```

4. Bonus - Fonction trace() :

Nous allons maintenant tracer la courbe exponentielle et représenter les différents points sous et au-dessus de la courbe.

1. Dans un premier temps trace la courbe exponentielle avec les fonctions plt.plot() et plt.show().
2. Dans un second temps, placer les points générés par la fonction random(). Si le point est sous la courbe il sera de couleur rouge, sinon il sera de couleur bleue.



5. Finalisation :

Les fonctions mises au point précédemment vous permettent de calculer une probabilité de variable aléatoire qui suit une loi exponentielle. La méthode de Monté Carlo permet de calculer une intégrale en générant des points de manière aléatoire.
Pour l'exemple d'une loi exponentielle de paramètre $\lambda = 0{,}0004$, la probabilité $P\ (500 \leq T \leq 2000)$ a été calculée avec un nombre de points $n = 100\ 000$. Quel est le nombre n de points nécessaires pour obtenir un résultat $P\ (500 \leq T \leq 2000)$ avec une précision au centième ? (faire les essais nécessaires)

⇨ Rendre votre code dans votre fichier nommé *loiExponentielle_monNom.py* . La réponse à la question précédente sera écrite en commentaire dans ce fichier.

⇨ Déposer ce fichier dans le répertoire *Devoir* du dossier *btsirer* …