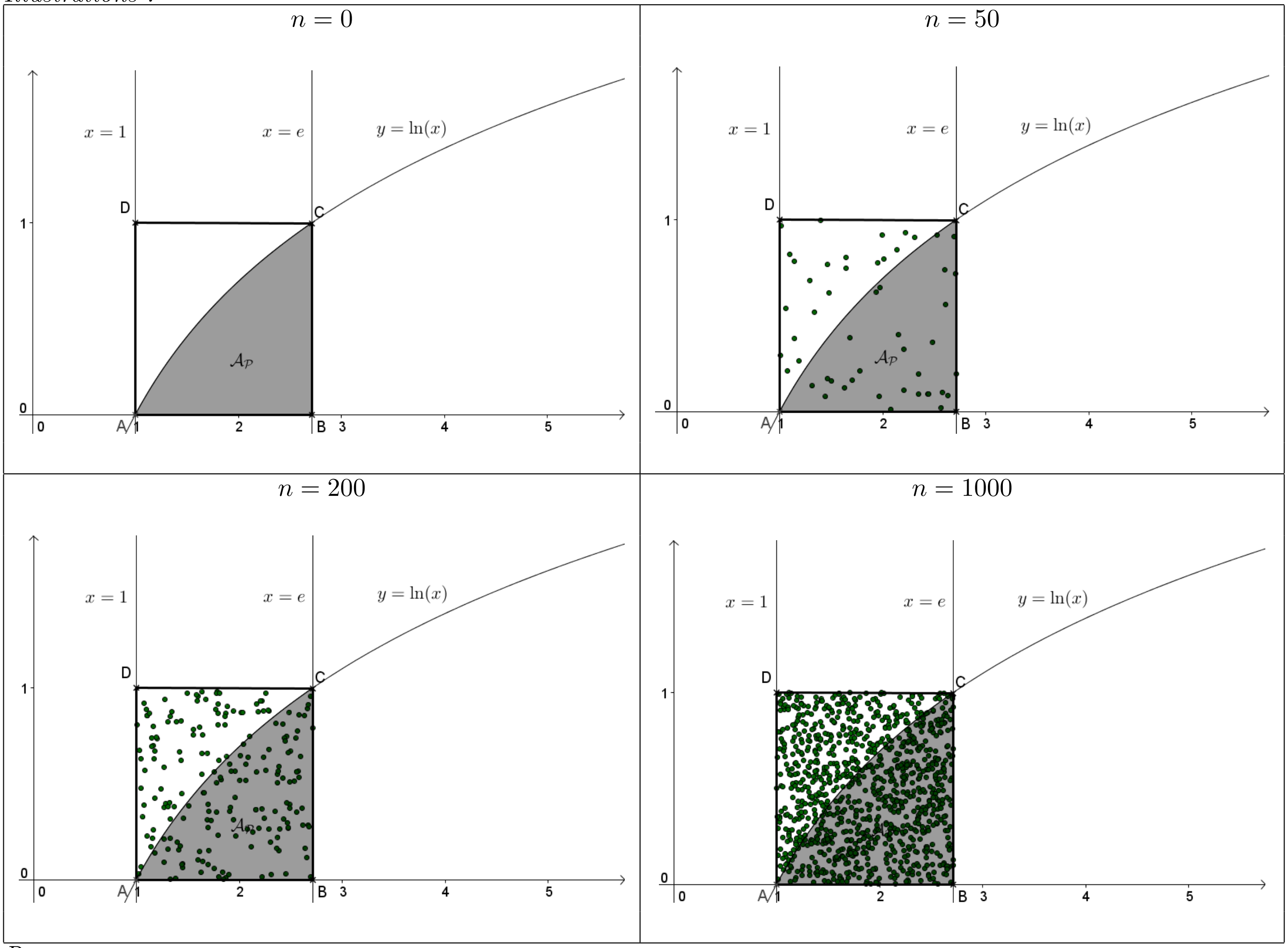
Compétence **modéliser-calculer** : méthode de Monte-Carlo

La méthode de Monte-Carlo consiste à estimer l'aire d'une figure à partir d'algorithme et de probabilité.
Soit un repère orthonormal $\left(O\ ;\ \vec{i}\ ;\ \vec{j}\right)$.
Le but de cet introduction est de déterminer l'aire $\mathcal{A}_{\mathcal{P}}$ de la partie $\mathcal{P}$ délimitée par les droites $y = 0$, $x = 1$ et $x = e$ et la courbe d'équation $y = \ln(x)$.
Pour ce faire on place n points au hasard dans le rectangle ABCD de côté 1 et e, d'aire e (unité d'aire, u.a).
La proportion de points dans la partie $\mathcal{P}$, $\dfrac{c}{n}$ où c est le nombre de points dans la partie $\mathcal{P}$ est ainsi proche de $\dfrac{\text{aire de } \mathcal{P}}{\text{aire du rectangle ABCD}} = \dfrac{\mathcal{A}_{\mathcal{P}}}{e-1}$. Ainsi $\mathcal{A}_{\mathcal{P}} \simeq \dfrac{c.(e-1)}{n}$.

Illustrations :

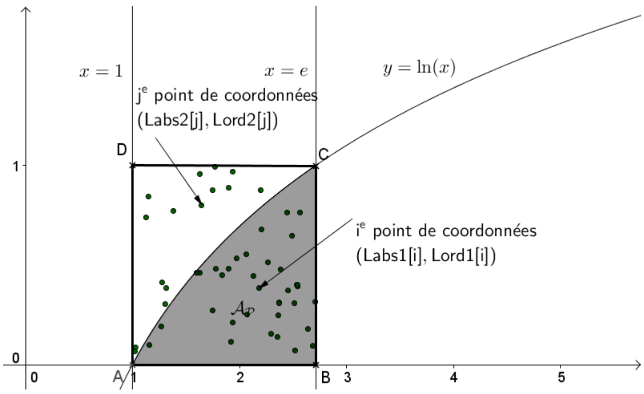


Remarque :
Plus le nombre de points est grand plus l'approximation $\dfrac{c.e}{n}$ est proche de $\mathcal{A}_{\mathcal{P}}$.

Modélisation de la méthode de Monte-Carlo sur Python

Pour distinguer les n points de ceux qui sont dans la partie $\mathcal{A}_{\mathcal{P}}$ de ceux qui ne le sont pas on va construire 4 listes Labs1, Lord1 pour les listes des abscisses et ordonnées des points qui sont dans $\mathcal{A}_{\mathcal{P}}$, ainsi un i^e point des n points qui est dans la partie $\mathcal{A}_{\mathcal{P}}$ a pour coordonnées (Labs1[i],Lord1[i]).
Les coordonnées des autres points sont repérés par les listes Labs2 et Lord2.

Illustration :



a est un nombre au hasard de l'intervalle $[1\ ;\ e]$ et b un nombre au hasard de l'intervalle $[0\ ;\ 1]$.

1. À quelle condition le point $(a\ ;\ b)$ est un point de la partie $\mathcal{A}_{\mathcal{P}}$?
2. Sur Python, recopier et compléter le programme suivant :

```
#méthode de Monte_Carlo

from math import*
from random import*
import matplotlib.pyplot as plt



n=10000
Labs1=[]
Lord1=[]
Labs2=[]
Lord2=[]
c=0
for i in range(0,n):
    a=random()*(exp(1)-1)+1
    b=random()
    if ______________:
        Labs1.append(___)
        Lord1.append(___)
        c=________________
    else :
        Labs2.append(___)
        Lord2.append(___)


print("aire approchée :", _____________)

#représentaiton graphique
#construction des n points
plt.plot(Labs1, Lord1, 'rx') #en rouge et représenté par une croix x
plt.plot(Labs2, Lord2, 'bx') #en bleu et représenté par une croix x
#Construction des axes abscisses et ordonnées
plt.axis([0, 3, 0, 2])
plt.show()
```

methode_monte_carlo_eleve.py

3. (a) Exécuter le programme et conjecturer la valeur de $\displaystyle\int_1^e \ln(x)\mathrm{d}x$.

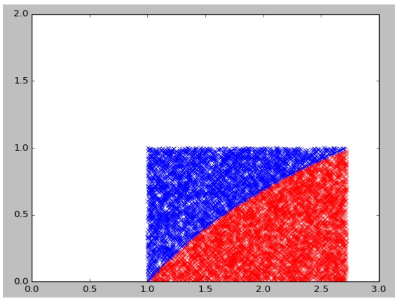
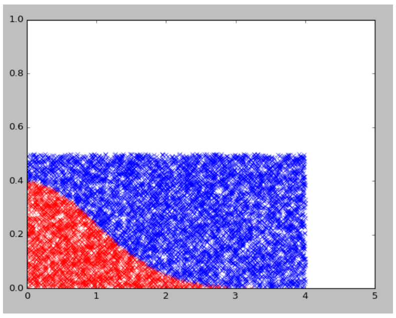
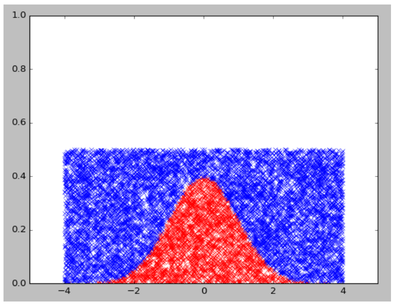
   (b) Montrer que la fonction F définie par $F(x) = x\ln(x) - x$ et dérivable sur $]0\ ;\ +\infty[$ est une primitive de ln.

   (c) En déduire la valeur de $\displaystyle\int_1^e \ln(x)\mathrm{d}x$.

Applications : une loi continue, fonction de densité de la loi Normale Soit la fonction f définie et continue sur $\mathbb{R}$ par $f(x) = \dfrac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2}$.

1. Montrer que la fonction f est paire.
2. Déterminer $\lim\limits_{x \to +\infty} f(x)$.
3. Déterminer le maximum de $f(x)$ sur $\mathbb{R}$.
4. Par la méthode de Monte-Carlo, sur Python, écrire un programme qui permet de justifier que $\displaystyle\int_0^{+\infty} f(x)\mathrm{d}x = 0,5$.
   On pourra restreindre la conjecture à l'intervalle $[0\ ;\ 4]$ et utiliser le rectangle définit par les points de coordonnées $(0\ ;\ 0)$, $(4\ ;\ 0)$, $(4\ ;\ 0,5)$ et $(0\ ;\ 0,5)$. (On pourrait aussi remplacer l'ordonnée $0,5$ par le maximum de $f(x)$.)
5. En admettant la conjecture précédente, justifier que la fonction f est une fonction de densité.

Annexe :
Graphiques sur Python :

$f(x) = \ln(x)$	$f(x) = \dfrac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2}$ sur $[0\ ;\ +\infty[$	$f(x) = \dfrac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2}$ sur $]-\infty\ ;\ +\infty[$
		
$\mathcal{A}_{\mathcal{P}} = 1$	$\mathcal{A}_{\mathcal{P}} = 0.5$	$\mathcal{A}_{\mathcal{P}} = 1$