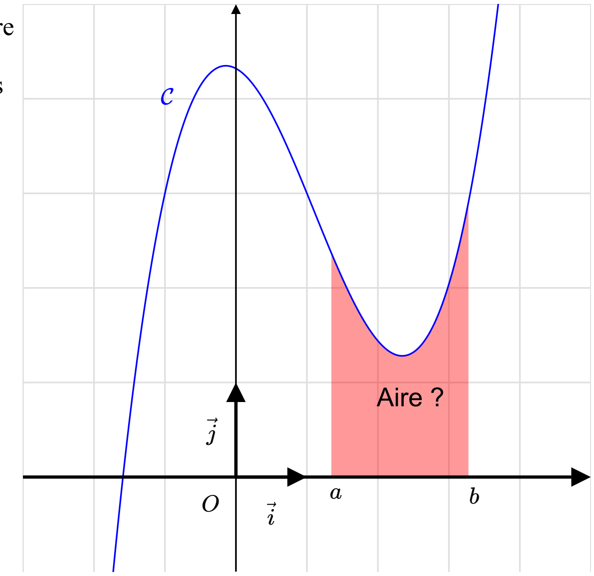
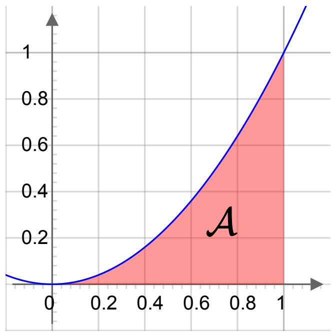
DM ∼ Aire sous une courbe

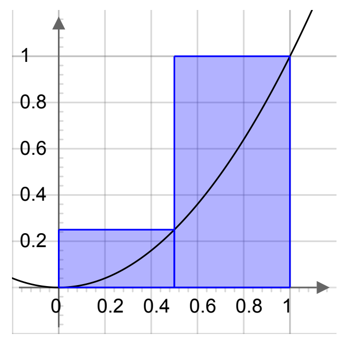
Nous savons calculer les aires de certaines figures simples : rectangles, triangles, trapèzes etc. La formule de l'aire d'un disque est connue mais assez difficile à démontrer, alors que pour les figures précédentes tout découle de l'aire du rectangle que l'on découpe de manière appropriée.
Question : Comment trouver l'aire entre une courbe représentative de fonction et l'axe des abscisses ?
Nous allons nous intéresser ici à la courbe représentative de la fonction carrée sur l'intervalle $[0;1]$. On note $\mathcal{A}$ l'aire entre la courbe, l'axe des abscisses et les droites d'équation $x=0$ et $x=1$.



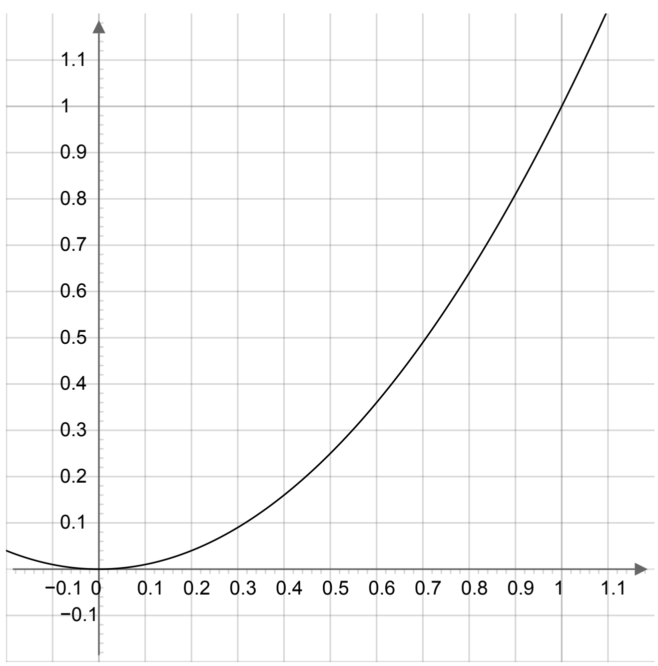


1 Première approximation

1. Nous allons, pour commencer, trouver un majorant de $\mathcal{A}$.
   a. Trouver l'aire de chacun des deux rectangles de la figure ci-contre et trouver ainsi un majorant de $\mathcal{A}$.
   b. Dans la question précédente, nous avons trouver des rectangles qui se trouvent « au dessus » de la courbe et dont les sommets en haut à droite sont des points de la courbe.
   Trouver par une méthode équivalente, en considérant des rectangles qui se trouvent « au dessous » de la courbe, une minoration de $\mathcal{A}$.



2. L'encadrement précédent n'est pas assez précis, nous allons essayer de faire mieux.
   a. Dans le repère ci-dessous, construire quatre rectangles « au dessus » de la courbe uniformément répartis sur $[0\,;1]$, dont les sommets supérieurs droits sont des points de la courbe.



   b. Déterminer ensuite un ecandrement plus précis que celui de la question 1.

2 À l'aide d'un algorithme

1. On utilise dans cet question un algorithme qui généralise la méthode précédente en utilisant 10 rectangles.
   Compléter la fonction *aireInf* pour qu'elle retourne la minoration obtenue par la même méthode que *aireSup*.

```
def f(x):
    return x**2

def aireSup():
    x = 0
    pas = 1.0/10
    aire = 0
    for i in range(0,10):
        x = x+pas
        aire = aire+pas*f(x)
    return aire

def aireInf():
    x = 0










print(aireSup())
print(aireInf())
```

2. On souhaite écrire un algorithme dans lequel la fonction *aireSup* prend comme paramètre un nombre n qui correspond au nombre de rectangles de la méthode utilisée.
   Compléter le code ci-dessous dans ce but.
3. En utilisant ce dernier algorithme avec des valeurs de n très grandes, quelle conjecture peut-on émettre sur la valeur de $\mathcal{A}$?

```
def f(x):
    return x**2

def aireSup(n):
    x =
    pas = 1.0/
    aire = 0
    for i in range( , ):
        x =
        aire =
    return aire

print(aireSup(2))
print(aireSup(4))
print(aireSup(10))
print(aireSup(100))
```

3 À l'aide de deux suites

Soit n un entier naturel non nul. On partage l'intervalle $[0\,;1]$ en n intervalles de même longueur, en le découpant aux points d'abscisses :

$$0; \frac{1}{n}; \frac{2}{n}; \cdots ; \frac{k}{n}; \cdots ; \frac{n-1}{n}; \frac{n}{n} = 1.$$

On construit à partir de ces points n rectangles supérieurs et n rectangles inférieurs encadrant la courbe représentative de la fonction carrée avec la méthode étudiée précédemment.
On note i_n la somme des aires des rectangles inférieurs et s_n celle des rectangles supérieurs. Nous avons alors que : $i_n \leq \mathcal{A} \leq s_n$.

1. Démontrer que $i_n = \dfrac{1}{n^3}\displaystyle\sum_{k=0}^{n-1} k^2$ et $s_n = \dfrac{1}{n^3}\displaystyle\sum_{k=0}^{n-1}(k+1)^2$.
2. On rappelle que :
   $$\sum_{k=0}^{n} k = \frac{n(n+1)}{2} \text{ et } \sum_{k=0}^{n} k^2 = \frac{(2n+1)n(n-1)}{6}.$$
   En déduire que $i_n = \dfrac{(2n-1)(n-1)}{6n^2}$ et $s_n = \dfrac{(2n+1)(n+1)}{6n^2}$.
3. Trouver alors les limites de ces deux suites et en déduire la valeur de $\mathcal{A}$.