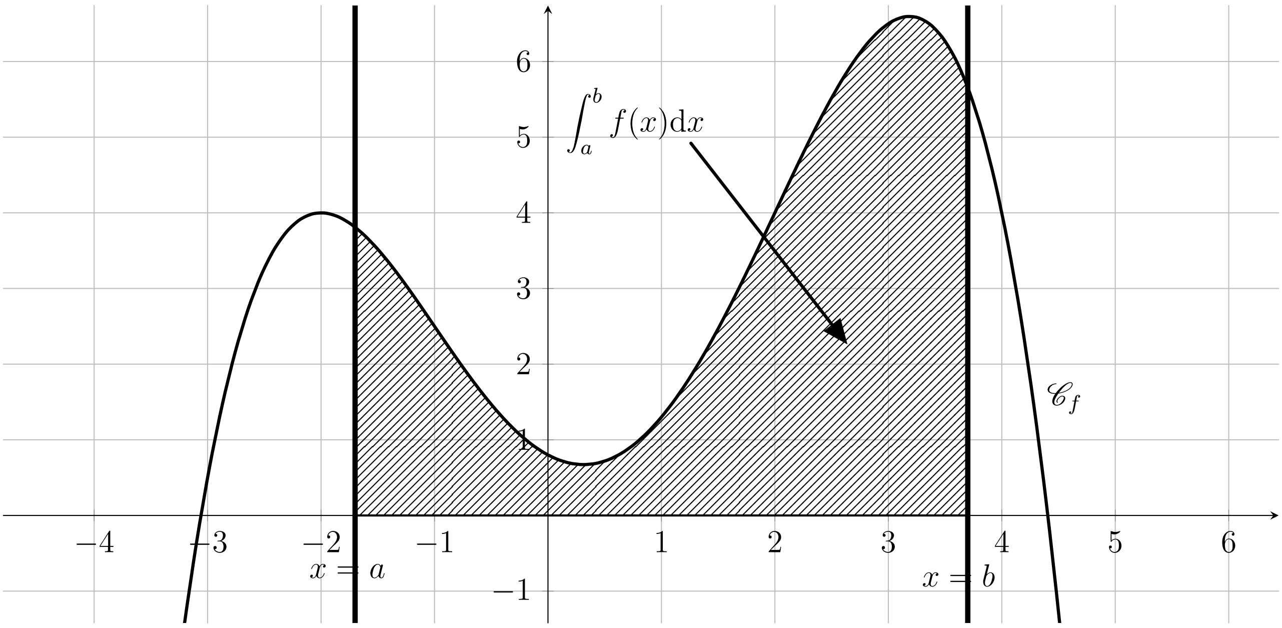
⧫ TP3 – Calculs approchés d'une intégrale

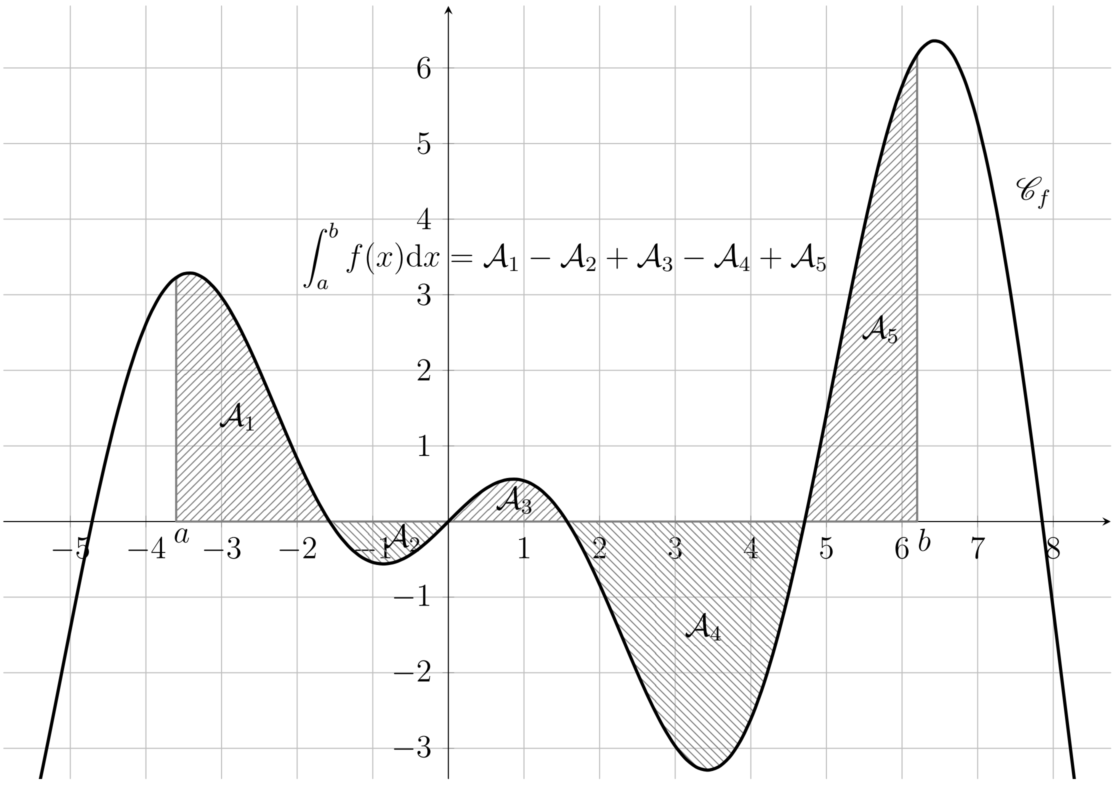
I. — Introduction

Dans tout ce qui suit, on considère une fonction continue sur un intervalle $[a\,;b]$. On munit le plan d'un repère orthogonal $\left(\mathrm{O}\,;\vec{i},\vec{j}\right)$ et on note $\mathscr{C}_f$ la courbe représentative de f dans le repère $\left(\mathrm{O}\,;\vec{i},\vec{j}\right)$.

Si f est positive alors, par définition, l'intégrale de a et b de f est l'aire sous la courbe de f entre a et b i.e. l'aire, exprimée en unité d'aire, de la partie du plan délimitée par l'axe des abscisses, la courbe $\mathcal{C}_f$ et les droites d'équations $x=a$ et $x=b$.



Si f change de signe, l'intégrale est la somme des aires algébriques i.e. la somme des aires comptées avec un signe $+$ lorsque f est positive et avec un signe $-$ lorsque f est négative.



On sait qu'on peut calculer l'intégrale d'une fonction continue à l'aide d'une primitive. Cependant, il n'est pas toujours simple, voire possible, de déterminer la primitive d'une fonction continue sur un intervalle. On peut, par exemple, démontrer que la fonction $x \longmapsto \mathrm{e}^{-x^2}$ admet des primitives sur $\mathbb{R}$ mais qu'il est impossible de les exprimer à l'aide des fonctions de référence et des opérations usuelles.

Il est donc nécessaire de disposer de méthodes pour calculer des intégrales de façons approchées.

L'idée générale des méthodes qui suivent est de se donner un entier naturel n non nul, de découper l'intervalle $[a\,;b]$ en n intervalles de longueur $\dfrac{b-a}{n}$ et d'approcher, sur chacun de ces intervalles, l'intégrale de la fonction f par l'aire algébrique d'une forme plus simple (des rectangles dans le premier cas et des trapèzes dans le second).

Plus précisément, on fixe $n \in \mathbb{N}^*$ et on pose, pour tout $k \in [\![0, n]\!]$, $x_k = a + k\dfrac{b-a}{n}$. Ainsi, $x_0 = a$, $x_n = b$ et, pour tout $k \in [\![0, n-1]\!]$,

$$x_{k+1} - x_k = a + (k+1)\frac{b-a}{n} - \left(a + k\frac{b-a}{n}\right) = (k+1-k)\frac{b-a}{n} = \frac{b-a}{n}$$

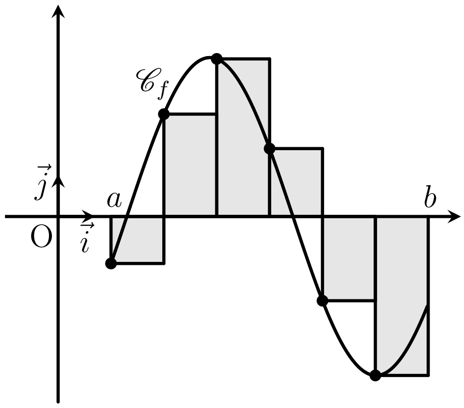
donc $[x_k\,;x_{k+1}]$ est un intervalle de longueur $\dfrac{b-a}{n}$. Si, sur chacun des intervalles $[x_k\,;x_{k+1}]$, on approche l'intégrale de f par l'aire d'une forme F_k alors, par la relation de Chasles,

$$\int_a^b f(x)\,\mathrm{d}x = \sum_{k=0}^{n-1} \int_{x_k}^{x_{k+1}} f(x)\,\mathrm{d}x \approx \sum_{k=0}^{n-1} \mathrm{aire}(F_k).$$

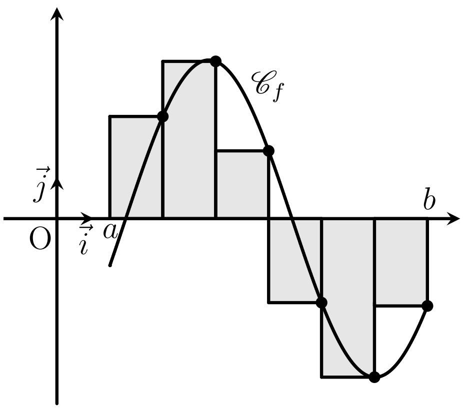
On peut démontrer que lorsque n tend vers $+\infty$, ces approximations convergent vers l'intégrale de f entre a et b. Ainsi, pour n suffisamment grand, on obtient de bonnes approximations de l'intégrale considérée.

II. — Méthode des rectangles

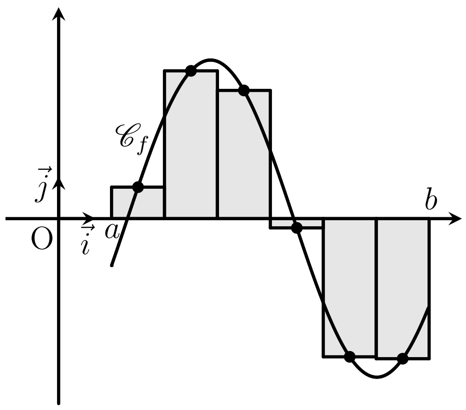
Dans cette méthode, l'idée est d'approcher, sur chaque intervalle $[x_k\,;x_{k+1}]$, la fonction f par une fonction constante ce qui revient à approcher l'intégrale de f par l'aire d'un rectangle F_k. La largeur de ce rectangle est $x_{k+1} - x_k = \dfrac{b-a}{n}$ et, pour sa hauteur algébrique (i.e. positive ou négative selon le cas), il y a plusieurs choix possibles. On peut prendre $f(x_k)$ et on obtient alors les *rectangles à gauche*, on peut prendre $f(x_{k+1})$ et on obtient les *rectangles à droite* ou on peut prendre $f(\frac{x_k+x_{k+1}}{2})$ et on obtient les *rectangles au centre*.



rectangles à gauche avec $n = 6$



rectangles à droite avec $n = 6$



rectangles au centre avec $n = 6$

1) Un premier exemple

Dans tout ce paragraphe, on considère la fonction $f : x \longmapsto x^3 - 7x^2 + 14x - 7$ définie sur $\mathbb{R}$ et $I = \displaystyle\int_{0,5}^{4} f(x)\,\mathrm{d}x$.

Ici, on sait calculer explicitement I donc, en soi, un calcul approché n'a pas d'intérêt. On considère cet exemple uniquement pour s'approprier la méthode des rectangles et pour comparer les différents résultats obtenus avec la valeur exacte.

1. Calculer I à l'aide d'une primitive de f.

Solution.

$$I = \int_{0,5}^{4} x^3 - 7x^2 + 14x - 7\,\mathrm{d}x = \left[\frac{x^4}{4} - \frac{7}{3}x^3 + 7x^2 - 7x\right]_{0,5}^{4}$$
$$= \frac{4^4}{4} - \frac{7}{3} \times 4^3 + 7 \times 4^2 - 7 \times 4 - \left(\frac{0,5^4}{4} - \frac{7}{3} \times 0,5^3 + 7 \times 0,5^2 - 7 \times 0,5\right)$$

soit $\boxed{I = \dfrac{133}{192}}$.

2. On souhaite illustrer la méthode des rectangles avec $n = 7$.

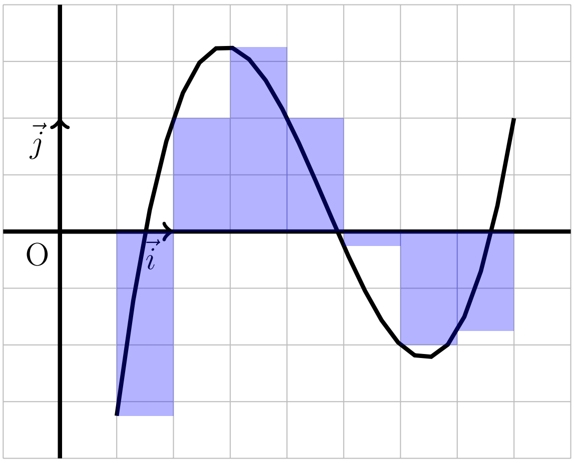
a. Que valent a et b ici ?

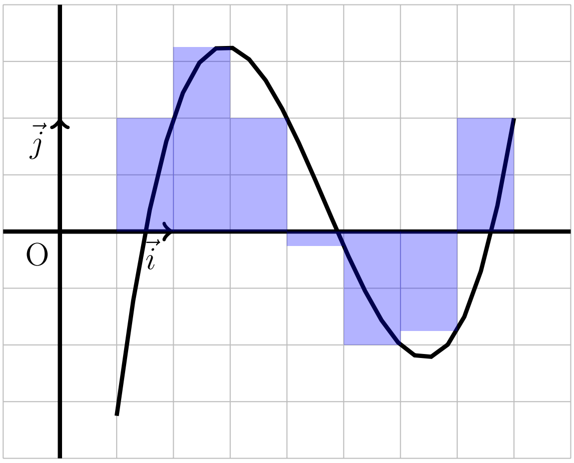
Solution. Ici, $a = 0,5$ et $b = 4$.

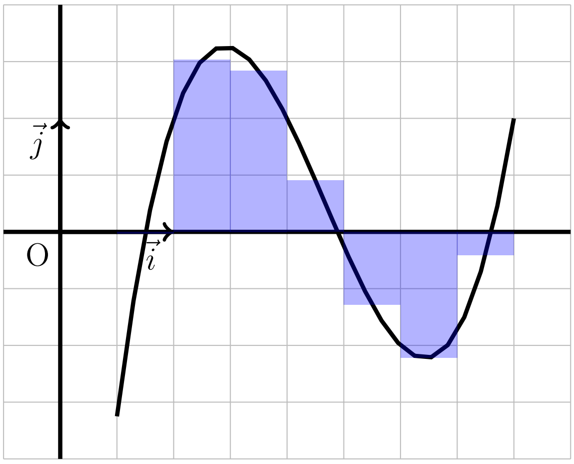
b. Déterminer x_0, x_1, x_2, x_3, x_4, x_5, x_6 et x_7.

Solution. Ici $\frac{b-a}{n} = \frac{4-0,5}{7} = \frac{3,5}{7} = 0,5$ donc $x_0 = 0,5$, $x_1 = 0,5 + 0,5 = 1$, $x_2 = 0,5+2\times0,5 = 1,5$, $x_3 = 0,5+3\times0,5 = 2$, $x_4 = 0,5+4\times0,5 = 2,5$, $x_5 = 0,5+5\times0,5 = 3$, $x_6 = 0,5 + 6 \times 0,5 = 3,5$ et $x_7 = 0,5 + 7 \times 0,5 = 4$.

c. On a représenté ci-dessous 3 exemplaires de la courbe de f sur $[0,5\,;4]$. Construire sur le premier graphique les rectangles à gauche, sur le second les rectangles à droite et sur le troisième les rectangles au centre.







d. Calculer, dans chaque cas, la somme des aires algébriques des rectangles et en déduire trois approximations de I.

Solution. Dans tous les cas, la base des rectangles est $\frac{b-a}{n} = 0{,}5$.
Dans le cas des rectangles à gauche, la hauteur algébrique des rectangles est la valeur de f au début de l'intervalle donc la somme des aires des rectangles est

$$0{,}5f(0{,}5) + 0{,}5f(1) + 0{,}5f(1{,}5) + 0{,}5f(2) + 0{,}5f(2{,}5) + 0{,}5f(3) + 0{,}5f(3{,}5) = 1{,}3125$$

Dans le cas des rectangles à droite, la hauteur algébrique des rectangles est la valeur de f au début de l'intervalle donc la somme des aires des rectangles est

$$0{,}5f(1) + 0{,}5f(1{,}5) + 0{,}5f(2) + 0{,}5f(2{,}5) + 0{,}5f(3) + 0{,}5f(3{,}5) + 0{,}5f(4) = 1{,}3125$$

Dans le cas des rectangles au centre, la hauteur algébrique des rectangles est la valeur de f au centre de l'intervale donc la somme des aires des rectangles est

$$0{,}5f\left(\frac{0{,}5+1}{2}\right) + 0{,}5f\left(\frac{1+1{,}5}{2}\right) + 0{,}5f\left(\frac{1{,}5+2}{2}\right) + 0{,}5f\left(\frac{2+2{,}5}{2}\right) + 0{,}5f\left(\frac{2{,}5+3}{2}\right)$$
$$+0{,}5f\left(\frac{3+3{,}5}{2}\right) + 0{,}5f\left(\frac{3{,}5+4}{2}\right) = 0{,}7109375$$

e. Quel choix de rectangles donne la meilleure approximation ?

Solution. Comme $I = \dfrac{133}{192} \approx 0{,}69$, le choix qui donne la meilleure approximation est celui des rectangles au centre.

3. Sans tracer les rectangles, calculer les approximations de I en utilisant les 3 types de rectangles mais en prenant cette fois-ci $n = 10$.

Solution. Si $n = 10$, la base des rectangles des $\frac{4-0{,}5}{10} = 0{,}35$. On en déduit qu'on obtient :

avec les rectangles à gauche : $\displaystyle\sum_{k=0}^{9} 0{,}35f(0{,}5 + k \times 0{,}35) = 0{,}21546875$

avec les rectangles à droite : $\displaystyle\sum_{k=1}^{10} 0{,}35f(0{,}5 + k \times 0{,}35) = 1{,}13421875$

avec les rectangles au centre : $\displaystyle\sum_{k=0}^{9} 0{,}35f\left(\frac{0{,}5 + k \times 0{,}35 + 0{,}5 + (k+1) * 0{,}35}{2}\right) = 0{,}701640625$

4. On voit que les calculs deviennent vite fastidieux et il est donc avantageux de programmer la méthode à l'aide de Python.

a. Définir une fonction `f` qui prend en argument un réel x et qui renvoie la valeur de $f(x)$ i.e. $x^3 - 7x^2 + 14x - 7$.

Solution.

```
def f(x):
  return x**3 - 7*x**2 + 14*x - 7
```

b. La fonction suivante renvoie la valeur approchée de I obtenue à l'aide des rectangles à gauche pour une valeur de n passée en argument.

```
def rectangles_gauche(n):
  S = 0
  for k in range(n):
    S += 3.5/n*f(0.5+k*3.5/n)
  return S
```

Implémenter cette fonction et vérifier les résultats trouvés aux questions **2.d.** et **3.**.

Solution.

```
> rectangles_gauche(7)
0.0
> rectangles_gauche(10)
0.21546874999999974
```

On retrouve bien les mêmes résultats que dans les questions **2.d.** et **3.**.

c. En s'inspirant du code de la fonction `rectangles_gauche`, écrire une fonction `rectangles_droite` et une fonction `rectangles_centre` qui prennent en argument une entier n et qui renvoient les valeurs approchées de I obtenues en considérant respectivement n rectangles à droite et n rectangles au centre.

Solution. Pour les rectangles à droite, on peut écrire

```
def rectangles_droite(n):
  S = 0
  for k in range(1,n+1):
    S += 3.5/n*f(0.5+k*3.5/n)
  return S
```

ou

```
def rectangles_droite(n):
  S = 0
  for k in range(n):
    S += 3.5/n*f(0.5+(k+1)*3.5/n)
  return S
```

Pour les rectangles au centre, on peut écrire

```
def rectangles_centre(n):
  S = 0
  for k in range(n):
    S += 3.5/n*f((0.5+k*3.5/n+0.5+(k+1)*3.5/n)/2)
  return S
```

d. Écrire trois fonctions `seuil_gauche`, `seuil_droite` et `seuil_centre` prenant en argument un entier naturel p et renvoyant le nombre minimal de rectangles nécessaires pour obtenir une valeur approchée de I à 10^{-p} près en utilisant respectivement les rectangles à gauche, les rectangles à droite et les rectangles au centre.

Solution.

```
def seuil_gauche(p):
  n=1
  while abs(133/192-rectangles_gauche(n)) > 10**(-p):
    n +=1
  return n

def seuil_droite(p):
  n=1
  while abs(133/192-rectangles_droite(n)) > 10**(-p):
    n +=1
  return n

def seuil_centre(p):
  n=1
  while abs(133/192-rectangles_centre(n)) > 10**(-p):
      n +=1
  return n
```

e. Afficher les résultats obtenus en appelant les trois fonctions précédentes pour différentes valeurs de p.

Que constate-t-on ?

Solution.

```
> seuil_gauche(1), seuil_droite(1), seuil_centre(1)
(47, 46, 3)
> seuil_gauche(2), seuil_droite(2), seuil_centre(2)
(460, 459, 10)
> seuil_gauche(3), seuil_droite(3), seuil_centre(3)
(4595, 4594, 30)
```

On constate que l'utilisation des rectangles au centre donne des résultats bien meilleurs que les rectangles à gauche ou à droite (qui sont sensiblement équivalents).

5. Adapter le code de `rectangles_gauche` pour écrire une fonction `rectangles_gauche_ab` prenant en arguments deux réels a et b et un entier naturel n et renvoyant les valeurs approchées de $\int_a^b f(x)\,\mathrm{d}x$ obtenues en utilisant n rectangles à gauche.

Solution.

```
def rectangles_gauche_ab(a,b,n):
  S = 0
  for k in range(n):
    S += (b-a)/n*f(a+k*(b-a)/n)
  return S
```

6. Faire de même avec `rectangles_droite` et `rectangles_centre`.

Solution.

```
def rectangles_droite_ab(a,b,n):
  S = 0
  for k in range(n):
    S += (b-a)/n*f(a+(k+1)*(b-a)/n)
  return S
```

```
def rectangles_centre_ab(a,b,n):
  S = 0
  for k in range(n):
    S += (b-a)/n*f((a+k*(b-a)/n+a+(k+1)*(b-a)/n)/2)
  return S
```

7. Calculer la valeur exacte de $J = \int_0^5 f(x)\,\mathrm{d}x$.

Solution.

$$\begin{aligned} J = \int_0^5 x^3 - 7x^2 + 14x - 7\,\mathrm{d}x &= \left[\frac{x^4}{4} - \frac{7}{3}x^3 + 7x^2 - 7x\right]_{0,5}^{4} \\ &= \frac{5^4}{4} - \frac{7}{3} \times 5^3 + 7 \times 5^2 - 7 \times 5 - \left(\frac{0^4}{4} - \frac{7}{3} \times 0^3 + 7 \times 0^2 - 7 \times 0\right) \end{aligned}$$

soit $\boxed{J = \dfrac{55}{12}}$.

8. Combien de rectangles faut-il au minimum, dans chacun des trois cas (rectangle à gauche, à droite et au centre) pour obtenir une valeur approchée de J à 10^{-3} près ?

Solution. On adapte les fonctions `seuil_gauche`, `seuil_droite` et `seuil_centre` en remplaçant respectivement `rectangles_gauche`, `rectangles_droite` et `rectangles_centre` par `rectangles_gauche_ab`, `rectangles_droite_ab` et `rectangles_centre_ab` :

```
def seuil_gauche_ab(a,b,p,I):
  n=1
  while abs(I-rectangles_gauche_ab(a,b,n)) > 10**(-p):
    n +=1
  return n

def seuil_droite_ab(a,b,p,I):
  n=1
  while abs(I-rectangles_droite_ab(a,b,n)) > 10**(-p):
    n +=1
  return n

def seuil_centre_ab(a,b,p,I):
  n=1
  while abs(I-rectangles_centre_ab(a,b,n)) > 10**(-p):
    n +=1
  return n
```

On appelle ces fonctions avec $a = 0$, $b = 5$, $p = 3$ et $I = 55/12$:

```
> seuil_gauche_ab(0,5,3,55/12)
50000
> seuil_droite_ab(0,5,3,55/12)
50001
> seuil_centre(0,5,3,55/12)
73
```

2) Un deuxième exemple

On considère ici la fonction $g : x \longmapsto x\mathrm{e}^{-x}$ définie sur $\mathbb{R}$ et $K = \displaystyle\int_{-1}^{3} g(x)\,\mathrm{d}x$.

1. Que suffit-il de modifier dans les fonctions précédentes pour obtenir des valeurs approchées de K à l'aide des trois types de rectangles ?

Solution. Il suffit de modifier la définition de la fonction `f`

```
from math import *

def f(x):
  return x*exp(-x)
```

2. Calculer la valeur exacte de K à l'aide d'une intégration par parties.

Solution. On pose $u : x \longmapsto x$ et $v : x \longmapsto -\mathrm{e}^{-x}$ qui sont de classe $\mathscr{C}^1$ sur $\mathbb{R}$ et telles que $u' : x \longmapsto 1$ et $v' : x \longmapsto \mathrm{e}^{-x}$ donc, en intégrant par parties :

$$\begin{aligned} K &= \left[x \times (-\mathrm{e}^{-x})\right]_{-1}^{3} - \int_{-1}^{3} 1 \times (-\mathrm{e}^{-x})\,\mathrm{d}x = -3\mathrm{e}^{-3} - (-1) \times (-\mathrm{e}^{-1}) + \left[-\mathrm{e}^{-x}\right]_{-1}^{3} \\ &= -3\mathrm{e}^{-3} - \mathrm{e}^{-1} - \mathrm{e}^{-3} - (-\mathrm{e}^{-1}) \end{aligned}$$

donc $\boxed{K = -4\mathrm{e}^{-3}}$.

3. Combien de rectangles faut-il au minimum, dans chacun des trois cas (rectangle à gauche, à droite et au centre) pour obtenir une valeur approchée de K à 10^{-3} près ?

Solution.

```
> seuil_gauche_ab(-1,3,3,-4*exp(-3))
5737
> seuil_droite_ab(-1,3,3,-4*exp(-3))
5734
> seuil_centre_ab(-1,3,3,-4*exp(-3))
61
```

3) Un troisième exemple

On considère ici la fonction $h : x \longmapsto \mathrm{e}^{-x^2}$ définie sur $\mathbb{R}$ et $H = \displaystyle\int_{0}^{3} h(x)\,\mathrm{d}x$.
En s'inspirant de ce qui précède, déterminer une valeur approchée de H à 10^{-3} près.

Solution. Ici, on ne connaît pas la valeur exacte de l'intégrale donc on ne peut pas utiliser les fonctions de seuil. On peut cependant utiliser la fonction `rectangle_centre_ab` et afficher les valeurs obtenues pour différentes valeurs de n et estimer une valeur approchée de H.

```
from math import *

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

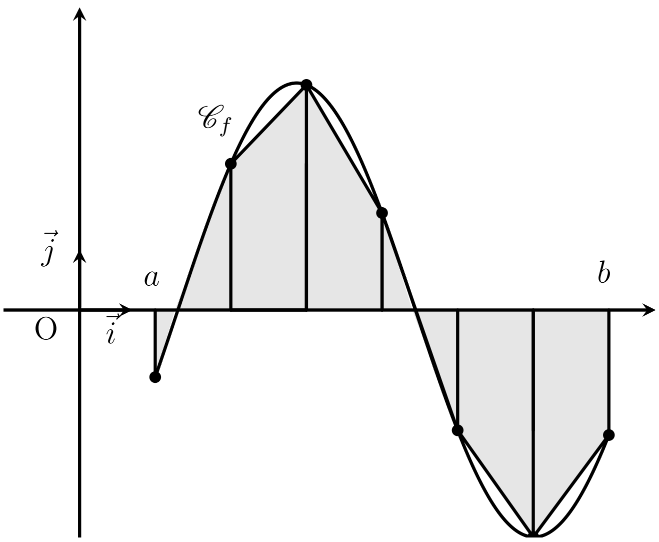
```
for n in range(1,11):
  print(rectangles_centre_ab(0,3,n))
```

```
0.5692694423100262
0.2966892076223455
0.4208978163665776
0.316197673685593
0.864168810237613
0.886130461769497
0.8862183766398241
0.8862155894632231
0.8862135935945482
0.8862121931006739
0.8862111933886581
0.8862104631033071
0.8862099170464343
```

On peut donc estimer que $H \approx 0{,}886$.

III. — Méthode des trapèzes

Dans cette seconde méthode, on approche, sur chaque intervalle $[x_k\,;x_{k+1}]$, la fonction f non plus par une fonction constante mais par la fonction affine prenant les mêmes valeurs que f aux bornes de l'intervalle. Cela revient alors à approcher l'intégrale de f entre x_k et x_{k+1} par l'aire algébrique d'un trapèze (celui-ci pouvant être croisé si la fonction change de signe sur l'intervalle).

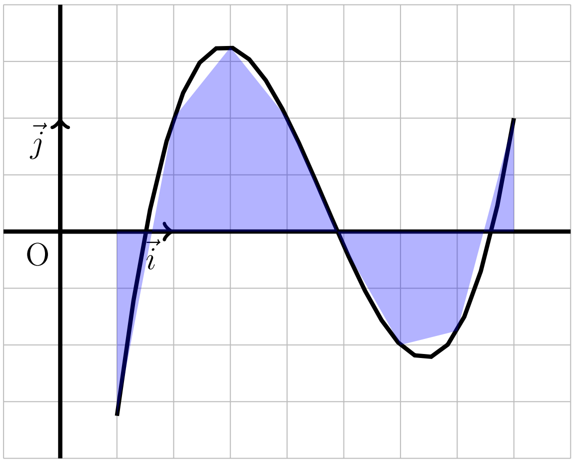


On admet que l'aire du trapèze considéré sur $[x_k\,;x_{k+1}]$, qu'il soit croisé ou non, est

$$\frac{b-a}{n} \times \frac{f(x_k)+f(x_{k+1})}{2}.$$

1. On considère à nouveau la fonction $f : x \longmapsto x^3 - 7x^2 + 14x - 7$ et $I = \displaystyle\int_{0,5}^{4} f(x)\,\mathrm{d}x$.

a. Sur le graphique ci-dessous, construire les trapèzes pour $n = 7$ et en déduire une valeur approchée de I.



On en déduit qu'une approximation de I est

$$0{,}5 \times \frac{f(0{,}5)+f(1)}{2} + 0{,}5 \times \frac{f(1)+f(1{,}5)}{2} + 0{,}5 \times \frac{f(1{,}5)+f(2)}{2} + 0{,}5 \times \frac{f(2)+f(2{,}5)}{2}$$
$$+0{,}5 \times \frac{f(2{,}5)+f(3)}{2} + 0{,}5 \times \frac{f(3)+f(3{,}5)}{2} + 0{,}5 \times \frac{f(3{,}5)+f(4)}{2} = 0{,}65625$$

b. Adapter la fonction **II.4.b.** pour écrire une fonction `trapezes` qui prend en argument un entier n et qui renvoie la valeur approchée de I obtenue en considérant n trapèzes.

Solution.

```
def trapezes(n):
  S = 0
  for k in range(n):
    S += 3.5/n*(f(0.5+k*3.5/n)+f(0.5+(k+1)*3.5/n))/2
  return S
```

c. Utiliser la fonction `trapezes` pour vérifier le résultat de la question **a.** puis pour obtenir des valeurs approchées de I pour $n = 10$ puis pour $n = 20$.

Solution.

```
> trapezes(7)
0.65625
> trapezes(10)
0.6748437499999997
> trapezes(20)
0.6882421875000018
```

d. Comparer ses valeurs avec `(rectangle_gauche(n)+rectangle_droite(n))/2` pour `n` valant 10 ou 20 et proposer une explication aux résultats observés.

Solution.

```
> (rectangles_gauche(10)+rectangles_droite(10))/2
0.6748437499999997
> (rectangles_gauche(20)+rectangles_droite(20))/2
0.688242187500002
```

On constate qu'on obtient les mêmes résultats. Ceci s'explique par le fait que, sur chaque intervalle $[x_k\,;x_{k+1}]$, pour les rectangles à gauche, on calcule $0{,}5 \times f(x_k)$, pour les rectangles à droite, on calcule $0{,}5 \times f(x_{k+1})$ donc, quand on exécute `(rectangle_gauche(n)+rectangle_droite(n))/2`, on calcule $\frac{0{,}5f(x_k)+0{,}5f(x_{k+1})}{2} = 0{,}5\frac{f(x_k)+f(x_{k+1})}{2}$, ce qui correspond exactement à l'aire du trapèze sur $[x_k\,;x_{k+1}]$.

e. Écrire une fonction `seuil_trapèze` prenant en argument un entier naturel p et renvoyant le nombre minimal de trapèzes nécessaires pour obtenir une valeur approchée de I à 10^{-p} près.

Solution.

```
def seuil_trapeze(p):
  n=1
  while abs(133/192-trapezes(n)) > 10**(-p):
    n +=1
  return n
```

f. Afficher les résultats obtenus en appelant la fonction `seuil_trapèze` pour différentes valeurs de p et comparer avec la méthode des rectangles.

Solution.

```
for p in range(1,6):
  print(seuil_trapeze(p))
```

```
5
14
43
134
423
```

On voit que la méthode des rectangles est bien plus efficace que celle des rectangles à gauche ou à droite mais sensiblement du même ordre que les rectangles au centre.

2. Adapter le code de la fonction `trapezes` pour écrire une fonction `trapezes_ab` prenant en arguments deux réels a et b et un entier naturel n et renvoyant la valeur approchée de $\int_a^b f(x)\,\mathrm{d}x$ obtenue à l'aide de n trapèzes.

Solution.

```
def trapezes_ab(a,b,n):
  S = 0
  for k in range(n):
    S += (b-a)/n*(f(a+k*(b-a)/n)+f(a+(k+1)*(b-a)/n))/2
  return S
```

3. Utiliser la fonction `trapezes_ab` pour calculer des valeurs approchées de K et H pour différentes valeurs de n et comparer la vitesse de convergence avec la méthode des rectangles.

Solution. Pour K :

n	rectangles à gauche	rectangles à droite	rectangles au centre	trapèzes
10	-0.846107634724169	0.3009495787008859	-0.1625768477351729	-0.27257902801164163
20	-0.5043422412296708	0.06918636548285642	-0.18994248615085957	-0.21757793787340737
30	-0.39852139149876664	-0.016168987023748086	-0.19505160398081015	-0.2073451892612575
50	-0.31680598321760356	-0.08739454053259238	-0.19767251116382906	-0.20210026187509836
100	-0.25723924719071606	-0.14253352584821063	-0.1987792314439256	-0.19988638651946358
1000	-0.20489094105177202	-0.19342036891752154	-0.19914458271630817	-0.1991556549846476

Pour H :

n	rectangles à gauche	rectangles à droite	rectangles au centre	trapèzes
10	1.0361835221667866	0.7362205451080127	0.8862099170464343	0.8862020336373999
20	0.9611967196066107	0.8112152310772237	0.8862080289438237	0.8862059753419171
30	0.9362005637900083	0.8362129047704169	0.886207654099489	0.8862067342802127
50	0.9162034242268731	0.8562108288151182	0.8862074589791809	0.8862071265209955
100	0.9012054416030277	0.9012054416030277	0.8862073760048715	0.8862072927500886
1000	0.887707162589475	0.8847075328188873	0.8862073485371923	0.8862073477041799

Là encore, on voit que les méthodes des rectangles au centre et des trapèzes sont équivalentes et beaucoup plus performantes que les méthodes des rectangles à droite ou à gauche. C'est d'autant plus flagrant dans le cas de H : les deux premières méthodes donnent 5 décimales exactes dès $n = 10$ alors que les deux dernières donnent seulement 2 décimales exactes pour $n = 1000$!