TP Scilab n°15 – calcul approché d'intégrale

1 Méthode des rectangles

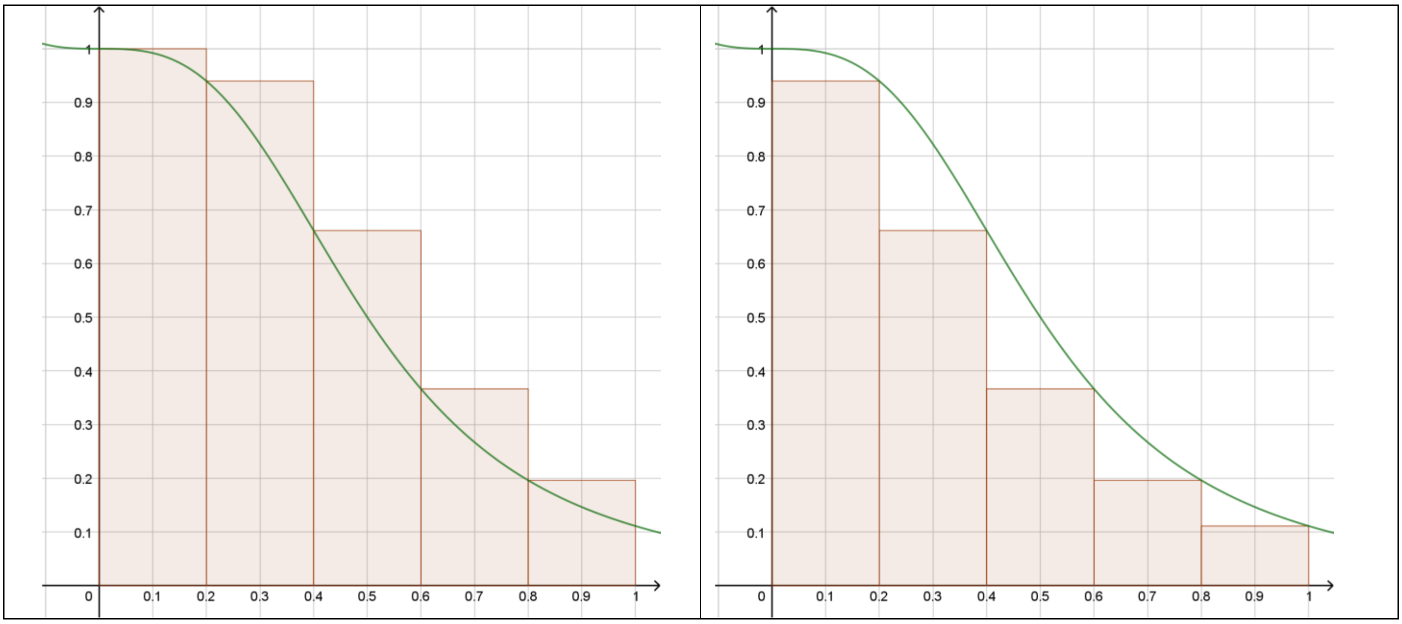
1.1 Rappels sur les intégrales de Riemann

Soit f une fonction continue sur $[0,1]$. On pose, pour tout $n \in \mathbb{N}^*$:

$$S_n = \frac{1}{n}\sum_{k=0}^{n-1} f\left(\frac{k}{n}\right) \quad \text{et} \quad T_n = \frac{1}{n}\sum_{k=1}^{n} f\left(\frac{k}{n}\right)$$

Question 1 :

On a tracé ci-dessous la courbe d'une fonction f définie sur $[0,1]$. Représenter sur le graphique de gauche les rectangles donc la somme des aires est égale S_5 et sur celui de droite les rectangles dont la somme des aires vaut T_5.



Question 2 :

Rappeler quelle est la limite de (S_n) et de (T_n) :

$$\lim_{n\to+\infty} \boldsymbol{S_n} = \lim_{n\to+\infty} \boldsymbol{T_n} = \int_0^1 \boldsymbol{f(t)\, dto}$$

1.2 Approximation de $\int_0^1 f(t)\,dt$ pour une fonction f particulière.

Dans cette partie, on considère que f est la fonction définie par :

$$f(x) = \frac{1}{1+8x^3}.$$

Question 3.a :

Réécrivez les définitions de S_n et de T_n en remplaçant $f\left(\frac{k}{n}\right)$ par son expression et en simplifiant au maximum. Indiquez ci-dessous l'expression obtenue.

$$S_n = \frac{1}{n}\sum_{k=0}^{n-1} \frac{1}{1+8\left(\frac{k}{n}\right)^3} = \frac{1}{n}\sum_{k=0}^{n-1} \frac{n^3}{n^3+8k^3} \quad \text{et} \quad T_n = \frac{1}{n}\sum_{k=1}^{n} \frac{n^3}{n^3+8k^3}$$

Question 3.b :

Ecrire un programme qui demande à l'utilisateur la valeur de n et affiche ensuite la valeur de S_n et de T_n. Vous devez trouver :

$$S_{100} \approx 0.549442839808 \quad \text{et} \quad T_{100} \approx 0.540553950919$$

Recopiez votre programme ci-dessous :

```
//Question 3.b
n=input("entrer la valeur de n : ")

S=0
for k=0:n-1
    S=S+n^3/(n^3+8*k^3)
end
S=1/n*S

T=0
for k=1:n
    T=T+n^3/(n^3+8*k^3)
end
T=1/n*T
disp("S vaut : ",S, "T vaut : ", T)
```

Question 3.c :

Mettre votre programme ci-dessus en commentaire et en faire deux fonctions : Sn(n) et Tn(n), qui renvoient respectivement la valeur de S_n et de T_n.

Ecrire ci-dessous le script de ces deux fonctions :

```
function S=Sn(n)
    S=0
    for k=0:n-1
        S=S+n^3/(n^3+8*k^3)
    end
    S=1/n*S
endfunction
```

```
function T=Tn(n)
    T=0
    for k=1:n
        T=T+n^3/(n^3+8*k^3)
    end
    T=1/n*T
endfunction
```

Question 3.d :

Ecrire un programme qui utilise les fonctions ci-dessous pour calculer S_n et T_n pour toutes les valeurs de n allant de 1 à 100 et qui stocke ces valeurs dans deux variables S et T puis qui affiche un graphique représentant ces deux suites.

Ecrire votre programme ci-dessous :

```
//Question 3.d

S=[]
T=S
for k=1:100
    S=[S Sn(k)]
    T=[T Tn(k)]
end
N=[1:100]
clf
plot2d(N,S,style=-2)
plot2d(N,T,style=-2)

//on peut aussi tracer les deux suites d'un coup,
//mais alors elles doivent être dans des matrices colonne :
clf
N=N' //On transpose N pour la mettre en colonne
S=S' //idem avec S
T=T' //idem avec T
plot2d(N, [S T],style=[-2 -2])
```

1.3 Approximation de $\int_0^1 f(t)\, dt$ pour une fonction f quelconque.

Passer tous les script des parties précédentes en commentaire.

Plutôt que d'avoir une fonction qui calcule la valeur de $\int_0^1 f(t)\, dt$ pour une fonction f particulière, on aimerait pouvoir calculer $\int_0^1 f(t)\, dt$ pour une fonction f quelconque que l'on définira par ailleurs.

Question 4.a

Compléter le script ci-dessous pour définir une fonction f1 correspondant la fonction $f : x \mapsto \frac{1}{1+8x^3}$ de la partie précédente.

```
//Question 4a

function y=f(x)
    y=1/(1+8*x^3)
endfunction
```

Question 4.b

Modifier les fonctions Sn(n) et Tn(n) créées à la question 3.c en des fonctions Sn(f,n) et Tn(f,n) pour que ces fonctions calculent les valeurs des deux suites S_n et T_n pour la valeur de n et la fonction f entrées en paramètre.
Ecrire ci-dessous le script de ces deux fonctions :

```
function S=Sn(f,n)
    S=0
    for k=0:n-1
        S=S+f(k/n)
    end
    S=1/n*S
endfunction
```

```
function T=Tn(f,n)
    T=0
    for k=1:n
        T=T+f(k/n)
    end
    T=1/n*T
endfunction
```

Question 4.c

4.c.i Calculer « à la main » l'intégrale suivante :

$$\int_0^1 \frac{4}{1+x^2}\,dx$$

$$\int_0^1 \frac{4}{1+x^2}\,dx = 4\int_0^1 \frac{1}{1+x^2}\,dx = 4\Big[\arctan x\Big]_0^1 = 4\left(\frac{\pi}{4}-0\right) = \pi$$

4.c.ii Définir dans Scilab la fonction $g : x \mapsto \frac{4}{1+x^2}$ puis calculer une valeur approchée de $\int_0^1 \frac{4}{1+x^2}dx$ à l'aide des fonctions Sn et Tn définies à la question 4.b. On prendra pour cela $n = 100\,000$.

```
//Question 4.c
function y=g(x)
    y=4/(1+x^2)
endfunction
```

Dans la console, on écrit ensuite :

```
--> Sn(g,100000)
 ans  =

   3.1416026535731527147277

--> Tn(g,100000)
 ans  =

   3.1415826535731525837036
```

1.4 Approximation de $\int_a^b f(t)\,dt$ pour une fonction f quelconque et un intervalle $[a,b]$ quelconque.

Question 5

Modifier les fonctions `Sn(f,n)` et `Tn(f,n)` créées à la question 4.b en des fonctions `Sn(f,a,b,n)` et `Tn(f,a,b,n)` pour que ces fonctions calculent les valeurs des deux suites de Riemann S_n et T_n pour la valeur de n et la fonction `f` entrées en paramètre telles qu'elles ont été définies dans le cours.

Ecrire ci-dessous le script de ces deux fonctions :

```
//Question 5
function S=Sn(f,a,b,n)
    S=0
    for k=0:n-1
        S=S+f(a+k*(b-a)/n)
    end
    S=(b-a)/n*S
endfunction

function T=Tn(f,a,b,n)
    T=0
    for k=1:n
        T=T+f(a+k*(b-a)/n)
    end
    T=(b-a)/n*T
endfunction
```

2 Méthode des trapèzes.

Plutôt que d'utiliser la méthode des rectangles, on peut utiliser la méthode des trapèzes qui est illustrée par l'animation Geogebra disponible ici.

Question 6 : Créer une fonction `trap(f,a,b,n)` qui calcule l'aire des n trapèzes. Ecrire ci-dessous le script de cette fonction.

Calculer alors la valeur approchée de $\int_0^1 \frac{4}{1+x^2}\,dx$ à l'aide de la méthode des trapèzes avec $n = 100\,000$ et comparer la précision obtenue par rapport à la méthode des rectangles.

```
function S=trap(f,a,b,n)
    S=0
    L=(b-a)/n //"largeur" des trapèzes
    for k=0:n-1
        // le k-ième trapèze a pour première base f(a+k*L)
        // et pour deuxième base f(a+(k+1)*L)
        // et pour "hauteur" (en fait pour largeur) L
        b1=f(a+k*L)
        b2=f(a+(k+1)*L)
        S=S+L*(b1+b2)/2
        //NB : les variables b1 et b2 servent uniquement
        // à rendre le programme lisible. On pourrait
        // très bien s'en passer.
    end
endfunction
```