

TP 10

Intégration numérique

Sources :

Proposition de corrigé

Activité 1 : Fractions rationnelles

Q 1 :

```
a=0
b=1
n=100
h=(b-a)/n
x=a
somme=f(a)
for k in range(1,n) :
    x=x+h
    somme=somme+f(x)
print(somme*h)
```

Q 2 :

```
def rect_gauche(f,a,b,n):
    h=(b-a)/n
    x=a
    somme=f(a)
    for k in range(1,n) :
        x=x+h
        somme=somme+f(x)
    return somme*h
```

Q 3 :

```
def rect_droit(f,a,b,n):
    h=(b-a)/n
    x=a+h
    somme=f(x)
    for k in range(1,n) :
        x=x+h
        somme=somme+f(x)
    return somme*h
```

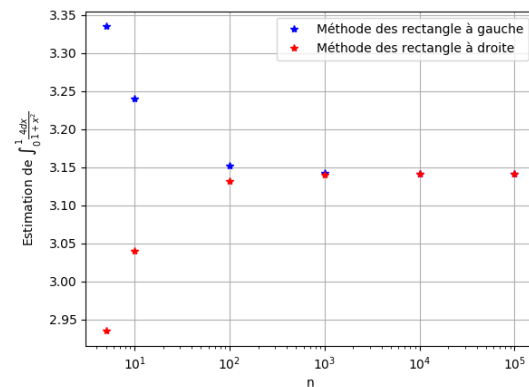
Q 4 :

```
def f(x):
    return 4/(1+x**2)
```

Q5:

```
liste_n=[5,10,100,1000,10000,int(1E5)]
yrd=[rect_droit(f,0,1,n) for n in liste_n]
yrg=[rect_gauche(f,0,1,n) for n in liste_n]
```

```
plt.clf()
plt.semilogx(liste_n,yrg, 'b*',label='Méthode des rectangle à gauche')
plt.semilogx(liste_n,yrd, 'r*',label='Méthode des rectangle à droite')
plt.legend()
plt.grid()
plt.xlabel('n')
plt.ylabel('Estimation de  $\int_0^1 \frac{dx}{1+x^2}$ ')
plt.savefig('tp10_q5_durif.png')
```



Q6:

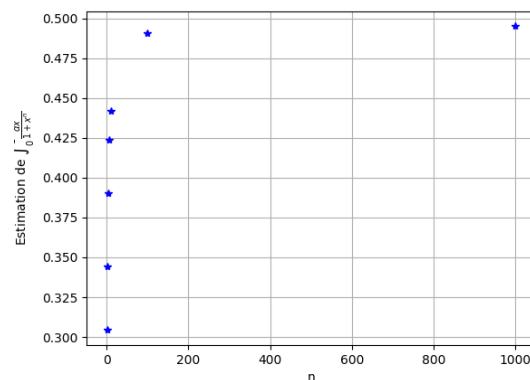
Q7:

```
def calcul_in(n):
    return rect_gauche(lambda x:x/(1+x**n),a,b,100)
```

Q8:

```
liste_n=[1,2,4,7,10,100,1000]
y=[calcul_in(n) for n in liste_n]
```

```
plt.clf()
plt.plot(liste_n,y, 'b*')
plt.xlabel('n')
plt.ylabel('Estimation de  $\int_0^{\alpha} \frac{dx}{1+x^n}$ ')
plt.grid()
plt.savefig('tp10_q8_durif.png')
```



Activité 2 : Mise hors gel des canalisations d'eau

Q9:

```
def trapeze(f,a,b,nb):
    res = 0
    pas = abs(b-a)/nb
    for i in range(nb):
        x0 = a+pas*i
        x1 = a+pas*(i+1)
        res = res + (x1-x0)*0.5*(f(x0)+f(x1))
    return res
```

Q10:

```
from math import pi,exp

def erf(x,nb):
    """
    x : borne supérieure de l'intégrale
```

```
nb : nombre d'échantillons
"""
if x>=0:
    return trapeze(fexp,0,x,nb)
else :
    return None
```

```
def fexp(u):
    return (2/math.sqrt(math.pi))*exp(-u*u)
```

Q 11:

```
# Paramètres physiques du problème
```

```
D = 2.8E-7      # Diffusivité thermique du sol terrestre
To = 278        # Température de la surface du sol terrestre pour t < 0
T1 = 258        # Température de la surface du sol terrestre pour t >= 0
Tf = 273.15     # Température de fusion de l'eau sous la pression de 1,013 bar
```

```
def Temperature (z, t) :
    u = z / (2 * sqrt (D * t))
    T = T1 + (To - T1) * erf (u,500)
    return T
```

Q 12:

```
ListeErreur = list ()

for i in range (0, 41, 1) :
    x = 0 + i * 0.05
    ListeErreur.append (erf (x,500))
```

Q 13:

```
z = 0
t = 864000      # 864 000 = 10 * 24 * 3600 secondes = 10 jours
T = Temperature (z, t)
while T < Tf :
    z = z + 0.01 # 0.01 m = 1 cm : précision de la recherche
    T = Temperature (z, t)

print ("La canalisation doit être enterrée à une profondeur minimale de", int (z * 100), "cm")
```