Intégration numérique – Méthode des petits et des grands rectangles
Bac S Polynésie juin 2013 – Exercice 1 questions 2.a. et 2.b.

E. Problème d'intégration numérique – Terminale S (fichier professeur Darboux0.py)

- Il s'agit de calculer une approximation numérique de l'intégrale $\int_0^1 (x+2)\mathrm{e}^{-x}dx$ par l'aire de 4 grands rectangles, puis de généraliser l'algorithme à N grands rectangles.
- Si l'exercice est rédigé en proposant le calcul approché de l'aire située entre la courbe représentative de la fonction f : x → (x + 2)e^{-x} définie sur l'intervalle [0 ; 1], et l'axe des abscisses, l'activité peut être menée à partir de la seconde.

E.1. Mise en œuvre en Python de l'algorithme proposé dans l'exercice.

- La mise en œuvre en python de l'algorithme de l'exercice 1 question 2.a. figure **lignes 9 à 25**. On vérifie la concordance des résultats trouvés à la main avec ceux du programme Python, pour 4 grands rectangles.

E.2. Modification de l'algorithme pour une approximation par N GRANDS rectangles

- La mise en œuvre en python de l'algorithme de l'exercice 1 question 2.b. figure **lignes 28 à 54**.
- Une version en lignes de commandes figure **lignes 41 à 46**.
- Une version sous forme de fonction Python ayant N en paramètre, figure **lignes 48 à 54**.

☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ **PROLONGEMENTS** ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺

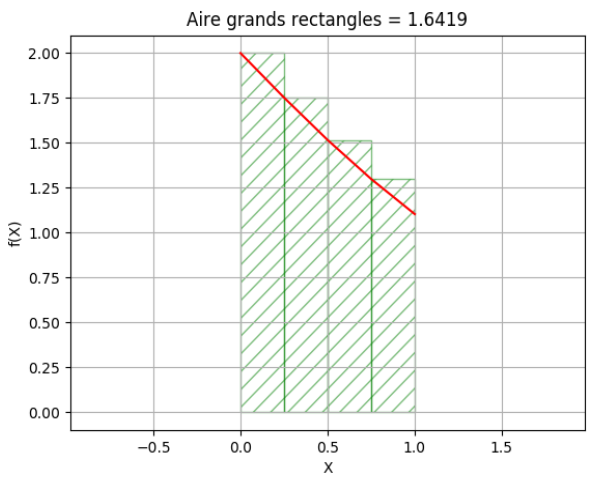
Les prolongements concernent d'une part les représentation graphiques des rectangles et d'autres parts, la généralisation, pour la même fonction, de la méthode des grands rectangles sur un intervalle [a ; b], puis en utilisant la méthode des petits rectangles, puis comparer, lorsque N augmente, les résultats des deux méthodes. La question de changer de fonction va se poser qui sera l'occasion de réfléchir à l'adéquation de l'algorithme dans le cas d'une fonction positive et strictement croissante, puis à la généralisation à un fonction continue intégrable dans le cadre du programme du lycée.

E.3. Prolongement 1 : Représentation graphique des 4 GRANDS rectangles

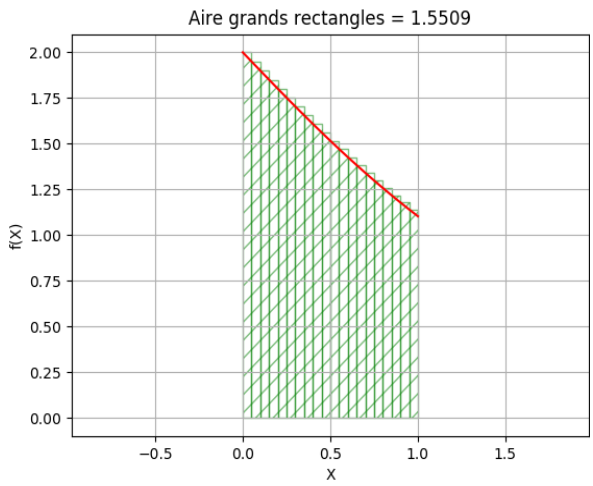
- L'illustration graphique des 4 grands rectangles comme dans le sujet figure **lignes 57 à 98.**
- Utilisation des tuples.

► Un exemple des résultats obtenus avec la fonction `aireGRN01(...)` :

`aireGRN01(N = 4)`



`aireGRN01(N = 20.)`



E.4. Prolongement 2 : Représentation graphique des N GRANDS rectangles

- Tout comme l'exercice propose d'étendre le calcul de l'approximation à N grands rectangles, il s'agit d'étendre la représentation graphique à N grands rectangles, en faisant afficher sur le graphique, l'aire calculée. La mise en œuvre en **lignes de commandes** figure lignes 101 à 144, une **fonction Python** correspondant figure **lignes 148 à 171**.

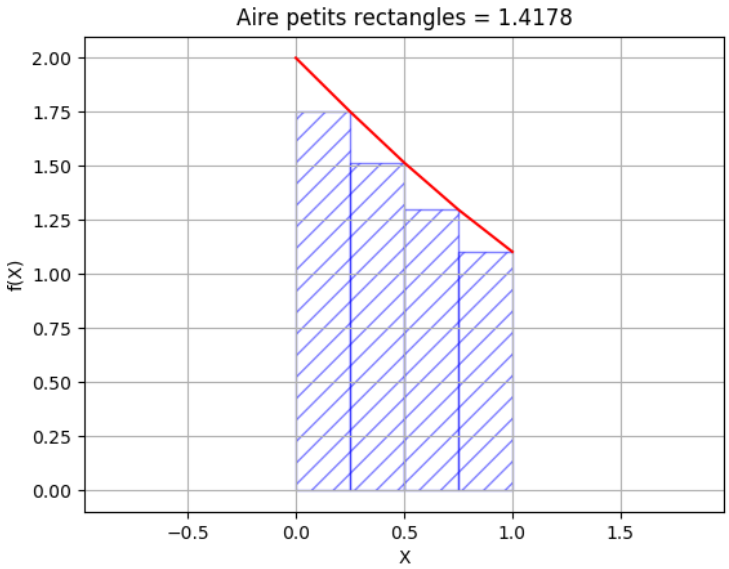
E.5. Prolongement 3 : Remplacer l'intervalle [0 ; 1] par un intervalle [a ; b]

- Avec la même fonction, peut-on appliquer la même méthode, avec N grands rectangles, sur un intervalle [a ; b] quelconque ? Comparaison tuple, liste.
- Un premier algorithme **(lignes 180 à 189)** effectue le calcul de l'aire des N grands rectangles.
- Un deuxième algorithme **(lignes 192 à 207)** affiche l'aire de chacun des N rectangles et l'aire totale.
- L'algorithme suivant **(lignes 210 à 239)** doit déterminer la longueur de la base et la hauteur de chacun des N grands rectangles puis en effectuer le tracé et calculer l'aire totale. La **fonction Python** correspondante figurent lignes **(lignes 242 à 267)**.

E.6. Prolongement 4 : Remplacer les grands rectangles par les PETITS rectangles.

- Le graphique ci contre illustre les petits rectangles. On travaille toujours avec la même fonction, avec N rectangles, sur un intervalle [a ; b].
- L'objectif et d'obtenir une fonction Python ayant N, a, b pour arguments, et produisant le graphique des rectangles avec l'affichage de l'aire totale.
  Cette **fonction Python** utilisera la fonction mathématique étudiée f qui aura été codée en fonction Python f(x).
- Les **fonctions Python** correspondantes figurent **lignes 270 à 298**.

Résultat de la fonction Python
`airePRNab(N = 4, a = 0, b = 1)`



E.7. Prolongement 5 : Juxtaposer horizontalement les graphiques des grands et des petits rectangles.

- Il s'agit de reprendre les graphiques obtenus prolongements 3 et 4 et de juxtaposer horizontalement les deux graphiques avec la fonction Python `subplot(…)` de la librairie matplotlib.pyplot.
- Les **lignes de commandes** correspondantes figurent **lignes 303 à 387,** la **fonction Python** correspondantes figure **lignes 390 à 430**. Les listes sont utilisées.

E.8. Prolongement 6 : Adapter la méthode et l'algorithme à une fonction positive strictement croissante sur un intervalle [a ; b].

- Comment adapter la méthode lorsque la fonction est strictement croissante ?
- La **fonction Python** figure **lignes 433 à 475**.

E.9. Prolongement 7 : Adapter la méthode à une fonction continue positive sur un intervalle [a ; b]

À venir...

Nom Prénom :

TP_ALGO_E.1. Calcul de l'aire des 4 GRANDS rectangles pour f sur [0 ; 1] (fichier élève Darboux0_Elev.py, lignes 10 à 25)

1° Dans Spyder, exécuter les **lignes de commandes** des **lignes 17 à 25** et compléter le tableau suivant en recopiant les résultats affichés dans la console.

```
"Exercice 1 Question 2.a. Sujet juin 2013 Polynésie série S - Aire totale des 4 grands rectangles "
"Traduction 'mot pour mot' de l'algorithme de l'énoncé"
"Soit la fonction f, positive et strictement décroissante sur [0 ; 1], définie sur R par"
from math import exp
def f(x):
    return((x + 2) * exp(-x))

"Calcul de la somme des aires
des 4 grands rectangles"
S = 0
for k in range(4):
    S = S + 1 / 4 * f(k / 4)
print("S = ", round(S, 4))
```

Pour visualiser les valeurs successives de S, exécuter d'abord la ligne S = 0 puis sélectionner la ligne S = S + 1 / 4 * f(k / 4) et l'exécuter 4 fois en notant la valeur de S dans la fenêtre "Explorateur de variables" de Spyder. Vérifier le résultat final en sélectionnant et en exécutant les 4 lignes.

2° Proposer ci dessous un algorithme permettant d'afficher successivement les aires des 4 rectangles.

Valeur initiale de S puis résultats successifs des 4 boucles : 0 ; ... ; ... ; ...; ...

3° Dans le repère ci-contre représenter la courbe de f et les 4 rectangles après avoir calculé les coordonnées de leurs sommets.

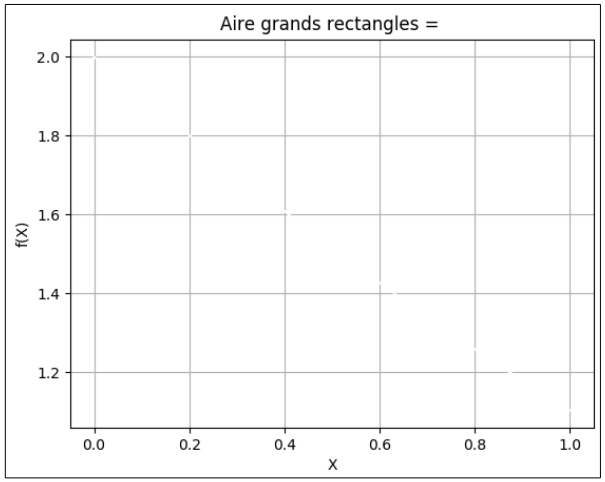
Compléter les coordonnées des sommets (dans le sens trigonométrique), calculer l'aire de chacun des rectangle et l'inscrire sur le graphique.

Rectangle 1, : A1(0;0), B1(0,25;0), C1(0,25; ....), D1(0; .....),

Rectangle 2, : A2(0,25;0), B2(0,5;0), C2(0,5; ....), D2(0,5; ....),

Rectangle 3, : A3(....;....), B3(....;....), C3(....;....), D3(....;....),

Rectangle 4, : A4(....;....), B4(....;....), C4(....;....), D4(....;....),



Nom Prénom :

TP_ALGO_E.2. Calcul de l'aire des N GRANDS rectangles pour f sur [0 ; 1] (fichier élève Darboux0_Elev.py, lignes 29 à 54)

1° Dans Spyder, exécuter les **lignes de commandes** des **lignes 37 à 46** et compléter le tableau suivant en recopiant les résultats affichés dans la console.

2° Écrire ci-dessous puis compléter les **lignes 51 à 54** pour obtenir une **fonction Python** calculant l'aire totale des N rectangles. L'exécuter et l'utiliser pour calculer l'aire de **20** puis **200** puis **2000** puis **20 000** rectangles. Donner les résultats avec 4 décimales.

```
" Exercice 1 Question 2.b. - Aire totale des N grands rectangles "
from math import exp
def f(x):
    return((x + 2) * exp(-x))

"Calcul de la somme des aires des N grands rectangles."
N = 20
S = 0
for k in range(N):
    S = S + 1 / N * f(k / N)
print("S = ", round(S, 4))

Résultat : Valeur de S :

"La même chose sous forme d'une fonction de paramètre N, c'est plus 'pratique'."
"Utilise la fonction f positive et strictement décroissante sur [0 ; 1]."

def aireGR(N = 4):

....

....

........

....

Résultats : Aire totale pour N = 20 : .......; N = 200 : .......; N = 2000 : .......; N = 20 000 : .......
```

Nom Prénom :

TP_ALGO_E.3. Prolongement 1 : Représentation graphique des 4 grands rectangles pour f sur [0 ; 1] (fichier élève Darboux0_Elev.py, lignes 57 à 98)

1° Dans Spyder, exécuter les **lignes de commandes** des **lignes 66 à 75** et compléter le tableau suivant en recopiant les résultats affichés dans la console ou dans la fenêtre de l'explorateur de variables.

2° Exécuter les **lignes de commande** des **lignes 84 à 98** noter l'aire totale des 4 grands rectangles puis copier coller le graphique dans un fichier texte.

```
"Pour tracer la portion de la courbe de f sur [0 ; 1].Je crée le tuple des abscisses et le tuple des"
"    ordonnées de quelques points. Entre ces point la commande plot(...) 'complétera' le tracé."
from math import exp
def f(x):
    return((x + 2) * exp(-x))

tupleX = (0, 1/4, 2/4, 3/4, 4/4)
tupleY = tuple((f(i) for i in tupleX))
tuple(round(tupleY[i], 2) for i in range(4))

Résultats : tupleX : .......... ; .......... ; .......... ; ..........

Résultats : tupleY : .......... ; .......... ; .......... ; ..........

"Je trace les 4 grands rectangles, en 'passant' par les 4 sommets de chacun des 4 rectangles :"
import matplotlib.pyplot as plt
airetotGran = 0
plt.figure()
plt.plot(tupleX, tupleY, 'r-')
for i in range(4):
    plt.fill((tupleX[i], tupleX[i], tupleX[i + 1], tupleX[i + 1]),
             (0, tupleY[i], tupleY[i], 0),
             alpha = 0.5, fill = False, hatch = '//', color = 'green')
    airetotGran = airetotGran + (1/4 * tupleY[i])
plt.title('Aire grands rectangles = ' + str(round(airetotGran, 4)))
plt.xlabel('X')
plt.ylabel('f(X)')
plt.grid() ; plt.axis('equal') ; plt.show()

Résultats : Aire totale des 4 grands rectangles =
```

Nom Prénom :

TP_ALGO_E.4. Prolongement 2: Représentation graphique des N g. rectangles pour f sur [0 ; 1] (fichier élève Darboux0_Elev.py, lignes 101 à 171)

1° Dans Spyder, exécuter **les lignes de commandes** des **lignes 110 à 144** et compléter le tableau suivant en recopiant les résultats affichés dans la console.

2° Dans le tableau ci-dessous, compléter les **11 lignes manquantes** pour obtenir une **fonction Python** réalisant le graphique des N rectangles et calculant leur aire totale. Les saisir **lignes 156 à 166** dans l'éditeur puis exécuter et utiliser la fonction pour obtenir les graphiques et les aires pour N = 10 puis 20 puis 100 rectangles. Noter les aires ci-dessous et copier les graphiques dans un fichier texte.

```
"-- PROLONGEMENT 2 - Représentation graphique de N grands rectangles"
from math import exp
def f(x):
    return((x + 2) * exp(-x))

def aireGRN01(N = 4):
    import matplotlib.pyplot as plt
....
....
....
....
....
....
........
........
........
........
....
    plt.xlabel('X')
    plt.ylabel('f(X)')
    plt.axis('equal')
    plt.grid() ; plt.show()
```

Résultats du 1° : Aire totale pour N = 5 : .....; N = 10 : .....; N = 30 : .....; N = 100 : .....

Résultats du 2° : Aire totale pour N = 6 : .....; N = 12 : .....; N = 32 : .....; N = 128 : .....

Nom Prénom :

TP_ALGO_E.5. Prolongement 3 : Calcul de l'aire totale des N GRANDS rectangles pour f sur [a ; b], détailler l'aire de chacun des N rectangles (fichier élève Darboux0_Elev.py, lignes 174 à 209) fiche 1 / 3 , les lignes de commande.

1° Dans Spyder, exécuter **les lignes de commandes** des **lignes 180 à 189** et compléter le tableau suivant en recopiant les résultats affichés dans la console ou dans la fenêtre de l'explorateur de variables.

2° Modifier l'algorithme du **1°** pour qu'il calcule et affiche l'aire de chacun des N grands rectangles et l'aire totale. On limitera cet affichage pour N ≤ 30.

3° Saisir cet algorithme **lignes 200 à 208** et l'exécuter pour tester ses résutats. On peut utiliser des listes et la commande `liste.append(x)`.

```
1° "-- PROLONGEMENT 3 - Calcul et affichage de l'aire totale des N GRANDS RECTANGLES"
from math import exp
def f(x):
    return((x + 2) * exp(-x))

N = 5 ; a = 1 ; b = 3
pas = (b - a) / N
S = 0
for k in range(N):
    S = S + pas * f(a + k * pas)
print("S = ", round(S, 4))
Résultats : Aire totale S pour N = 5 : .......; N = 10 : .......; N = 30 : .......; N = 100 : .......

2° "Calcul et affichage de l'aire de chacun des N GRANDS RECTANGLES et de l'aire totale"









Résultats : Aire des 5 rectangles dans [1 ; 3] A1 = .... ; A2 = .... ; A3 = .... ; A4 = .... ; A5 = ....
```

Nom Prénom :

TP_ALGO_E.5. Prolongement 3 : Calcul de l'aire totale et représentation graphique des N GRANDS rectangles pour f sur [a ; b] (fichier élève Darboux0_Elev.py, lignes 210 à 239) fiche 2 / 3, les lignes de commandes.

1° Dans Spyder exécuter les **lignes 212 à 224** et copier les résultats dans le tableau ci-dessous. Vérifier l'algorithme avec N = 4, a = 0, b = 1.

2° Exécuter les **lignes 226 à 239** et copier coller le graphique dans un fichier texte (libre office writer par exemple).

```
from math import exp
import matplotlib.pyplot as plt

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

N = 5 ; a = 0 ; b = 2
pas = (b - a) / N
pas
Résultat : pas =
tupleX = tuple(a + pas * i for i in range(N + 1))
tupleX
Résultats : tupleX : .......... ; .......... ; .......... ; .......... ; ..........
tupleY = tuple(f(i) for i in tupleX)
tupleY
Résultats : tupleY : .......... ; .......... ; .......... ; .......... ; ..........
airetotGran = 0
plt.figure()
plt.plot(tupleX, tupleY, 'r-')
for i in range(N):
    plt.fill((tupleX[i], tupleX[i], tupleX[i + 1], tupleX[i + 1]),
             (0, tupleY[i], tupleY[i], 0),
             alpha = 0.5, fill = False, hatch = '//', color = 'green')
    airetotGran = airetotGran + (pas * tupleY[i])
plt.title('Aire grands rectangles = ' + str(round(airetotGran, 4)))
plt.xlabel('X') ; plt.ylabel('f(X)') ; plt.grid() ; plt.axis('equal')
plt.show()
```

Nom Prénom :

TP_ALGO_E.5. Prolongement 3 : Calcul de l'aire totale et représentation graphique des N GRANDS rectangles pour f sur [a ; b] (fichier élève Darboux0_Elev.py, lignes 242 à 267) fiche 3 / 3, une fonction Python.

1° En utilisant les lignes de commandes du tableau précédant, rédiger une **fonction Python** faisant la représentation graphique des N grands rectangles et calculant leur surface totale.

2° Saisir les **lignes 251 à 262** puis exécuter la fonction et l'utiliser sur une exemple connu pour vérifier les résultats. Copier coller les graphiques obtenus dans un fichier texte (libre office writer par exemple).

```
def aireGRNab(N = 5, a = 0, b = 2):
    import matplotlib.pyplot as plt
....
....
....
....
....
....
....
........
........
........
........
....
    plt.xlabel('X')
    plt.ylabel('f(X)')
    plt.grid() ; plt.axis('equal') ; plt.show()
```

Nom Prénom :

TP_ALGO_E.6. Prolongement 4 : Calcul de l'aire totale et représentation graphique des N PETITS rectangles pour f sur [a ; b] (fichier élève Darboux0_Elev.py, lignes 270 à 298), une fonction Python.

1° En s'inspirant des algorithmes précédants, rédiger une **fonction Python** faisant la représentation graphique des N PETITS rectangles et calculant leur surface totale. On peut proposer une modification des **lignes 289 à 292** permettant de réaliser le graphique et le calcul demandés.

2° Saisir les **lignes 289 à 292** puis exécuter et utiliser la fonction. Copier coller les graphiques obtenus dans un fichier texte (libre office writer par exemple).

```
def airePRNab(N = 4, a = 0, b = 1):
    import matplotlib.pyplot as plt
    pas = (b - a) / N
    tupleX = tuple(a + pas * i for i in range(N + 1))
    tupleY = tuple(f(i) for i in tupleX)
    airetotPetit = 0
    plt.figure()
    plt.plot(tupleX, tupleY, 'r-')
    for i in range(N):

........


........


........


........


........

    plt.title('Aire petits rectangles = ' + str(round(airetotPetit, 4)))
    plt.xlabel('X')
    plt.ylabel('f(X)')
    plt.grid() ; plt.axis('equal') ; plt.show()
```