TP	TP 2.1 (1)	TSI1
	Intégration numérique	Période 2
		1h

Ouvrir le fichier **TP_2.1.py** dont certaines fonctions vues en td ont été saisies.

1) Définir une fonction **cosp()** admettant en entrée une variable **x** et qui renvoie en sortie :
0 sur l'intervalle $\frac{\pi}{2} \leq x \ mod \ 2\pi \leq \frac{3\pi}{2}$ (**mod** s'écrit **%** en python)
cos(x) lorsque $0 < x \ mod \ 2\pi < \frac{\pi}{2}$ et $-\frac{\pi}{2} < x \ mod \ 2\pi \leq 0$.

2) Proposer une instruction à écrire dans le programme principal qui permet de disposer de la fonction **cos()** à partir de la bibliothèque **math**.

3) Ecrire l'instruction qui permet de stocker, dans la variable **int_cosp**, l'intégrale numérique obtenue par la fonction **rectangleg()** avec 4 partitions sur l'intervalle [0,2π] pour la fonction **cosp** (remarque : analytiquement on obtient 2). Afficher le résultat.

4) Tracer le tableau d'évolution des variables **k**, **f(a+k*h)** et **s** dans le cas de l'instruction précédente.

h =				
k	0	1	2	3
f(a+k*h)				
s				

5) Comparer l'erreur faite à la question précédente et celle annoncée par l'expression
$E = \max_{a \leq x \leq b}\left(\left|f^{'}(x)\right|\right) * \frac{(b-a)^2}{2N}$.

Comparatif des méthodes

6) Ecrire une fonction **trapeze()** qui permet d'obtenir l'intégrale de **f** entre **a** et **b** pour **N** partitions.

7) Quel est l'intérêt et l'inconvénient de la fonction **trapeze()** par rapport à la fonction **rectangleg()** ? Corriger éventuellement votre fonction **trapeze()** pour qu'elle ne nécessite que **N** appels à la fonction **f**.

8) Créer une fonction **tracer()** qui admet comme argument **f**, **a**, **b** et **N** et qui trace les courbes intégrales de la fonction **f** par les trois méthodes d'intégration (on pourra définir un label pour chacune des courbes afin de les distinguer ; voir annexe suivante). Observer l'effet de **N** sur l'évolution de la précision de l'intégration et le rapprochement des courbes.

ANNEXE Tracés avec la bibliothèque **matplotlib**

pyplot de **matplotlib** étant importé par l'instruction : **import matplotlib.pyplot as plt**

Tracé de la courbe **y(x)** en rouge (*)avec **'lab'** comme label	**plt.plot(x,y,'r',label='lab')**
Afficher la légende (labels) des courbes	**plt.legend()**
Afficher et fermer la figure en cours	**plt.show()**

(*) Noms raccourcis des couleurs : rouge(**'r'**), bleu (**'b'**), magenta ('**m'**), vert (**'g'** ; green), jaune (**'y'** ; yellow), noir (**'k'**; black).

Pseudo code de la fonction tracer(f: function, a: float, b:float, N: int)→float

```
h ← (b-a)/N
créer les listes vides suivantes : X, Yg, Yd, Ym et Yt

Pour i variant de 0 à N inclus répéter
    ajouter a+i*h à X
    ajouter à Yg l'intégrale par la méthode des rectangles à gauche pour f entre a et a+i*h
    ajouter à Yd l'intégrale par la méthode des rectangles à droite pour f entre a et a+i*h
    ajouter à Ym l'intégrale par la méthode des rectangles au milieu pour f entre a et a+i*h
    ajouter à Yt l'intégrale par la méthode des trapèzes pour f entre a et a+i*h
Fin Pour

tracer la courbe Yg(X) en rouge avec le label 'rectg'
tracer la courbe Yd(X) en bleu avec le label 'rectd'
tracer la courbe Ym(X) en magenta avec le label 'rectm'
tracer la courbe Yt(X) en rouge avec le label 'trap
afficher la légende des courbes
afficher et fermer la figure en cours
```

Fin de la fonction tracer

9) En utilisant la fonction **tracer()**, montrer que l'intégrale de **cos(x)** est bien **sin(x)** (on tracera d'abord la courbe **sin(x)** avec le label **'sin'**).

10) En utilisant la fonction **tracer()**, montrer que l'intégrale de **sin(x)** est **c–cos(x)** à une constante **c** près que l'on définira.