

Si l'espace prévue pour une réponse ne suffit pas, veuillez continuer au verso ou annexer une feuille supplémentaire.

Nom & prénom :

Classe : Atelier : Traitement & Transmission de Signal

Enseignant : A. Mhamdi



Ne rien écrire dans ce tableau.

Question	1	2	3	4	5	6	Total
Barème	1	1	2	2	2	2	10
Note							

On se propose d'étudier la décomposition en série de **Fourier** d'un signal carré, qu'on dénote par x .

1. (1 point) Expliquez l'intérêt du code de la cellule suivante :

```
[1]: import numpy as np
      from scipy import signal
      import matplotlib.pyplot as plt

      plt.style.use("ggplot")
```

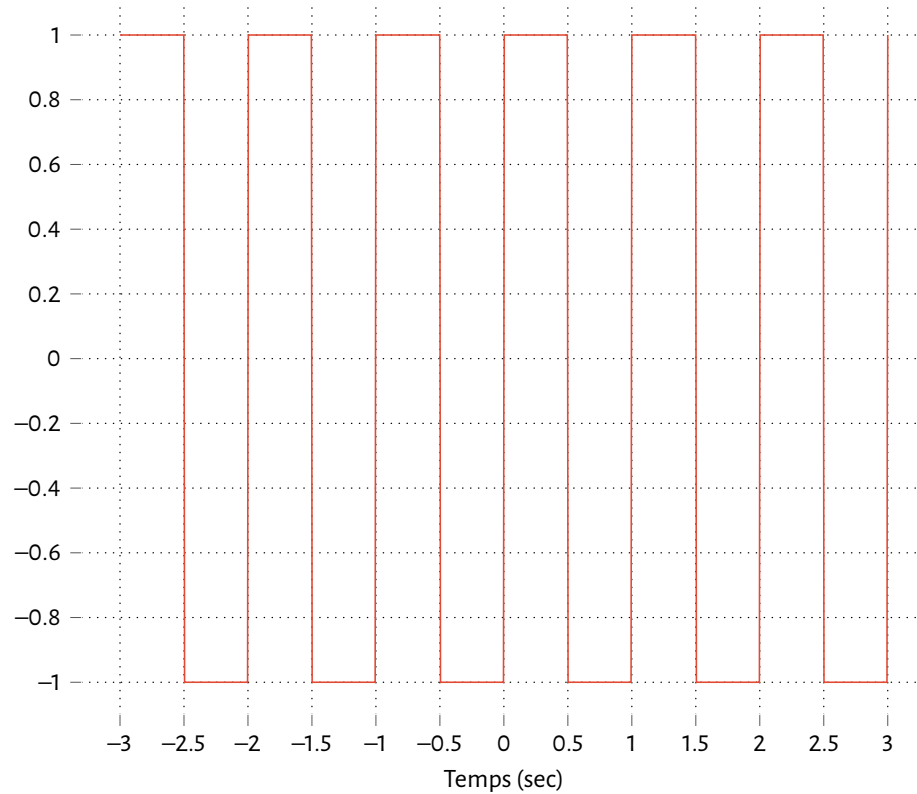
.....
.....
.....

2. (1 point) Générez un vecteur temps t de 1000 points, allant de -3 à 3 .

```
[2]: nt = 1000
      t = np.linspace(-3,3,nt)
      wt = 2 * np.pi * t/1 #  $F = 1\text{Hz}$ , i.e.  $T = 1\text{ sec}$ 
```

3. (2 points) Écrivez le code qui permet de générer le signal de la figure ci-dessous.

```
[3]: x = signal.square(wt, 0.5) # Amplitude  $A = 1$ 
      plt.plot(t,x)
      plt.xlabel('Temps (sec)')
      plt.grid()
```



4. (2 points) Le signal x peut être approximé par la fonction suivante :

$$x(t) \approx \sum_{k=1}^{\infty} b_k \sin(k\omega t) \quad b_k = \frac{4}{k\pi}$$

Expliquez l'origine de cette approximation.

.....

.....

.....

5. (2 points) Implémentez cette approximation pour $k = 10$.

```
[4]: xapp_lst = [ 4/(k*np.pi) * np.sin(k * wt) for k in range(1,20,2) ]
      print(type(xapp_lst))
      xapp_np = np.asarray(xapp_lst) # Conversion list --> numpy
      print(type(xapp_np))
      print(xapp_np.shape)
      xapp = xapp_np.sum(axis=0) # Somme sur les colonnes
```

6. (2 points) Tracez sur une même figure les deux fonctions : originale et approximée.

```
[4]: plt.plot(t, xapp)
      plt.xlabel('Temps (sec)')
```

```
plt.grid()
```

