TP : Filtrage de signaux périodiques

Commencer par importer les modules `numpy`, `matplotlib` et `math`

```
import numpy as np
import matplotlib.pyplot as plt
from math import *
```

Dans ce TP, on veut reproduire l'effet de différents filtres sur un signal créneau, puis sur un signal triangle.

I Construction du signal d'entrée

On considère un signal d'entrée $e(t)$, périodique, de fréquence f_e (par exemple $f_e = 1$ kHz). Sa pulsation est $\omega_e = 2\pi f_e$.
On va poser ces grandeurs comme variables globales :

```
fe = 1000 # fréquence fondamentale du signal d'entrée, en hertz
we = 2*pi*fe
Te = 1/fe
```

La décomposition de Fourier de ce signal est $\boxed{e(t) = c_0 + \sum_{n=1}^{+\infty} c_n \cos(n\omega_e t + \varphi_n)}$

En python, on se donne la liste des c_n et des φ_n pour le signal d'entrée, par exemple pour le signal $e(t) = 0,5 + \cos(\omega_e t + \pi)$

```
c_entree=[0.5,1]
phi_entree=[0,pi] #on donne la valeur 0 au 1er élément de la liste,
#mais il ne sert pas puisque le terme constant n'a pas de phase
```

1. Écrire une fonction `signal(c,phi,t)` qui prend en argument la liste `c` des c_n, la liste `phi` des φ_n et la liste des temps `t` et qui renvoie la valeur prise par le signal à chaque instant de la liste `t`.
   On pourra tester cette fonction pour faire apparaître le signal $e(t)$ précédent avec :

   ```
   t=np.linspace(0,5*Te,500)
   ```

2. On veut maintenant créer les liste des c_n et des φ_n pour un signal créneau. On donne la décomposition en série de Fourier d'un signal créneau prenant 0 et A comme valeurs :

$$e_{carre}(t) = \frac{A}{2} + \sum_{k=1}^{+\infty} c_k \sin(k\omega_e t)$$

avec, en prenant l'origine des temps sur un front montant du créneau :

$$c_k = \begin{cases} 0 & \text{si } k \text{ pair} \\ \dfrac{2A}{k\pi} & \text{si } k \text{ impair} \end{cases}$$

Écrire une fonction `carre(A,n)` qui renvoie les listes contenant les n premiers éléments des c_k et des φ_k pour un signal créneau prenant 0 et A comme valeurs.
Vérifier que votre fonction donne le résultat attendu en utilisant la fonction `signal`. On pourra également remarquer que plus n est grand, plus le signal est proche d'un signal créneau.

II Signal de sortie du filtre

On rappelle l'action d'un filtre sur un signal d'entrée décomposé en série de Fourier :

$$e(t) = c_0 + \sum_{n=1}^{+\infty} c_n \cos(n\omega_e t + \varphi_n) \quad \overset{\text{sys}}{\rightarrow} \quad s(t) = c'_0 + \sum_{n=1}^{+\infty} c'_n \cos(n\omega_e t + \varphi'_n)$$

$$\text{avec } c'_n = c_n \times |\underline{H}(n\omega_e)|, \text{ et } \varphi'_n = \varphi_n + \arg(\underline{H}(n\omega_e))$$

Commençons par un filtre passe-bas. On rappelle sa fonction de transfert, son gain et son argument :

$$\underline{H}(\omega) = \frac{1}{1 + j\dfrac{\omega}{\omega_0}} \quad G = |\underline{H}(\omega)| = \frac{1}{\sqrt{1 + \omega^2/\omega_0^2}} \quad \arg(\underline{H}(\omega)) = -\arctan\frac{\omega}{\omega_0}$$

On déclare en variable globale la pulsation de coupure :

```
f0 = 1e3
w0 = 2*pi*f0
```

3. Écrire les fonctions `G(w)` et `argH(w)` qui renvoie le gain et la phase pour une pulsation `w` donnée.
4. Écrire la fonction `sortie(c_e,p_e)` qui renvoie sous forme de listes les c'_k et φ'_k pour une entrée dont la décomposition de Fourier est décrite par `c_e` et `p_e`.

5. Tracer la sortie (et l'entrée sur le même graphe) et vérifier que le filtre passe-bas agit comme prévu par le cours de physique pour différentes pulsations du signal d'entrée et différentes pulsations de coupure.

Reprendre les questions précédentes pour un passe-bande :

$$\underline{H}(\omega) = \frac{1}{1 + jQ\left(\dfrac{\omega}{\omega_0} - \dfrac{\omega_0}{\omega}\right)} \quad G = |\underline{H}(\omega)| = \frac{1}{\sqrt{1 + Q^2\left(\dfrac{\omega}{\omega_0} - \dfrac{\omega_0}{\omega}\right)^2}}$$

$$\arg(\underline{H}(\omega)) = -\arctan\left(Q\left(\frac{\omega}{\omega_0} - \frac{\omega_0}{\omega}\right)\right)$$

III Filtrage numérique

Si on n'a pas la décomposition en série de Fourier du signal d'entrée, on ne peut appliquer la méthode précédente.
Imaginons qu'on a acquis un signal $e(t)$ directement pendant une durée T en prenant une valeur tous les τ (on parle de pas d'échantillonage). On a donc uniquement accès à la liste `E` des valeurs prises par le signal à différents instants (c'est-à-dire que `E[k]`$= e(t_k)$ avec $t_k = k\tau$).

Pour déterminer le signal de sortie d'un fitre passe-bas par exemple, on va devoir repasser la fonction de transfert dans le domaine temporel afin de résoudre l'équation :

$$\frac{ds}{dt} = -\omega_0 s(t) + \omega_0 e(t)$$

6. Montrer que la méthode d'Euler pour la résolution de cette équation différentielle consiste à appliquer la relation de récurrence :

$$s_{k+1} = (1 - \omega_0\tau)s_k + \omega_0\tau e_k$$

   On notera $e(t_k) = e_k$ et $s(t_k) = s_k$

7. Implémenter la fonction `euler(E,tau,T)` qui prend en entrée la liste `E` contenant les valeurs du signal d'entrée, `tau` le pas d'échantillonage et `T` la durée totale de l'acquisition.
   On prendra par défaut $s_0 = 0$

8. Tracer la sortie pour l'entrée ci-dessous dont on fera varier le τ. Commenter.

```python
tau=1/(fe*100) #on prend 100 points par période
T=5/fe #on acquiert sur 5 périodes du signal
t2=np.linspace(0,T,int(T/tau))
E=signal(c_carre,phi_carre,t2)
```