Codes pour la conversion analogique - numérique et pour le filtrage numérique

```
import math as m
import matplotlib.pyplot as plt


# fonction pour obtenir un graphe
def graphe(func,a,b) :
    N = 1000 # nombre de points
    lx = [a+i*(b-a)/N for i in range(N+1)]
    ly = [func(x) for x in lx]
    plt.plot(lx,ly)
    plt.show()


# genere une fonction sinusoidale
def imp_cos(freq,amp,phase,offset) :
    def out(t) :
        return offset+amp*m.cos(2*m.pi*freq*t+phase)
    return out


# genere une fonction triangulaire
def trig(freq,amp,offset) :
    def out(t) :
        T = 1/freq
        t=t%T
        if t<=T/2 :
            return offset-amp+t*4*amp/T
        else :
            return offset+amp-(t-T/2)*4*amp/T
    return out


# conversion analogique - numerique
def can(func,fech,delta_t,n_bits) :
    l_float = []
    N = 2**n_bits
    v_min,v_max = func(0),func(0)
    for i in range(int(fech*delta_t)+1):
        v = func(i/fech)
        l_float.append(v)
        if v<v_min :
            v_min=v
        if v>v_max :
            v_max=v
    vm = max(abs(v_min),v_max)
    pas = 2*vm/(N-1)
    l_quant = [round(v/pas)*pas for v in l_float]
    return l_float,l_quant
```

```
# conversion numerique - analogique
def cna(l_quant,fech,delta_t) :
    def out(t) :
        i = round(t*fech)
        return l_quant[i]
    return out


# filtrage numerique passe bas
def passe_bas(l_quant,fech,fc):
    n = len(l_quant)
    out = [0 for _ in range(n)]
    out[0] = 0
    for i in range(n-1):
        out[i+1]=out[i]*(1-2*m.pi*fc/fech)+l_quant[i]*2*m.pi*fc/fech
    return out


# filtrage numerique passe haut
def passe_haut(l_quant,fech,fc):
    n = len(l_quant)
    out = [0 for _ in range(n)]
    out[0] = 0
    for i in range(n-1):
        out[i+1]=out[i]*(1-2*m.pi*fc/fech)+l_quant[i+1]-l_quant[i]
    return out
```