Devoir d'informatique numéro 3

Exercice 1 (Sujet 0 banque PT)

Les flottants simple précision occupent $32/8 = 4$ octets. Donc il faut prévoir 800 octets en RAM, i.e. choisir le modèle avec

1024 octet de RAM

Exercice 2

1) `real` contient la partie réelle et `imag` la partie imaginaire.

```
1  import numpy as np
2
3  def trans_fourier_freq(L,k) :
4      N = len(L)
5      real = 0
6      imag = 0
7      for n in range(N):
8          real = real+L[n]*np.cos(-2*np.pi*k*n/N)
9          imag = imag+L[n]*np.sin(-2*np.pi*k*n/N)
10     return real,imag
```

2) On suppose que NumPy reste chargé, bref que tout est dans un seul fichier.

```
1  def module(a,b):
2      return np.sqrt(a**2 + b**2)
```

3) Le module de la transformée de Fourier discrète :

```
1  def trans_fourier(L):
2      N = len(L)
3      TFL = np.zeros(N)
4      for k in range(N):
5          ak,bk = trans_fourier_freq(L,k)
6          TFL[k] = module(ak,bk)
7      return TFL
```

4) La fonction `trans_fourier_freq(L,k)` fait N appels à sinus, et N appels à cosinus (boucle `for`). Donc $2N$ appels en tout. Dans lors de la boucle `for` de `trans_fourier(L)`, on fait N appels à `trans_fourier_freq(L,k)`.

Au total il y a donc $2N^2$ appels à cosinus et sinus : la complexité est en $O(N^2)$.

Ce n'est pas une complexité rédhibitoire (comme le serait une complexité exponentielle), mais ce n'est pas non plus extrêmement efficace (comme une complexité linéaire).

5)

6) Pour $N = 11$, $[0, 1, 2, 3, 4, 5, -5, -4, -3, -2, -1]$

Pour $N = 10$, $[0, \frac{1}{2}, 1, \frac{3}{2}, 2, -\frac{5}{2}, -2, -\frac{3}{2}, -1, -\frac{1}{2}]$

7) On transcrit la description en python. `w` contient le résultat.

```python
def freq_fourier(U,Te):
    N = len(U)
    w = np.zeros(N)
    if ( N%2 == 1 ):
        for i in range(N//2+1):
            w[i] = i/(Te*N)
        for i in range(N//2+1,N):
            w[i] = (i-N)/(Te*N)
    else:
        for i in range(N//2):
            w[i] = i/(Te*N)
        for i in range(N//2+1,N):
            w[i] = (i-N)/(Te*N)
    return w
```

8) Pour comprendre de quoi il retourne, vous pouvez aller lire l'ensemble d sujet 0 sur le site de la banque PT : `http://www.banquept.fr`.

La boucle `for` est une recherche de maximum dans une liste : du cours.

```python
def freq_corde(U,Te):
    TF = trans_fourier(U)
    freq = freq_fourier(U,Te)
    maximum = 0
    indice_max = 0
    for i in range(len(U)):
        if (TF[i] > maximum) and (freq[i] > 0):
            maximum = TF[i]
            indice_max = i
    return freq[indice_max]
```

Exercice 3

Il faut commencer par stocker la valeur de `liste_freq_corde` que l'on va perdre (ce sera l'argument `a` de la fonction), puis on fait la mesure, et on passe en argument la longueur `N` de la liste et la nouvelle mesure `b`.

```python
def moyenne_optimisee(M,N,a,b):
    """M contient la moyenne de N termes, dont a
       La fonction renvoit la moyenne de N termes, en remplacant a par b
          """
    return (M+(-a+b)/float(N))
```

La fonction comporte 3 opération : elle est en $O(1)$, ne dépendant pas de N.

Un calcul classique de moyenne s'écrirait $\frac{1}{N}\sum k = 0^{N-1}U_k$: il y a N opération, ce serait donc en $O(N)$.

Exercice 4

Suite à un petit problème d'encodage, j'ai supprimé les accents, mais il faut évidemment les mettre.

```python
""" Usage : stoquer U[0] dans a avant de le perdre, faire la mesure,
    puis N=len(N) et b=U[N-1]"""

N=100
resultat=np.empty((N,0))

import matplotlib.pyplot as plt

plt.plot(resultat[:,0],resultat[:,1],'r') #'r' pour red
```

```
10 plt.xlabel('Duree (en jours)')
11 plt.ylabel('Deplacement de la fissure (en m)')
```