

On charge les bibliotheques dont on va avoir besoin.

```
In [1]: import numpy as np
```

On charge l'image

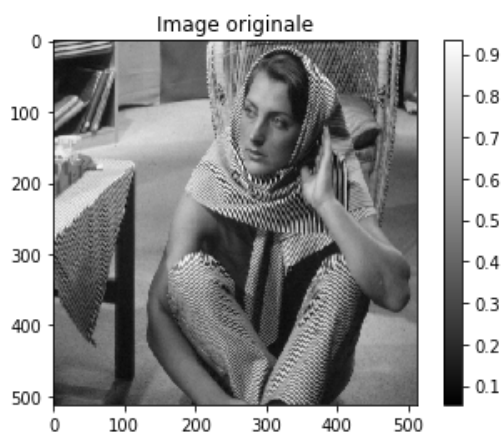
```
In [2]: from scipy import ndimage as ndim
#on charge l'image de lena à l'aide de la fonction imread de la bibliothèque nd
image de scipy
y =ndim.imread("barb.bmp")

#on ne garde que la première composante car niveau de gris
y=y[:,,:];
#l'image est de type int on la cast en double pour pouvoir faire des calculs
y=y.astype(float)
y=y/255

#la taille de l'image peut s'obtenir à l'aide de l'attribut shape
[n1,n2]=y.shape
```

On affiche l'image

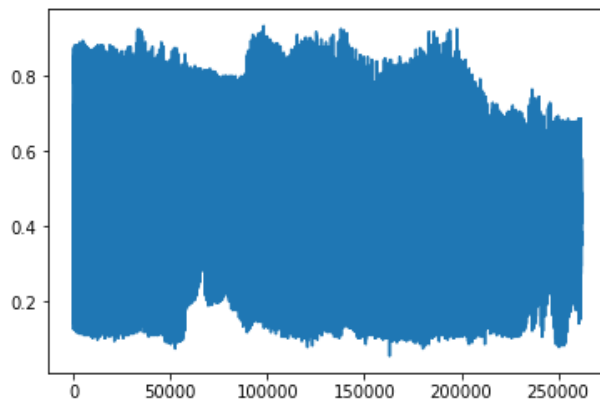
```
In [3]: import matplotlib.pyplot as plt
#on affiche l'image à l'aide de la fonction imshow de la bibliothèque pyplot de
matplotlib
#on donne un numero à la figure
plt.figure(1)
#on l'affiche en niveau de gris
plt.imshow(y, cmap="gray")
#on affiche la colorbar associée
plt.colorbar()
#on lui donne un titre
plt.title('Image originale')
#on affiche tout cela sur la figure 1
plt.show()
```



On redimensionne le tableau  $y$  qui contient l'image pour en faire un vecteur  $u$  de taille  $n_1 n_2$ .

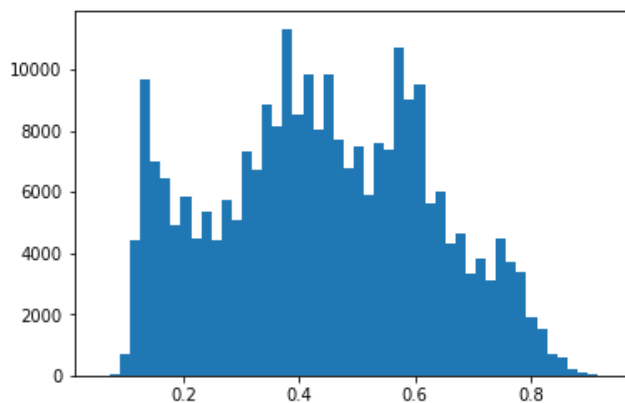
```
In [5]: u=np.reshape(y,n1*n2,1)
```

```
In [14]: # on visualise les valeurs prises par les pixels
plt.plot(u)
plt.show()
```



On trace l'histogramme des valeurs des pixels stockés dans  $u$ .

```
In [7]: plt.hist(u, bins=50)
plt.show()
```



On voit que toutes les valeurs entre 0 et 1 sont prises par l'image.

On change de base dans l'espace vectoriel  $\mathbb{R}^{n1n2}$

```
In [8]: # on charge la librairie ondelettes
import pywt

# on calcule l'ondelette de base
wavelet = pywt.Wavelet('db2')

#on calcule les coefficients de decomposition dans la base d'ondelette
coeffs = pywt.wavedec2(y, wavelet, mode = "periodization", level = 3)
```

On visualise les amplitudes des coefficients

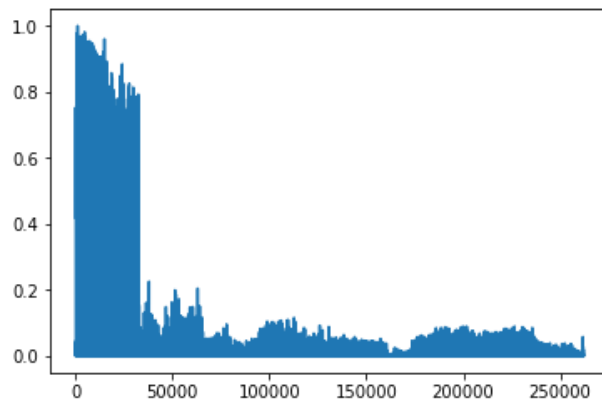
```
In [9]: arr, sli = pywt.coeffs_to_array(coeffs)
```

On normalise les coefficients entre 0 et 1 pour comparer a ce qu'on avait avant.

```
In [10]: arrn=arr/np.max(np.abs(arr))
```

```
In [11]: ucoeffs=np.reshape(arrn,n1*n2,1)
```

```
plt.plot(np.abs(ucoeffs))  
plt.show()
```



On fait l'histogramme des coefficients

```
In [12]: ucoeffs=np.reshape(arrn,n1*n2,1)  
plt.hist(np.abs(ucoeffs), bins=50)  
plt.title('histogramme des coefficients en ondelettes')  
plt.show()
```

