INFORMATIQUE PSI P2B

Année 2023/24

TP n° 6

Transformée de Fourier

Vincent ROHART

http://vrohart.e-monsite.com/

Il pourra être utile de créer une liste à N *éléments contenant que des 0, vus comme des nombres complexes :*

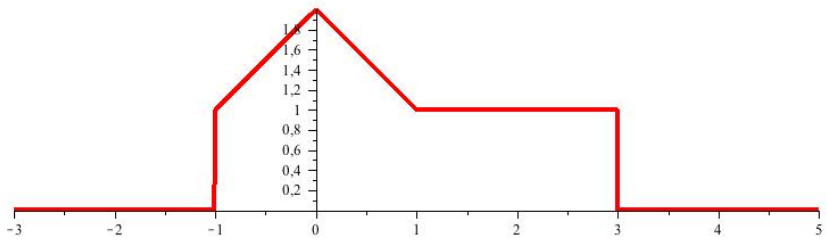
```
np.zeros(N,dtype=complex)
```

1. Programmer une fonction « triangle » f comme représentée page 5.

2. En déduire une fonction f_ε, pour tout $\varepsilon > 0$ qui soit une impulsion, afin d'approcher la distribution de Dirac δ_0.

3. Représenter f_ε pour $\varepsilon \in \{2^{-k} \mid k \in [\![1, 6]\!]\}$.

4. Programmer une fonction g (comme grange) dont la représentation graphique est la suivante.



5. Programmer la convolée $f_\varepsilon \star g$ et mettre en évidence que plus ε tend vers 0 , plus $f_\varepsilon \star g$ tend vers g.

6. Calculer la transformée de Fourier (= le spectre) de la fonction « triangle » f et représenter le module de ce spectre.

7. Augmenter la discrétisation de la fonction f en prenant $\mathrm{N} \in \{512, 1024, 2048, 4096\}$ (représenter le module des spectres correspondants sur un même graphique).

8. Représenter les spectres précédents dans la bonne indexation des indices k : on parle de spectre redressé. En effet, on rappelle que Python calcule $\hat{f}_k$ avec $k \in \left[-\frac{\mathrm{N}}{2}, \frac{\mathrm{N}}{2} - 1\right]$ **en mettant les valeurs négatives de k en fin de liste**. Autrement dit, $\hat{f}_k$ sera en position $k + \mathrm{N}$ dans la liste représentant le spectre.

9. Mettre en évidence, toujours avec la fonction f, la formule d'inversion de Fourier. Autrement dit, calculer la TF du spectre de f, et comparer graphiquement avec f. On prendra encore $\mathrm{N} \in \{512, 1024, 2048, 4096\}$.