Au cours de la deuxième moitié du 20ème siècle, le traitement de l'information est progressivement passé du tout analogique au tout numérique. Avec l'avènement de l'informatique moderne (et notamment des moyens de stockage de masse), il est devenu possible de garder en mémoire des signaux physiques stockés sous forme de suite de nombres, permettant ainsi une manipulation et une reproduction aisée de l'information.

Cela dit, le passage à un traitement de l'information numérique ne se fait pas sans contreparties et sans précautions. Il est nécessaire de comprendre les phénomènes qui surviennent lorsqu'on étudie un signal sous forme d'une suite finie de valeurs.

TABLE DES MATIERES

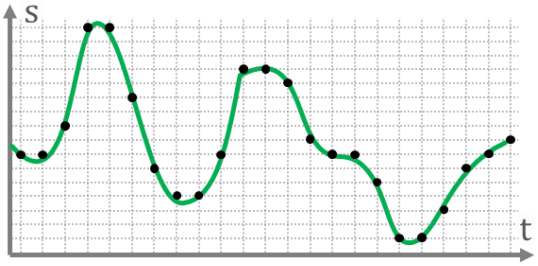
I - NOTION DE SIGNAL ET DÉCOMPOSITON DE FOURIER ---- - 1 -
I.1 - Signaux analogiques et numériques ---- - 1 -
I.2 - Numérisation d'un signal ---- - 2 -
II - ÉCHANTILLONNAGE D'UN SIGNAL ---- 2 -
II.1 - Définition et approche qualitative ---- - 2 -
II.2 - Spectre d'un signal échantillonné ---- - 3 -
II.3 - Repliement spectral ---- - 5 -
II.4 - Éviter le repliement spectral ---- 7 -
III - QUANTIFICATION D'UN SIGNAL ---- 8 -
IV - ANNEXE PRATIQUE : LA FFT AVEC PYTHON ---- 8 -

I - NOTION DE SIGNAL ET DÉCOMPOSITON DE FOURIER

I.1 - Signaux analogiques et numériques

Signal analogique, signal numérique

- Un signal analogique est une grandeur physique étudiée sous la forme d'une fonction $\mathbb{R} \to \mathbb{R}$. Dans la suite, on considèrera des signaux dépendants du temps, notés $s(t)$.
- Un signal numérique est une grandeur physique est une grandeur physique étudiée sous forme d'une fonction associant un ensemble fini de valeurs de $\mathbb{R}$ à un ensemble fini de valeurs de $\mathbb{R}$.



Signal analogique en vert
Signal numérique correspondant en noir

On pourrait tout à fait étendre la définition d'un signal à n'importe quelle grandeur physique représentée par une fonction $\mathbb{C}^k \to \mathbb{C}^n$, voire des cas encore plus généraux. En PT, on se limitera aux cas de grandeurs scalaires réelles dépendant du temps.

Puisqu'un intervalle de $\mathbb{R}$ contient toujours une infinité de valeurs, un signal physique analogique, même très court, ne peut pas être stocké dans un ordinateur[1]. Tout signal analogique stocké ou transformé par un ordinateur doit d'abord être converti en un signal numérique, stocké sur un nombre fini de valeurs :

t_i	10,00	10,15	10,30	...	19,70	19,85	20,00	Exemple de signal réel numérisé pour $t \in [10,20]$ s, tous les $\Delta t = 0{,}15$ s, stocké avec une précision de 0,002.
s_i	0,780	0,792	0,798	...	1,112	1,100	1,016	

I.1.A - Intérêt de la numérisation

Les intérêts de la numérisation sont multiples :

- Le stockage massif est aisé : il est possible de stocker 360 heures de fichier audio entre 0 et 40 kHz sans perte sur un disque dur de 1 To. Stocker la même quantité de données en format analogique (sur bande magnétique, ou vinyle) requerrait un volume conséquent.
- Les calculs et modifications du signal sont facilités, et non-destructifs : sur ordinateur, il est possible appliquer n'importe quel algorithme de compression, distorsion, etc. à un signal numérique, sans difficulté, et de manière réversible.

[1] On pourrait bien sûr penser stocker la fonction mathématique décrivant le signal, e.g. $s(t) = \sin(\omega t)$, mais les signaux réels ne suivent pas parfaitement les fonctions mathématiques définies par les modèles qui tentent de les décrire (quand un modèle existe…)

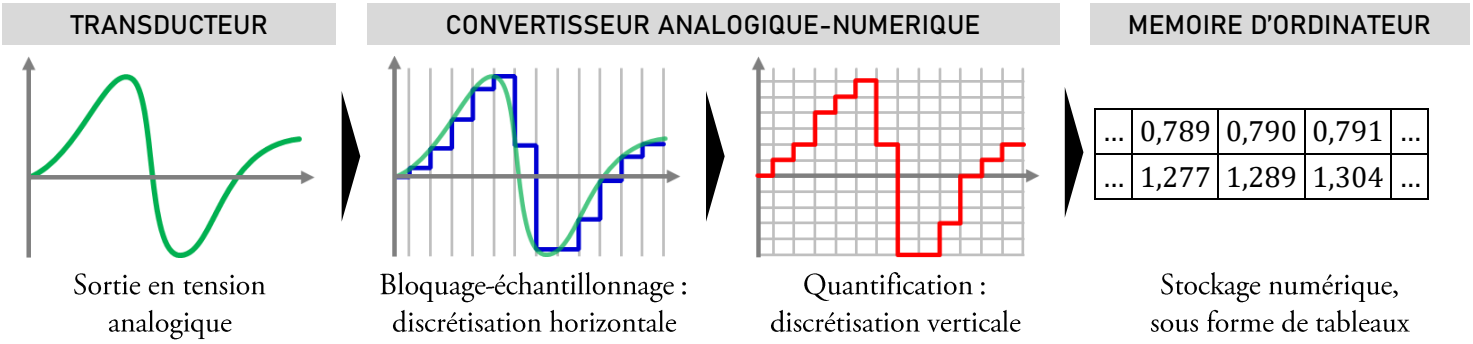
- La transmission et la copie sont facilitées et sûres : puisqu'un signal numérique varie par paliers, il est moins sensible aux perturbations extérieures, car il sera toujours possible de distinguer les différents paliers.

Dans la suite du chapitre, on s'intéressera à la manière dont est réalisée ce passage du signal analogique à numérique, puis aux écueils causés par le traitement numérique de l'information.

I.2 – Numérisation d'un signal

La conversion d'un signal physique quelconque (pression, vitesse, température, etc.) en un signal numérique suit toujours les mêmes étapes :

- Le signal analogique passe par un *transducteur*, qui convertit la grandeur physique en une tension (encore analogique) ;
- Cette tension passe par un *convertisseur analogique-numérique* (comme la carte d'acquisition SYSAM), qui effectue une discrétisation horizontale, puis verticale.



Sortie en tension analogique | Bloquage-échantillonnage : discrétisation horizontale | Quantification : discrétisation verticale | Stockage numérique, sous forme de tableaux

Puisque la majeure partie des problèmes liés à la numérisation d'un signal viennent de l'échantillonnage, on étudiera en détail cette partie dans la section suivante. La section qui suivra portera sur la quantification de manière plus succincte.

II – ÉCHANTILLONNAGE D'UN SIGNAL

II.1 – Définition et approche qualitative

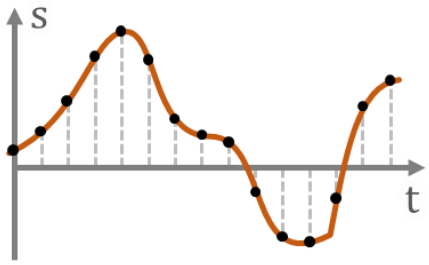
Puisqu'un signal ne peut être traité numériquement que s'il est stocké sous forme d'un tableau de valeurs en mémoire, l'une des premières étapes du traitement numérique des signaux est de prélever un ensemble fini de valeurs censées représenter fidèlement le signal à traiter.

Échantillonnage, paramètres d'échantillonnage

Échantillonner un signal analogique signifie prélever sa valeur à intervalles réguliers t_k ($k \in \mathbb{N}$) pendant un intervalle de temps Δt. Les paramètres d'échantillonnage sont :

- La fréquence d'échantillonnage T_e.
- La fréquence d'échantillonnage $f_e = 1/T_e$

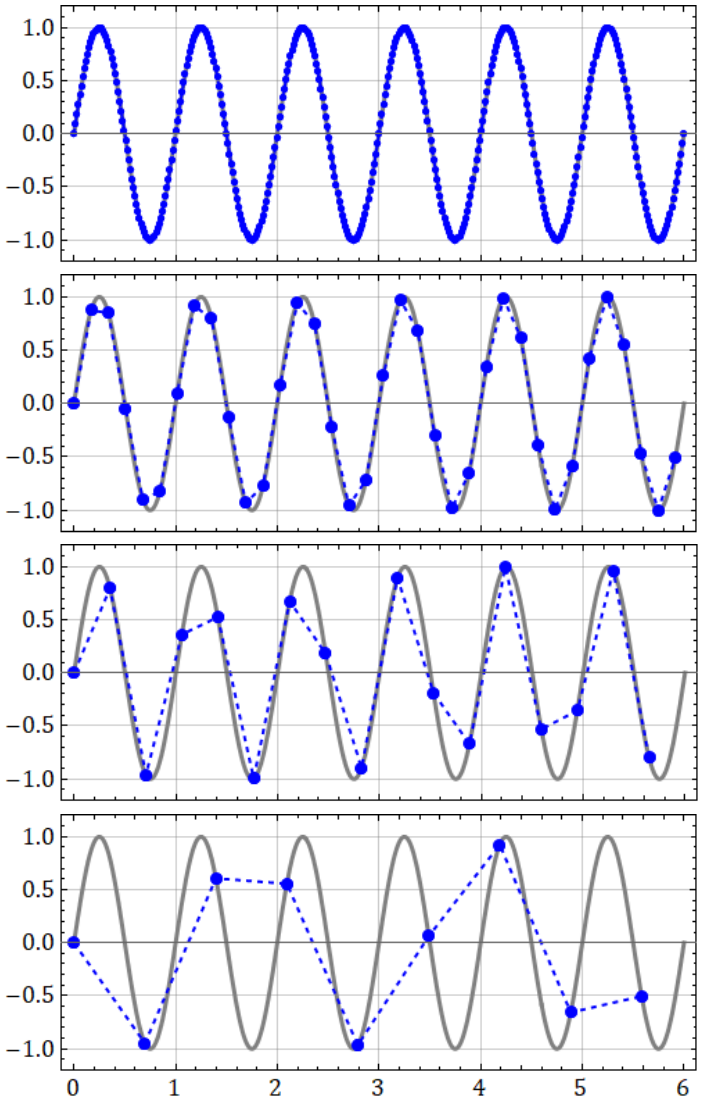
On a donc $t_k = k \cdot T_e$, et le nombre total de valeurs prélevées est $N_e = \Delta t/T_e$.



Le choix des paramètres d'échantillonnage répond à deux contraintes principales qui s'opposent. Le choix de T_e doit :

- Être suffisamment faible pour représenter fidèlement le signal réel ;
- Être suffisamment grand pour ne pas saturer la mémoire de l'ordinateur ou du logiciel avec lequel on numérise le signal.

Dans les exemples ci-dessous, on met en évidence les contraintes liées au choix de la fréquence d'échantillonnage, en prenant l'exemple d'un signal réel, de forme $s(t) = \sin(2\pi f_0 t)$ avec $f_0 = 1$ Hz, échantillonné en utilisant diverses valeurs de fréquence d'échantillonnage f_e.



On échantillonne à la fréquence $f_e \simeq 66,7$ Hz :

Le signal échantillonné est très proche du signal réel. Il est possible de reconstituer le sinus à partir du signal échantillonné. Cela dit, si l'acquisition dure longtemps, le signal échantillonné sature rapidement la mémoire vive allouée au logiciel utilisé.

On échantillonne à la fréquence $f_e \simeq 5,9$ Hz :

Le signal échantillonné est proche du signal réel. Il est possible de reconstituer le sinus à partir du signal échantillonné.

On échantillonne à la fréquence $f_e \simeq 2,8$ Hz :

Le signal échantillonné n'est plus proche du signal réel. Il est difficile de reproduire le sinus à partir de ce signal. Cela dit, il est toujours possible de lire la fréquence f_0 du sinus dans le signal échantillonné.

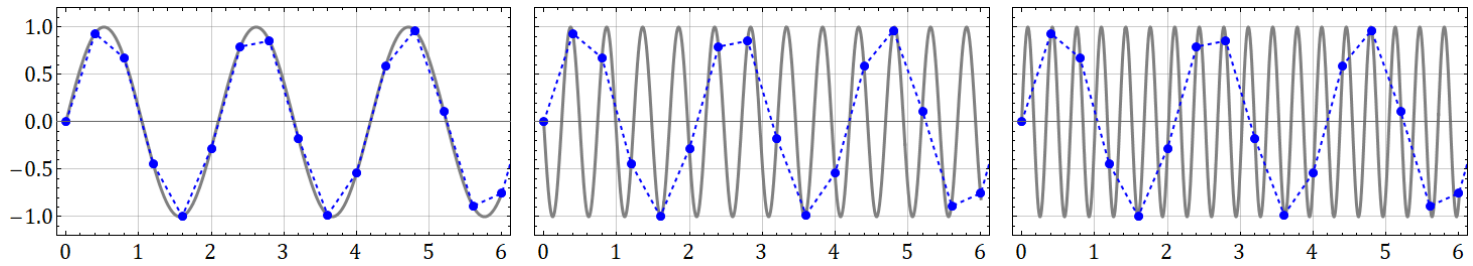
On échantillonne à la fréquence $f_e \simeq 1,4$ Hz :

Le signal échantillonné n'est plus proche du signal réel. Il est difficile de reproduire le sinus à partir de ce signal.

Pire, il est **impossible de retrouver la fréquence f_0** du sinus à partir du signal échantillonné.

Il existe donc au moins un critère à respecter lorsqu'on souhaite échantillonner un signal sans perdre trop d'informations. Il convient d'ajouter que même en présence d'un ordinateur sans limitation de mémoire vive, il est très souvent préférable de ne pas prendre une fréquence d'échantillonnage trop grande, car cela peut amener certaines inconvenances : ajout de bruit numérique, lenteurs dans le traitement des données en aval, etc.

On peut démontrer une particularité qui prendra plus tard une grande importance : à un même signal échantillonné peuvent correspondre plusieurs signaux réels ! On a représenté ci-dessous trois exemples.



$f_e = 0,5$ Hz | $f_0 = 0,5$ Hz $f_e = 0,5$ Hz | $f_0 = 2$ Hz $f_e = 0,5$ Hz | $f_0 = 3$ Hz

II.2 - Spectre d'un signal échantillonné

II.2.A - Spectre d'un signal analogique

On rappelle que selon la théorie de Fourier, tout signal périodique de période $T_0 = 2\pi/\omega_0$ peut s'écrire comme une série infinie :

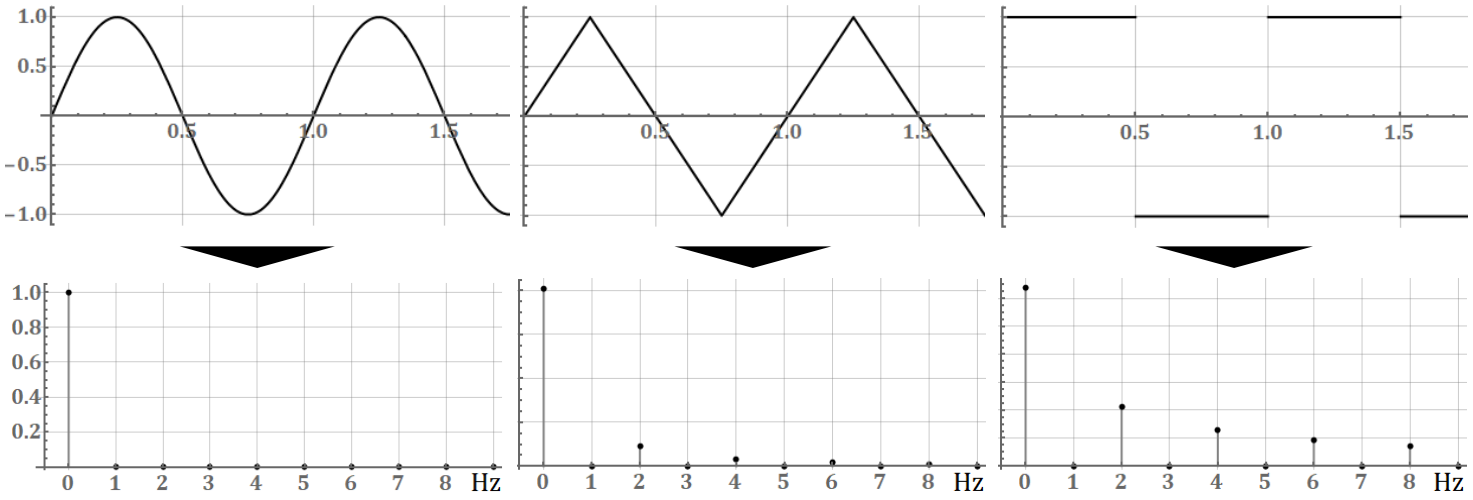
$$s(t) = \sum_{i=0}^{\infty} s_n \cos(n\omega_0 t + \varphi_n) = s_0 + \underbrace{s_1 \cos(\omega_0 t + \varphi_0)}_{\text{harmonique 1 (fondamental)}} + \underbrace{s_2 \cos(2\omega_0 t + \varphi_0)}_{\text{harmonique 2}} + \underbrace{s_3 \cos(3\omega_0 t + \varphi_0)}_{\text{harmonique 3}} + \cdots$$

La plupart du temps, on s'intéressera aux amplitudes de chaque harmonique, sans regarder les phases, dont l'intérêt est souvent moindre.

Spectre d'un signal

Le spectre d'un signal est la représentation des amplitudes des harmoniques du développement de Fourier.
On pourra représenter s_n, $|s_n|$ ou s_n^2 en fonction de l'indice n correspondant à l'harmonique de pulsation $n\omega$.

Qualitativement, plus le signal est proche d'une sinusoïde, plus ses amplitudes s_n décroissent rapidement avec n. Par exemple, dans les exemples ci-dessous, on voit que le spectre est d'autant plus riche que le signal est éloigné de la sinusoïde :



On retrouve ci-dessus le fait qu'un signal harmonique, purement sinusoïdal, est son propre développement de Fourier, et possède donc une unique harmonique.

Le développement de Fourier est unique et précisément déterminé lorsqu'on fait le développement d'un signal mathématique (donc parfaitement détermine $\forall t \in]-\infty, +\infty[$). Ce n'est pas le cas lorsqu'on calcule la transformée de Fourier d'un signal échantillonné.

II.2.B - Spectre d'un signal échantillonné

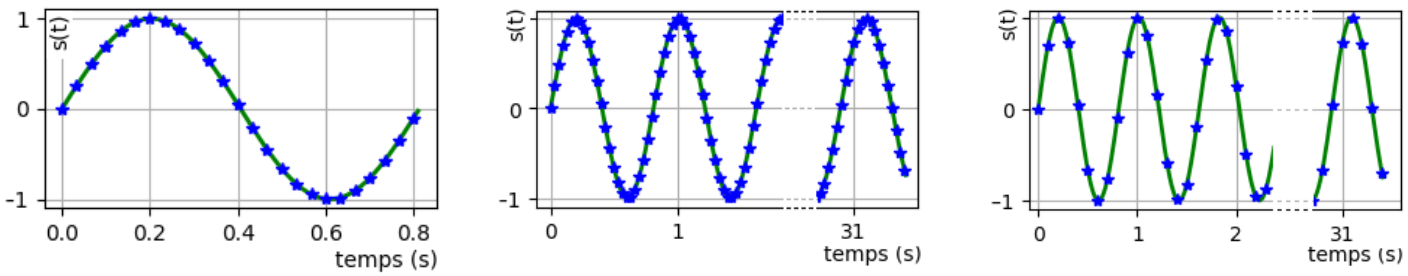
Pour un signal échantillonné, uniquement connu pour un nombre fini de temps t_n ($n \in [0, N]$), il est impossible d'appliquer les formules de calcul des coefficients de Fourier (hors-programme). On applique alors un algorithme dit de « transformée de Fourier rapide » (plus communément *FFT*, pour *Fast Fourier Transform*), qui s'applique à un ensemble fini de N valeurs, et renvoie un ensemble de N valeurs représentant le spectre du signal en fonction des paramètres ci-dessous :

Spectre FFT d'un signal échantillonné

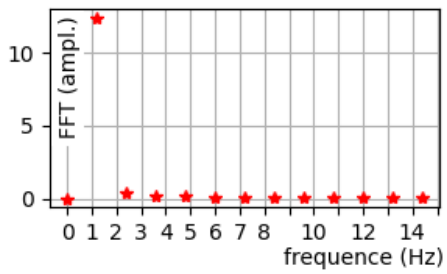
Un signal réel échantillonné dans un temps $t \in [0, t_f]$ avec une période d'échantillonnage $T_e = 1/f_e$, donne un spectre FFT réel avec les fréquences $f \in \left[0, \frac{f_e}{2}\right]$ et une « résolution en fréquence » $\Delta f = 1/t_f$.

Exemple – Divers paramètres d'échantillonnage et spectre FFT

On considère un signal de fréquence $f_0 = 1{,}23$ Hz (donc $T_0 = 0{,}81$ s). On l'échantillonne avec les divers paramètres d'échantillonnage : durée totale T_{tot}, période et fréquence d'échantillonnage T_e, f_e.

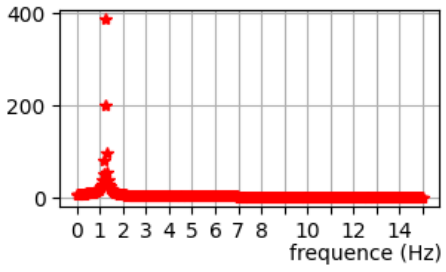


$T_{tot} \sim T_0$ et $f_e = 30$ Hz | $T_{tot} \simeq 38\,T_0$ et $f_e = 30$ Hz | $T_{tot} \simeq 38\,T_0$ et $f_e = 10$ Hz



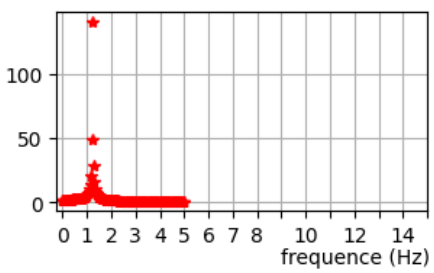
$\mathbf{f \in [0, f_e/2] = [0, 15]\ Hz}$

$\mathbf{\Delta f = \frac{1}{T_0} \simeq 1,2\ Hz}$



$\mathbf{f \in [0, f_e/2] = [0, 15]\ Hz}$

$\mathbf{\Delta f = \frac{1}{38\ T_0} \simeq 0,03\ Hz}$



$\mathbf{f \in [0, f_e/2] = [0, 5]\ Hz}$

$\mathbf{\Delta f = \frac{1}{38\ T_0} \simeq 0,03\ Hz}$

Paramètres d'échantillonnage, paramètres du spectre FFT

Les paramètres du spectre FFT (ensemble de définition et Δf) ne sont pas intuitivement liés aux paramètres d'échantillonnage :

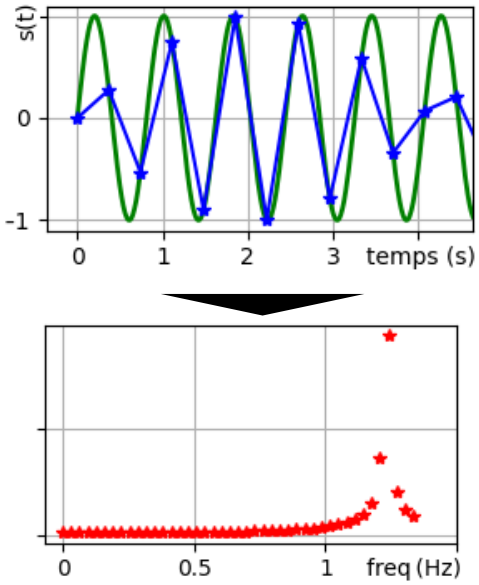
- Plus T_e est petit (ou f_e grand), plus l'ensemble de définition $[0, f_e/2]$ du spectre FFT est grand ;
- Plus la durée d'échantillonnage $[0, t_f]$ est grande, plus la précision Δf du spectre FFT est grande.

Avant d'étudier le spectre d'un signal échantillonné, on prendra gare à **choisir les paramètres d'échantillonnage** de sorte à obtenir les caractéristiques voulues.

II.3 - Repliement spectral

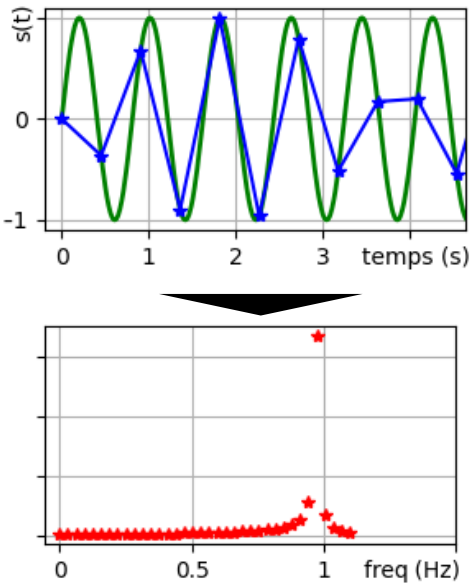
Dans cette partie, on s'intéresse à l'effet sur le spectre FFT d'un mauvais choix de paramètres d'échantillonnage, comme représenté dans la partie II.1. On reprend l'exemple d'une simple sinusoïde de fréquence $f_0 \simeq 1{,}23$ Hz, échantillonnée avec divers paramètres :

$f_e \simeq 2{,}7$ Hz $T_{tot} \simeq 38\ T_0 \simeq 31$ s



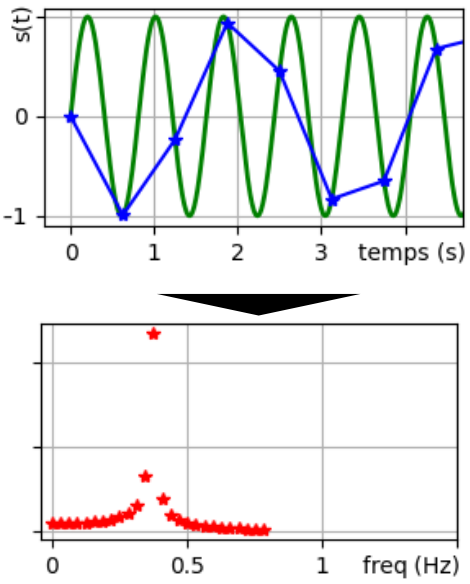
On distingue la fréquence du signal à $f \simeq f_0 \simeq 1{,}23$ Hz

$f_e \simeq 2{,}2$ Hz $T_{tot} \simeq 38\ T_0 \simeq 31$ s



On ne distingue plus la fréquence f_0. Pire, une fréquence inconnue apparait.

$f_e \simeq 1{,}6$ Hz $T_{tot} \simeq 38\ T_0 \simeq 31$ s



On ne distingue plus la fréquence f_0. La fréquence inconnue s'est déplacée !

II.3.A - Duplication du spectre FFT et critère de Shannon

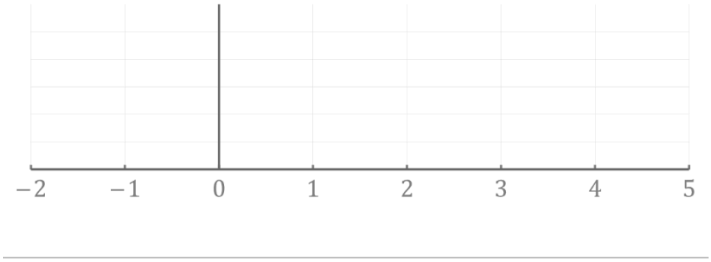
Duplication du spectre FFT

Puisqu'à un même échantillonnage correspondent plusieurs signaux réels, un mauvais choix d'échantillonnage peut conduire à l'apparition de fréquences non présentes dans le signal initial.

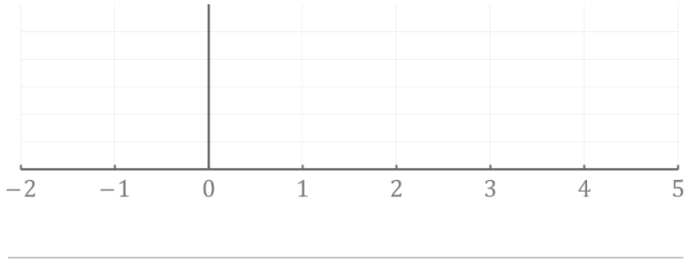
On peut montrer que ces fréquences sont situées à $\mathbf{f = k\ f_e \pm f_0}$ $(k \in \mathbb{N})$

Dès qu'un signal à fréquence f_0 est échantillonné, on s'attend donc à obtenir des fréquences non-physiques. On représente ci-dessous l'échantillonnage d'un signal de fréquence $\mathbf{f_0 \simeq 1,23\ Hz}$:

Spectre FFT dans le cas où $f_e \simeq 3$ Hz



Spectre FFT dans le cas où $f_e \simeq 1,6$ Hz



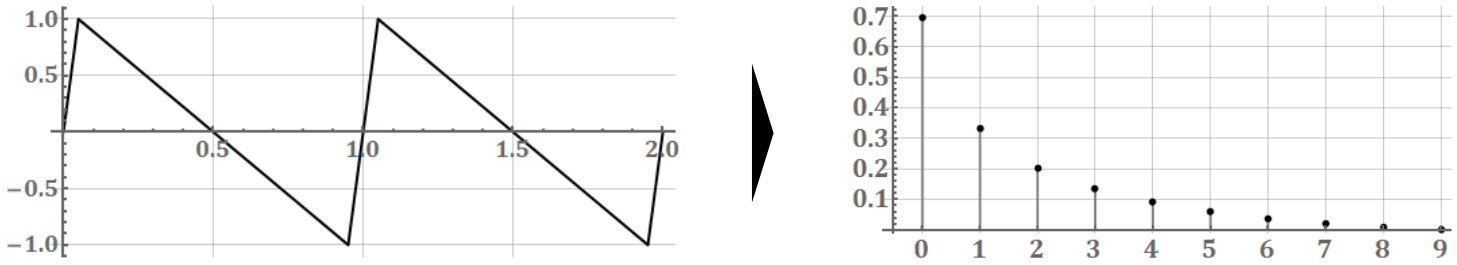
On en déduit un critère simple permettant d'éviter ce phénomène de **« repliement spectral »** :

> **Critère de Shannon**
>
> Pour éviter le **repliement spectral**, la fréquence d'échantillonnage doit vérifier : $\mathbf{f_e > 2 \cdot f_{max}}$, où f_{max} est la fréquence maximale du signal qu'on souhaite étudier.

Dans le cas d'un signal sinusoïdal de fréquence f_0, comme dans les exemples ci-dessus, éviter le repliement spectral est une tache simple. En revanche, lors de l'étude de signaux plus complexes, il est facile d'oublier ce critère.

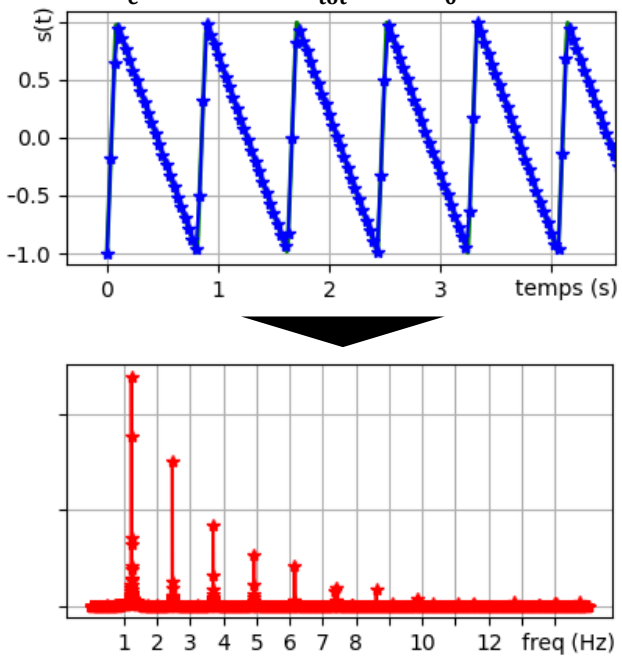
II.3.B – Repliement spectral d'un signal quelconque

On prend l'exemple d'un signal « dents de scie », représenté ci-dessous avec une période de $T_0 = 1$ s (et une allure 10 % montante et 90 % descendante) :

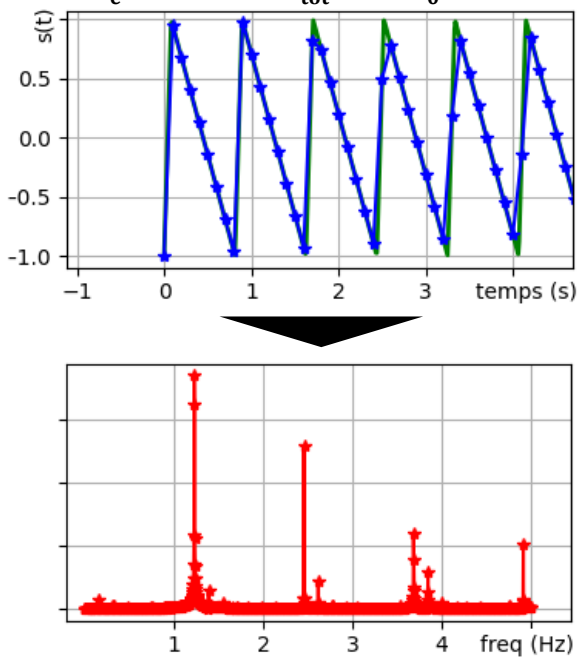


Dans les deux exemples ci-dessous, on choisit une fréquence d'échantillonnage f_e qui respecte le critère de Shannon. On trace alors le spectre du signal échantillonné :

$f_e \simeq 30$ Hz $T_{tot} \simeq 38\ T_0 \simeq 38$ s

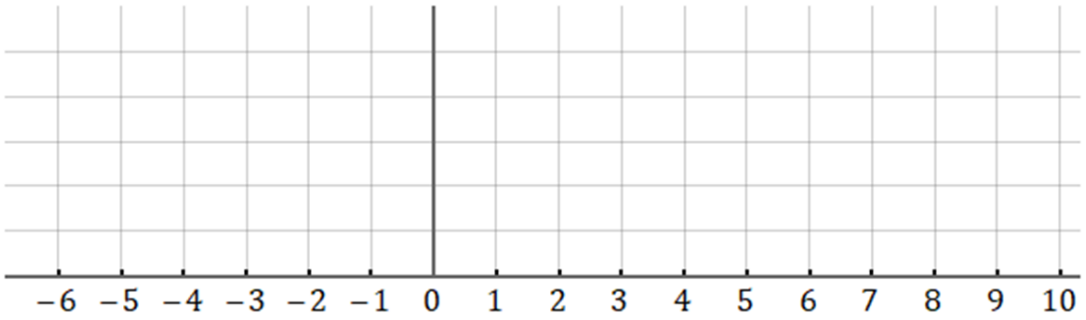


$f_e \simeq 10$ Hz $T_{tot} \simeq 38\ T_0 \simeq 31$ s



On constate que, bien qu'on respecte le critère de Shannon concernant la fréquence fondamentale (l'harmonique n°1), on ne le respecte pas pour certaines des fréquences intervenant dans la décomposition de Fourier. On observe un **repliement spectral** des harmoniques à hautes fréquences.

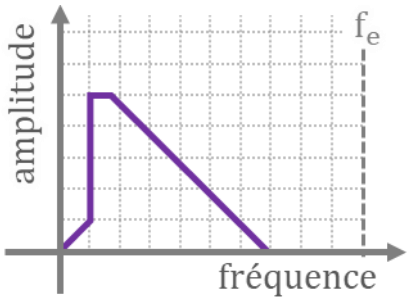
On représente ci-dessous le spectre du **signal dent de scie**, de fréquence $\mathbf{f_0 = 1\ Hz}$ (et son repliement) observé pour $\mathbf{f_e = 10\ Hz}$:



Repliement spectral pour un signal quelconque

On appelle **repliement spectral** ou **aliasing** la superposition du spectre physique d'un signal avec une de ses répliques lors du calcul d'un spectre FFT d'un signal échantillonné.

Ce phénomène n'est pas corrigeable *a posteriori* ; il doit être **anticipé avant de procéder à l'échantillonnage**.



Remarque : tout se passe comme si le spectre réel était « replié » sur lui-même lorsqu'il atteint la limite de l'ensemble de définition $[0, f_e/2]$: une partie du signal n'a alors pas de sens physique.

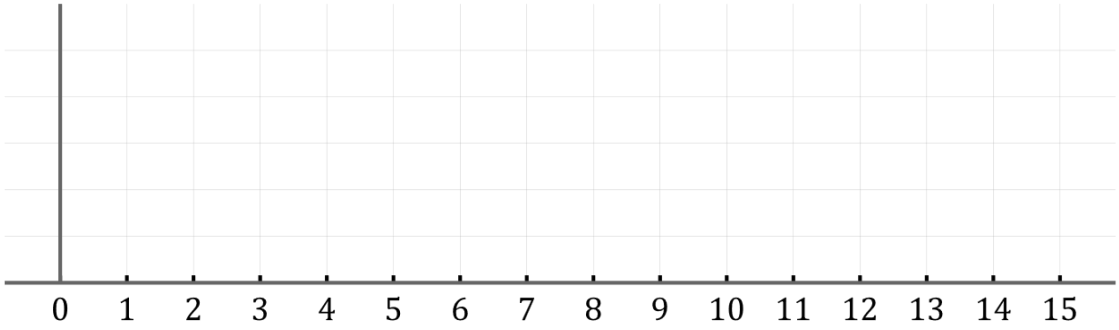
Application - Déterminer s'il y a repliement

On considère un signal carré d'amplitude U_0 et de fréquence $f_0 = 2$ kHz dont la décomposition de Fourier s'écrit :

$$s(t) = \frac{4U_0}{\pi} \sum_{i=0}^{\infty} \frac{1}{2i+1} \sin(2\pi f_0 (2i+1)t)$$

Le signal est échantillonné à $f_e = 14$ kHz pendant une durée de 0,1 s. On souhaite connaitre l'allure du spectre FFT et du repliement spectral éventuel.

1. Écrire les quelques premiers termes de la série de Fourier.
2. Déterminer les paramètres du spectre FFT (précision Δf et intervalle de définition).
3. Dessiner le spectre et l'éventuel repliement spectral.



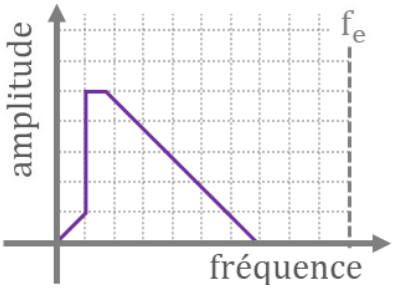
Remarque : bien que le repliement spectral soit la principale cause de problèmes, il existe d'autres effets (hors-programme) pouvant créer des pics non-physiques dans le spectre FFT.

II.4 - Éviter le repliement spectral

Puisque le repliement spectral ne peut pas être enlevé après l'échantillonnage, il est nécessaire d'agir sur le signal réel en amont.

Éviter le repliement spectral via un passe-bas

Puisque le repliement spectral touche les fréquences $f > f_e/2$, on pourra appliquer un **passe-bas** au signal temporel afin de supprimer **en amont** les fréquences au-delà de cette limite (en prenant une pulsation de coupure du filtre $\mathbf{f_c \simeq f_e/2}$)



III - QUANTIFICATION D'UN SIGNAL

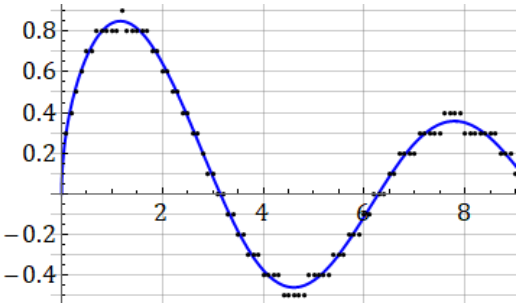
De la même manière qu'un signal physique analogique prend une continuité de valeurs en abscisse (pour le temps), il peut prendre une continuité de valeurs en ordonnée (pour la grandeur physique mesurée, le plus souvent convertie préalablement en tension).

> **Quantification d'un signal analogique**
>
> La quantification d'un signal correspond à la discrétisation des valeurs que peut prendre la grandeur physique mesurée. Le plus petit intervalle de valeurs est appelé le « pas de quantification ».

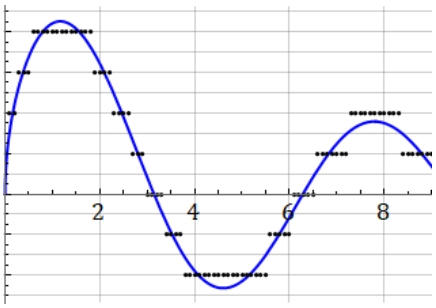
Le plus souvent, le processus de quantification ne peut se faire qu'après avoir renseigné les bornes de l'intervalle de quantification dans lequel le signal pourra être quantifié. Ce choix influe sur la précision avec laquelle le signal sera stocké en mémoire.

On représente ci-dessous l'allure d'un signal analogique (en trait continu) sur le quel on applique plusieurs paramètres de quantification :

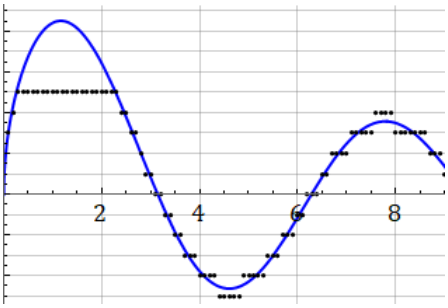
Pas de quantification $\Delta V = 0.1$ V
Intervalle $[-1, 1]$



Pas de quantification $\Delta V = 0.2$ V
Intervalle $[-1, 1]$



Pas de quantification $\Delta V = 0.1$ V
Intervalle $[-0,5 ; 0,5]$



La plupart du temps, le pas de quantification est déterminé par le calibre et la résolution de l'acquisition :

- Le calibre C renseigne l'intervalle $[-C, C]$ que le signal numérisé peut prendre. Il doit être supérieur à la valeur maximale du signal analogique, sans quoi le signal numérisé fait apparaître un phénomène de saturation (comme sur l'image de droite ci-dessus). La taille de l'intervalle, égale à $2C$, est la « tension de pleine échelle » du CAN.
- La résolution N indique le nombre de bit sur lequel est codé le signal : un codage sur N bits permet au signal de prendre 2^N valeurs. Le pas de quantification est alors $\Delta V = \frac{2C}{2^N - 1} \simeq \frac{2C}{2^N} \simeq \frac{C}{2^{N-1}}$.

Exemple : on parle parfois d'une image codée en 8 bit (par exemple pour une image stockée au format jpeg). Cela signifie que chaque sous-pixel (bleu, rouge, et vert) donne une valeur de luminosité codée sur 8 bit, c'est-à-dire entre 0 et $2^8 = 256$. Ainsi, le nombre de couleurs possible sur une image 8 bit est $(2^8)^3 \simeq 16{,}7 \cdot 10^7$. On parle parfois d'image « 16 millions de couleurs ».

IV - ANNEXE PRATIQUE : LA FFT AVEC PYTHON

La FFT est classiquement appelée via le module Scipy, importé par `from scipy import signal`. Le signal échantillonné est représenté par un simple array Numpy, importé par `import numpy as np`.

Pour réaliser la FFT d'un signal réél représenté par l'array nommé « signal », on utilise l'instruction `np.fft.rfft(signal)`, qui renvoie alors un tableau de réels représentant le spectre (la valeur absolue des amplitudes), sans préciser les ordonnées. Pour tracer le spectre, on a accès aux abscisses correspondantes via l'instruction `np.fft.rfftfreq(len(signal), dt)`, où le premier argument est la taille N du signal échantillonné, et le second est la période d'échantillonnage T_e.

L'array renvoyé est l'abscisse du spectre : l'intervalle $\left[0, \frac{f_e}{2}\right]$, avec un pas de $\Delta f = \frac{1}{t_f} = \frac{1}{NT_e}$. Le spectre peut alors être tracé via l'utilisation du sous-module Pyplot du module Matplotlib, appelé via l'instruction `import matplotlib.pyplot as plt`.