

Notes et fréquences

30 juin 2023

1 Les séries trigonométriques

Il y a plusieurs façons de définir les séries trigonométriques. La plus courante, si on veut éviter de passer par les nombres complexes, est :

Définition Une série trigonométrique est une série de fonctions dont les sommes partielles sont de la forme

$$S_N(t) = a_0 + \sum_{n=1}^N \left(a_n \cos\left(\frac{2\pi}{T}nt\right) + b_n \sin\left(\frac{2\pi}{T}nt\right) \right) \text{ avec } T > 0$$

Remarques S_N est une fonction périodique de période T .

Il existe une autre forme plus adaptée aux ondes sonores qui consiste à définir la série avec la seule fonction cos.

Définition 2 Une série trigonométrique est une série de fonctions dont les sommes partielles sont de la forme

$$S_N(t) = d_0 + \sum_{n=1}^N d_n \cos\left(\frac{2\pi}{T}nt + \varphi_n\right).$$

On passe de la première définition à la seconde en posant $d_0 = a_0$ et $d_n = \sqrt{a_n^2 + b_n^2}$; φ_n est défini à $2k\pi$ près par $\cos \varphi_n = \frac{a_n}{\sqrt{a_n^2 + b_n^2}}$ et $\sin \varphi_n = -\frac{b_n}{\sqrt{a_n^2 + b_n^2}}$.

Cette définition est intéressante pour les ondes sonores car l'oreille et le cerveau sont sensibles aux valeurs d_n ($n \geq 1$) mais pas aux phases φ_n .

2 Transformée de Fourier d'une fonction périodique

On démontre que toute fonction périodique de période T se décompose sous la forme d'une série trigonométrique.

De façon plus précise :

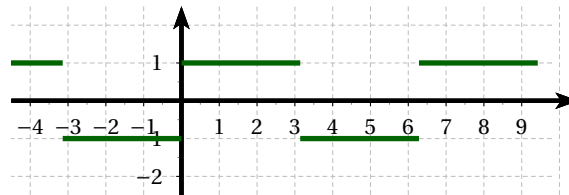
Théorème Si $s(t)$ est une fonction périodique de période $T > 0$ de classe \mathcal{C}_1 par intervalle, c'est à dire dérivable et de dérivée continue par intervalle, alors il existe une série de fonction $S_N(t)$ associée convergente vers $s(t)$.

Si s n'est pas continue en t alors $S_N(t)$ converge vers la moyenne des limites à droite et à gauche en t .

Exemples

Fonction « créneaux »

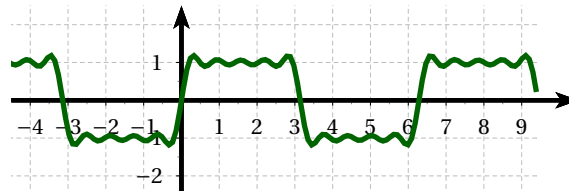
Il s'agit d'une fonction s définie par $s(t) = 1$ si t appartient à l'intervalle $[2n\pi, (2n + 1)\pi[$ et $s(t) = -1$ sinon.



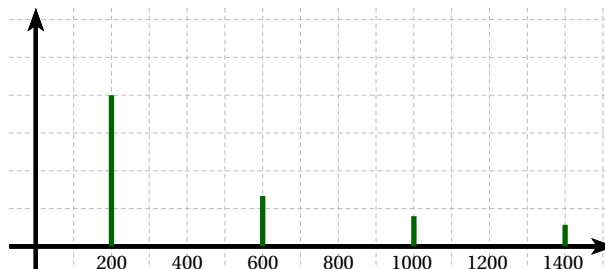
s est une fonction impaire et périodique de période 2π . On démontre que tous les a_n sont nuls. $b_n = 0$ si n est paire et $b_n = \frac{4}{\pi n}$ sinon. La série de Fourier associée pour $n = 10$ est :

$$S_{10}(t) = \frac{4}{\pi} \sin(t) + \frac{4}{3\pi} \sin(3t) + \frac{4}{5\pi} \sin(5t) + \frac{4}{7\pi} \sin(7t) + \frac{4}{9\pi} \sin(9t)$$

et a pour graphique

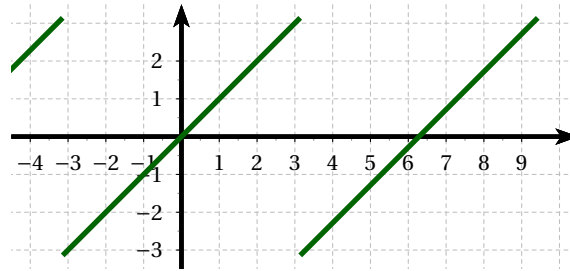


Le spectre de la fonction « créneaux » de 200 Hertz pourra être représenté par le graphique suivant



Fonction « dents de scie »

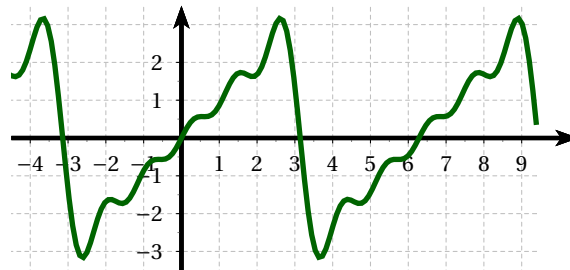
Il s'agit d'une fonction s définie par $s(t) = t$ sur l'intervalle $[-\pi, \pi[$ et périodique de période 2π .



C'est une fonction impaire. On démontre que tous les a_n sont nuls et que $b_n = -\frac{2}{n}(-1)^n$.
La série de Fourier pour $n = 5$ est

$$S_5(t) = 2 \left(\sin(t) - \frac{1}{2} \sin(2t) + \frac{1}{3} \sin(3t) - \frac{1}{4} \sin(4t) + \frac{1}{5} \sin(5t) \right)$$

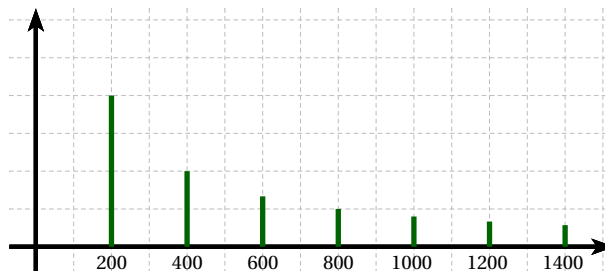
et a pour graphique



La fonction « dents de scie » (200Hertz) pourra être définie par la formule suivante

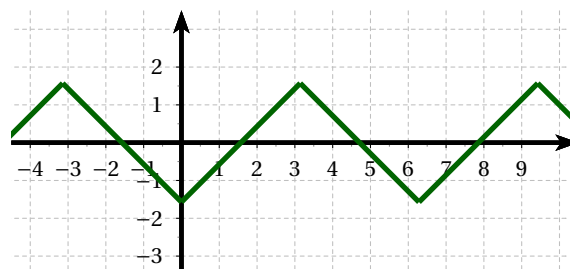
$$s(t) = \sin\left(\frac{2\pi}{0,005}t\right) + \frac{1}{2} \sin\left(\frac{2\pi}{0,005}2t + \pi\right) + \frac{1}{3} \sin\left(\frac{2\pi}{0,005}3t\right) + \frac{1}{4} \sin\left(\frac{2\pi}{0,005}4t + \pi\right) \dots$$

et son spectre pourra être représenté par le graphique suivant



Fonction « triangles »

Il s'agit d'une fonction s définie par $s(t) = |t| - \frac{\pi}{2}$ sur l'intervalle $[-\pi, \pi[$ et périodique de période 2π .



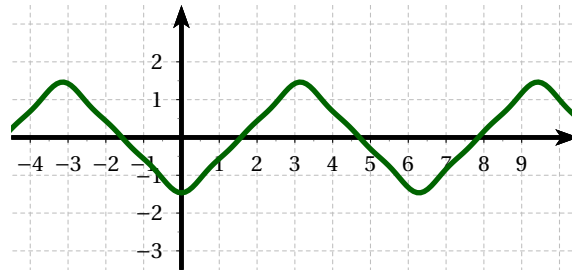
C'est une fonction paire. Tous les b_n sont nuls et $a_0 = 0$.

Si $n > 0$ est pair, $a_n = 0$ sinon, $a_n = -\frac{4}{\pi n^2}$.

La série de Fourier pour $n = 5$ est

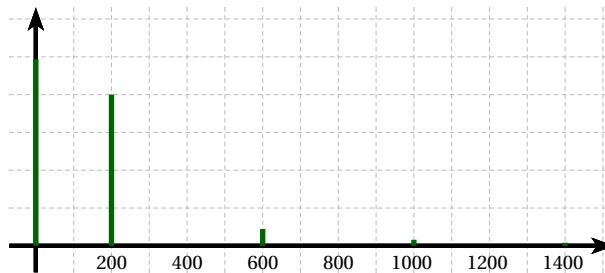
$$S_5(t) = -\frac{4}{\pi} \left(\cos(t) + \frac{1}{9} \cos(3t) + \frac{1}{25} \cos(5t) \right)$$

et a pour graphique



La convergence est très rapide grâce au coefficient $\frac{1}{n^2}$.

Le spectre de la fonction « triangles » (200 Hertz) pourra être représenté par le graphique suivant :



Utilisation de pages HTML écrites en Javascript

test-adsr.html

Il s'agit d'un simple testeur. Les sons générés sont tous identiques (de type sinusoïdal et de fréquence 220 Hz). On peut en générer autant qu'on veut en appuyant sur le bouton "son". On peut les arrêter, les uns après les autres en utilisant le bouton "stop".

notes-frequences-cossin-courbe-fft.html

Après avoir rentrés les a_n et b_n de la série trigonométrique selon la 1^{re} définition, ce programme affiche la courbe et le spectre du signal obtenu.

Pour une approximation d'une fonction « créneaux » il suffira de rentrer la série de nombres 1 0 0.333333 0 0.2 0 0.142857 0 0.111111 (correspondante à 1 0 1/3 0 1/5 0 1/7 0 1/9) dans le champ de saisie « sinus » et 0 dans le champ « cosinus », l'application complètera par des 0.

Pour la fonction « dents de scie » il suffira de rentrer la série 1 -0.5 0.333333 -0.25 0.2 dans le champ « sinus » (correspondante à 1 -1/2 1/3 -1/4 1/5) et 0 dans le champ « cosinus ».

Pour la fonction « triangles » il suffira de rentrer la série 1 0 0.111111 0 0.04 dans le champ « cosinus » (correspondante à 1 0 1/9 0 1/25) et 0 dans le champ « sinus ».

Notez aussi que pour simuler le son d'un cor la série 0 0.4 0.4 1 1 1 0.3 0.7 0.6 0.5 0.9 0.8 entrée au choix dans le champ « cosinus » ou dans le champ « sinus » convient très bien d'après <https://www.sitepoint.com/using-fourier-transforms-web-audio-api/>.

notes-frequences-cos-courbe-fft.html

Contrairement au programme précédent, on ne rentre ici que les termes a_n , les b_n étant mis automatiquement à 0.

Cela revient à n'utiliser que les cos de la 1^{re} définition ou la définition numéro 2 avec des phases égales à 0. Pour la génération d'un son ce programme est suffisant car nous ne sommes pas sensibles aux phases des harmoniques.

notes-frequences-cos-adsr.html

Comme précédemment ce programme permet de caractériser le signal en rentrant les composantes de la série de Fourier associée. La valeur par défaut est 1.0, cela correspond à une sinusoïde simple.

Il permet aussi de rentrer 4 nombres correspondant à l'ADSR (Attack, Decay, Sustain, Release) du son ainsi qu'un 5^e nombre représentant le niveau de l'attaque.

Les 5 nombres sont, dans l'ordre, la durée de l'attaque en secondes, le niveau de l'attaque (nombre inférieur à 1 que j'ai ajouté à l'ADSR), la durée de la décroissance toujours en secondes, le niveau du soutien et la durée du relâchement. Les valeurs par défaut sont "1 0.5 1 0.1 1".

On peut ensuite choisir le type de variation "linéaire" ou "exponentielle". La variation "exponentielle" semble plus naturelle car elle correspond à une perception logarithmique des niveaux. Si la variation est "exponentielle", le niveau de l'attaque ou du soutien devra être strictement positif, par exemple 0.0001, car $\log 0$ n'existe pas.

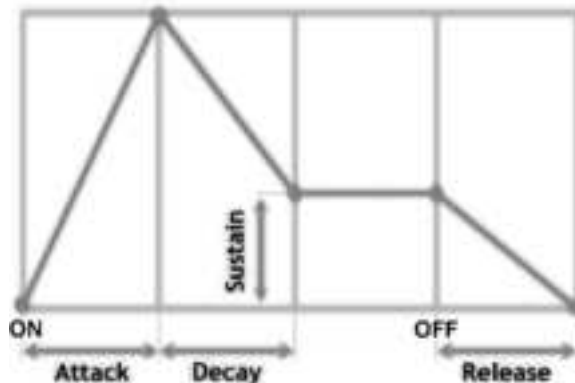


image Wikipédia

Le son démarre dès qu'on clique sur une touche du clavier. Après l'attaque et la décroissance, le niveau se stabilise au niveau du soutien. Pour arrêter le son il suffit de décocher la case au-dessus du clavier.

Si on appuie sur une autre touche, le nouveau son démarre pendant que le son précédent s'arrête suivant le temps consacré au relâchement. Ainsi, si on clique rapidement sur plusieurs touches, on pourra entendre une superposition de plusieurs sons. A l'inverse si on rentre la séquence AADSR "0 0 0 0.1 0" alors on aura un comportement analogue à celui du programme précédent.

notes-frequences-cos-adsr-courbe-fft.html

Ce programme propose en plus une visualisation du signal et de son spectre.

Pour les programmeurs :

Les deux derniers programmes utilisent un mini-buffer appelé "voltage" suivant une idée de Matt McKegg (<https://github.com/mmckegg/adsr>).

L'oscillateur est connecté à un "gainNode" qui permet le réglage du volume, "gainNode" étant lui-même connecté à "destination" pour que le son correspondant soit audible.

D'où le schéma :

osc -> gainNode -> destination

Par ailleurs, on crée un petit buffer de 2 éléments (2 à cause du navigateur Safari) appelé "voltage" qui est relié à "gainVoltage" relié lui même à gainNode.gain.

D'où le schéma :

voltage -> gainVoltage -> gainNode.gain

Par ailleurs, pour permettre la superposition de plusieurs sons, les oscillateurs et les dispositifs "gainNode", "voltage" et "gainVoltage" sont créés à chaque appui sur une touche du clavier, ils sont empilés puis dépilés lors du relâchement (appui sur une autre touche ou décochage de la case au-dessus du clavier).