Travaux pratiques d'acoustique, séances 1 et 2.
« Analyse de signaux sonores »

Licence 2, de physique et musicologie
(version : 11 janvier 2025)

Frédéric Faure
Université Grenoble Alpes, France
frederic.faure@univ-grenoble-alpes.fr

Abstract

Ces **travaux pratiques** concernent **l'analyse des signaux sonores** et sont destinés aux **étudiants de musicologie et de physique**. L'objectif est de montrer l'intérêt de l'analyse de Fourier pour étudier notre perception des sons et aussi la génération de sons par des instruments de musique. Ces T.P. sont directement reliés au chapitre 2 de ce cours cours d'acoustique musicale dont voici le fichier pdf, où l'on peut trouver des explications et solutions. Voici la page web du TP avec d'autres informations.

Contents

Remark 1. Dans la version électronique de ce document (pdf), vous accédez aux pages de cours et autres documents en **cliquant sur les liens de couleur qui entourent le texte**.

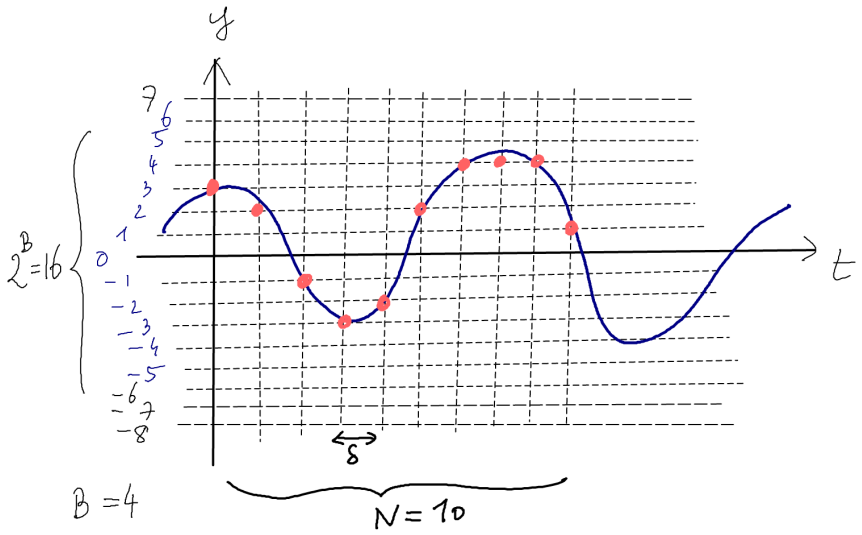
1 Définitions d'un signal et échantillonnage

Definition 1. Un **signal** u **échantillonné** de taille N codé sur B bits (en général $B \in \{8, 16, 24, 32, 64\}$) est:

$$u = (u_0, u_1, u_2, \dots u_{N-1}) \,,$$

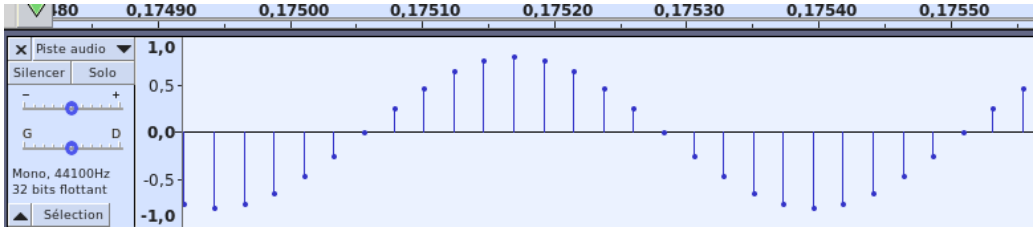
- où pour chaque indice $j \in [\![0, N-1]\!]$, la valeur $u_j \in [\![-2^{B-1}, 2^{B-1}-1]\!]$ est un entier codé avec un nombre B de bits en base 2. Ce codage appelé Complément à deux.
- Les dates t_j de chaque échantillon $u_j = u(t_j)$ sont espacées d'une durée $\delta > 0$ appelé **période d'échantillonnage**, par exemple $\delta = \frac{1}{44100} s$, (ainsi la fréquence d'échantillonnage $f_e = \frac{1}{\delta} = 44100$Hz est supérieure à 20000Hz qui est la limite supérieure des fréquences humainement audibles). Ainsi

$$t_j = j\delta, \quad j \in [\![0, N-1]\!] \,,$$

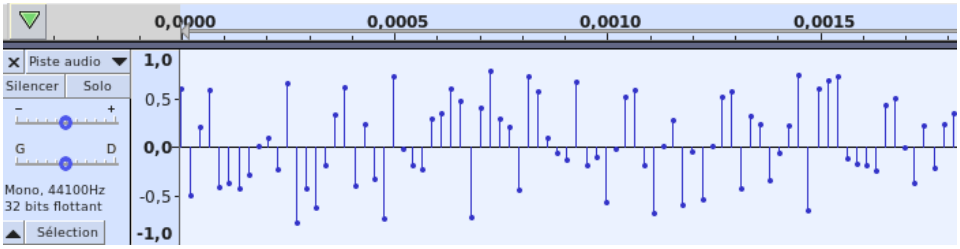


Exercise 1. (TP) **"Première utilisation de Audacity"**. Avec le logiciel **audacity**,

1. Créer un signal échantillonné de **forme sinusoidale** en faisant:
   — En bas à gauche, choisir "Taux du projet": 44100Hz.
   — Menu:Générer/Tonalité/Sinusoide, fréquence 2205Hz, amplitude 0.8, durée 0.5 sec.
   — Observer le signal (zoomer avec la souris). Voir figure ci-dessous.
   — Ecouter.
   — Calculer le nombre d'échantillons par période et vérifier.

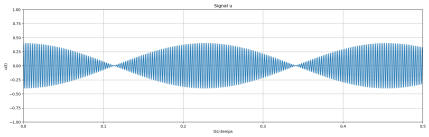


2. Créer un signal où chaque valeur $u_j \in [-1, 1]$ est choisi **au hasard** indépendamment des autres. Ce signal est appelé "**bruit blanc**". Pour cela Menu:Générer/Tonalité/Bruit/Blanc. écouter.



Exercise 2. (TP) **"Battements avec Audacity"**.

1. Avec le logiciel audacity, avec la commande "Pistes/Ajouter nouvelle piste mono" et "Générer/-tonalité", créer sur deux pistes différentes un signal de fréquence $f = 440$ Hz et $f_2 = 444$ Hz. Et faire la somme sur une nouvelle piste à l'aide de la commande "Pistes/Mix/rendu vers nouvelle piste". On obtient la superposition de deux signaux sinusoïdaux de fréquence $f = 440$ Hz et $f_2 = 444$ Hz (on verra que c'est une différence de 1/6 de demi-ton, à la limite du perceptible.



2. Vérifier que que l'on entend des **battements** dont la fréquence est

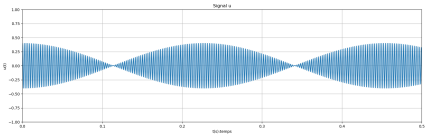
$$f_{\text{batt}} = |f_2 - f_1| = 4\,\text{Hz} \tag{1}$$

(i.e. on entends 4 battements par seconde).

3. Donner une explication simple de cette formule ou démontrer la formule en utilisant une formule de trigonométrie que l'on démontrera au préalable à partir de la définition $e^{i\theta} = \cos\theta + i\sin\theta$.

Exercise 3. (TP) **Optionnel "Battements avec Python"**.

1. Le programme python suivant, sinus_wav_file.py, qui génère un signal sinusoïdal de fréquence $f = 440$Hz et durée $D = 2$sec., défini par:
$$u_j = 0.2\cos(2\pi f t_j)$$
avec des instants $t_j = j\delta$, $j = 0 \to N-1$, avec $N = \left[\frac{D}{\delta}\right]$ données et le pas de temps $\delta = 1/44100$Hz. Ensuite le programme écrit ce signal dans un fichier sinus.wav au format wav. Executer le programme. Ouvrir le fichier sinus.wav avec audacity et écouter.
2. Modifier le programme pour générer le signal suivant
$$u(t) = 0.2\cos((2\pi)3ft) + 0.1\cos((2\pi)4ft)$$
et l'écouter et visualiser avec audacity. Qu'entendez vous? le signal est-il périodique?
3. Modifier le programme pour générer le signal suivant
$$u_1(t) = 0.2\cos(2\pi ft) + 0.2\cos((2\pi)(f+\epsilon)t) \quad (2)$$
avec $\epsilon = 4$Hz.



4. Essayer en stéréo: générer les signaux $u_1(t) = 0.2\cos(2\pi ft)$ et $u_2(t) = 0.2\cos((2\pi)(f+\epsilon)t)$ sur deux canaux différents [1]. Entendez vous des battements?

Exercise 4. (TP) **Battements avec des diapasons.**

1. Prendre deux diapasons de fréquence $f_1 = 440Hz$, $f_2 = (440+\epsilon)Hz$ où ϵ se règle avec la molette (diapason de droite sur la figure, ϵ est petit si la molette est en bas).
   (a) Régler ϵ pour avoir des battements de période 2sec. Ecouter la fréquence individuelle de chaque diapason. Entendez vous la différence?
   (b) Inversement régler ϵ pour entendre une différence de fréquence de 1/4 de ton. Quelle est la fréquence des battements?
2. Dans le cas de battements de période 2sec, où placer deux auditeurs A,B (ou deux micros) pour entendre des battements en opposition?.

1. Code python pour transformer deux signaux monos en un signal stéréo:

```
#-- examples of formula for two signals
L_u1 = 0.2* np.cos(2.0 * np.pi * f0 * L_t ) # signal 1
L_u2 = 0.2* np.cos(2.0 * np.pi * 1.01 * f0 * L_t ) #signal 2
# A 2D array where the left and right tones are contained in their respective rows
L_us = np.vstack((L_u1, L_u2)).transpose()
#---- write wavefile using the module wavfile.  The maximale amplitude is 1
wavfile.write('battement_stereo.wav', F, L_us)
```

Exercices à faire:

— Ex: 2.1.5, Ex.2.1.8, Ex. 2.1.14, Ex. 2.1.16 du cours d'acoustique,

2 Sonogramme, transformée par ondelette, transformée de Fourier

2.1 Signaux élémentaires: notes de musique, paquets d'ondes Gaussiens (ou ondelettes)

Une fonction parfaitement périodique comme $t \to \cos(2\pi f t)$ n'est pas réaliste et pas pratique non plus. Nous allons voir qu'il est beaucoup plus intéressant de considérer des fonctions appelées **paquet d'ondes Gaussien** (ou **ondelette** en théorie du signal) et que l'on appelle **note de musique** d'un point de vue musical, définie ainsi.

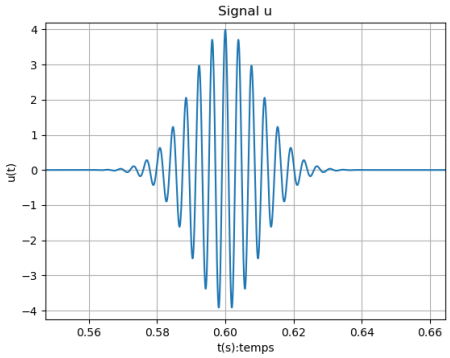
Definition 2. Un **paquet d'onde Gaussien** à l'instant $t_0 \in \mathbb{R}$ et de fréquence $f_0 \in \mathbb{R}$ de durée $\sigma > 0$ est

$$\varphi_{t_0,f_0,\sigma}(t) = e^{i2\pi f_0 t} e^{-\frac{1}{2}\left(\frac{t-t_0}{\sigma}\right)^2}. \tag{3}$$

Remark 2. Se rappeler que $\mathrm{Re}\left(e^{i\theta}\right) = \cos\theta$ et $\mathrm{Im}\left(e^{i\theta}\right) = \sin\theta$ donc la partie réelle du signal (3) est

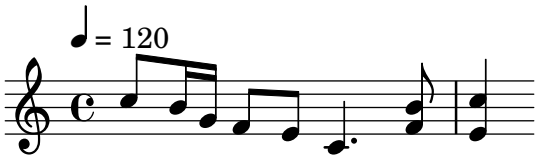
$$\mathrm{Re}\left(\varphi_{t_0,f_0}(t)\right) = \cos(2\pi f_0 t)\, e^{-\frac{1}{2}\left(\frac{t-t_0}{\sigma}\right)^2}$$

représenté ci-dessous dans le cas $t_0 = 0.6$, $f_0 = 260$ et $\sigma = 0.01$. La partie imaginaire est semblable.



Le premier facteur $e^{i2\pi f_0 t}$ de (3) s'appelle la **partie oscillante** et le deuxième facteur $e^{-\frac{1}{2}\left(\frac{t-t_0}{\sigma}\right)^2}$ l'**enveloppe** ou la **fenêtre Gaussienne**. Il peut y avoir d'autres choix intéressants de fenêtres ou window en mathématique et théorie du signal.

Exercise 5. (TP) **"Une mélodie"**. A partir de ce programme python wave_packet.py, créer un signal formé de la superposition de paquets d'ondes Gaussiens (ou notes de musique) de durée $\sigma = 0.02$s reproduisant la mélodie suivante. Visualiser le signal et écouter.



Exercices à faire:

— Ex: 2.2.9, Ex: 2.3.5, Ex: 2.3.26, Ex: 2.3.28, Ex: 2.3.30, Ex: 2.3.32 du cours d'acoustique,

2.2 Sonogramme, transformée de Fourier fenétrée ou transformée par ondelette

On utilise les paquets d'ondes (ou notes musicales élémentaires) (3) et le produit scalaire pour détecter la présence de chaque note $\varphi_{t,f}$ dans un signal u donné.

Definition 3. Si $u : t \to u(t)$ est un signal, sa **transformée par paquets d'ondes** (ou sonogramme) est la fonction complexe $(Tu)(t,f)$ sur le plan temps-fréquence suivante

$$(Tu)(t,f) := \frac{1}{\|\varphi_{t,f}\|}\langle\varphi_{t,f}|u\rangle \in \mathbb{C} \tag{4}$$

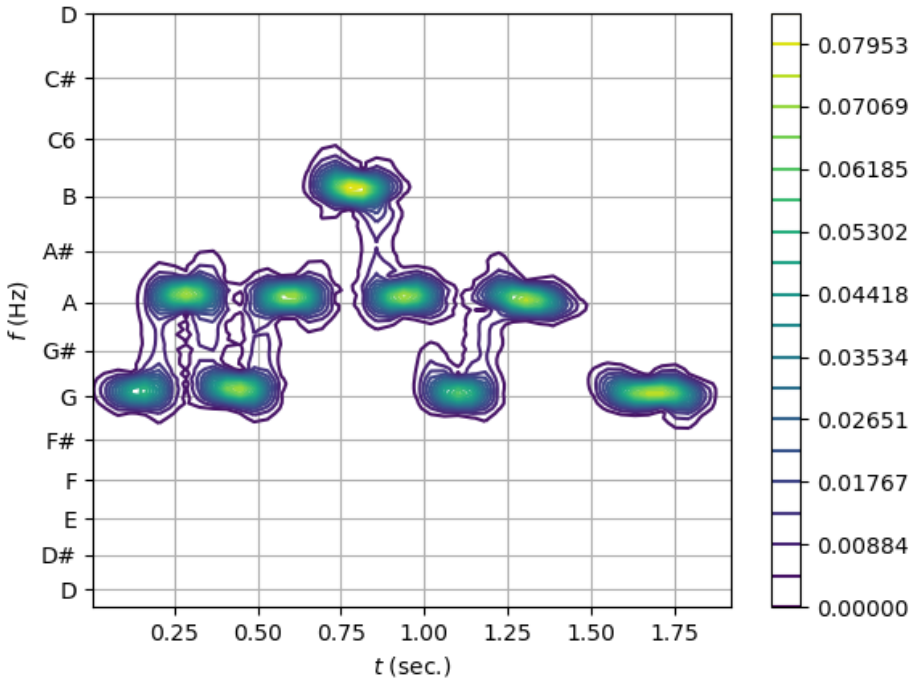
où

$$\langle\varphi_{t,f}|u\rangle = \int \overline{\varphi_{t,f}(t')}u(t')\,dt'.$$

Ainsi l'amplitude $|(Tu)(t,f)|$"mesure la présence" du paquet d'onde élémentaire (ou note) $\varphi_{t,f}$ dans le signal u.

Exercise 6. (TP) **"Sonograme d'un signal"**.

1. Ecouter ce signal sonore flute.wav. Utiliser ce programme python sonogram.py pour effectuer la transformée par paquets d'ondes (ou sonogramme) de ce signal. Choisir une durée $\sigma = 0.04$sec.



2. **Optionnel** Sachant que $\Delta t = \sigma$ est l'incertitude en temps et $\Delta f = \frac{1}{\sigma}$ l'incertitude en fréquence du paquet d'onde, quel intervalle acceptable a t-on pour σ pour bien distinguer les notes? Changer le paramètre σ dans le programme et observer les résultats si σ est trop grand ou trop petit?
3. Même exercice avec un fichier de votre création (enregistrement) ou ce fichier trumpet.wav.
4. **Optionnel** Même exercice avec la fonction (10) périodique qui contient 20 harmoniques.
5. Essayer le logiciel Spectroid sous Android ou logiciel équivalent. Régler: paramètres audio Traitement taille FFT: 1024, affichage des fréquences: Linéaire

2.3 Transformée de Fourier

Dans la limite de durée $\sigma \to \infty$, le paquet d'onde $\varphi_{t,f}$ en (3) devient tout simplement la fonction suivante appelée **mode de Fourier** ou **onde plane** et ne dépend plus des paramètres t_0, σ mais que de la fréquence f:

$$\varphi_f(t) = e^{i2\pi ft}. \tag{5}$$

Remark 3. La propriété remarquable de cette fonction est quelle est vecteur propre de l'opérateur dérivée

$$\frac{d}{dt}\varphi_f = (i2\pi f)\,\varphi_f$$

Pour cette raison il est souhaitable de décomposer un signal ou une fonction dans la base des modes de Fourier. Mais on perd l'avantage de la localisation temporelle du paquet d'onde (3).

Definition 4. (Joseph Fourier 1822 à Grenoble) La transformée de Fourier d'un signal $u : t \to u(t)$ est la fonction

$$f \in \mathbb{R} \to (\mathcal{F}u)(f) := \langle \varphi_f | u \rangle = \int_{\mathbb{R}} e^{-i2\pi ft} u(t)\, dt \tag{6}$$

Remark 4. $(\mathcal{F}u)(f) = \langle \varphi_f | u \rangle$, ce produit scalaire mesure la présence du mode de Fourier φ_f dans le signal u.

Avec l'ordinateur on utilise plutot la transformée de Fourier discrète qui correspond à un signal échantilloné:

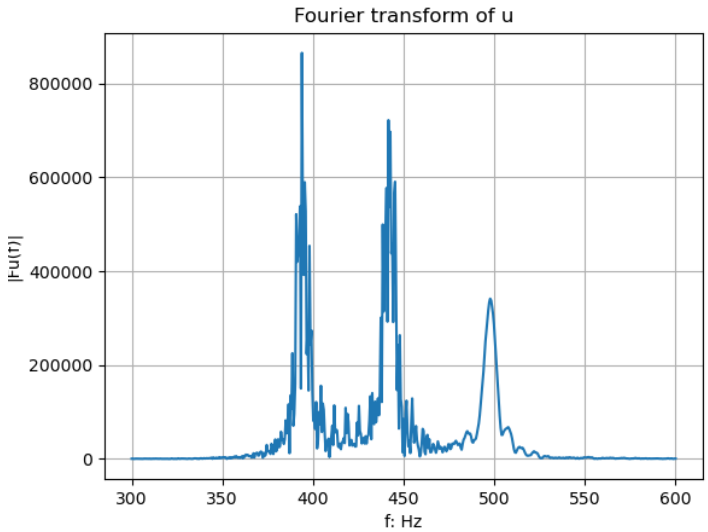
Theorem 1. *"**Transformée de Fourier discrète**, DFT". La transformée de Fourier d'un signal échantillonné de pas de temps δ et périodique de période T est un signal échantillonné de pas de fréquence $\frac{1}{T}$ et périodique en fréquence de période $\frac{1}{\delta}$.*
On a les formules suivantes pour un signal $u = (u_0, u_1, \dots u_{N-1})$. Les coefficients de Fourier sont

$$\hat{u}_n = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{-i2\pi jn/N} u_j \tag{7}$$

et inversement

$$u_j = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} e^{i2\pi jn/N} \hat{u}_n$$

Exercise 7. (TP) **"Transformée de Fourier discrète d'un signal"**. Voici un programme en python DFT_of_wavefile.py qui effectue la transformée de Fourier discrète du signal sonore flute.wav en appliquant directement la formule (7). On donne aussi le code qui utilise l'algorithme de la transformée de Fourier rapide (le résultat est le même, mais le calcul est plus rapide).



1. Executer ce programme et comparer le résultat avec le sonogramme de l'exercice 6. Commentez l'intérêt du sonogramme?
2. Enregistrer sa voix prononçant la voyelle "A" ou un instrument jouant une note tenue et observer la transformée de Fourier (discrète). Observer les harmoniques et les enveloppes du timbre, appelé formants.

3 Signaux périodiques, fréquences, notes musicales et pitch

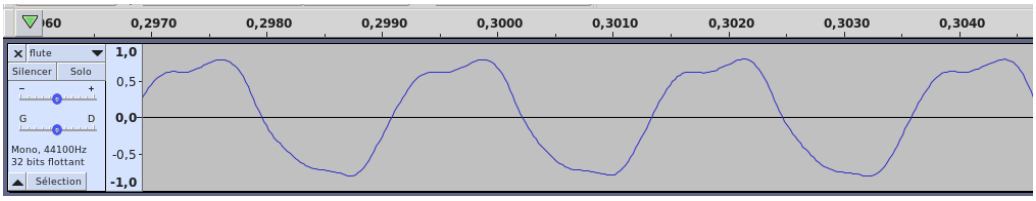
3.1 Signaux périodiques, séries de Fourier

Definition 5. Un **signal périodique** de **période** $T > 0$ est une fonction $t \in \mathbb{R} \to u(t) \in \mathbb{R}$ qui vérifie

$$u(t+T) = u(t), \forall t \in \mathbb{R}.$$

T est la **période fondamentale**, si c'est la valeur minimale vérifiant cela. On appelle $f = 1/T$ la **fréquence fondamentale** du signal périodique. Une **note musicale** est un signal périodique (ou presque en pratique).

Example 1. Avec Audacity, voici le signal d'une note de flûte de période approximative $T = 0.302 - 0.300 = 0.002$sec. donc de fréquence $f = 1/0.002 = 500$Hz:



Theorem 2. ***"Décomposition de Fourier".***
Un signal périodique réel de période $T = 1/f$ donnée s'écrit de façon unique

$$u(t) = \sum_{n \geq 0} A_n \cos(2\pi (nf) t + \varphi_n) \tag{8}$$

appelée ***décomposition de Fourier*** *ou* ***série de Fourier*** *du signal périodique, avec les amplitudes $A_n \geq 0$, des phases $\varphi_n \in [0, 2\pi]$ qui peuvent être arbitraires mais des fréquences $f_n = nf$ appelées les* ***harmoniques*** *du signal.*
Inversement A_n, φ_n s'obtiennent à partir du signal u par

$$A_n e^{i\varphi_n} = \frac{1}{T} \int_0^T e^{-i2\pi nt/T} u(t)\, dt, \quad n \in \mathbb{N}. \tag{9}$$

En particulier $\varphi_0 = 0$ et $A_0 = \frac{1}{T}\int_0^T u(t)\, dt$ est la moyenne du signal.

Exercise 8. (TP) **"Série de Fourier"**.

1. Ecrire un programme python (par exemple en modifiant sinus_wav_file.py) qui génère le fichier audio.wav du signal suivant sur une durée $D = 2$sec.

$$u(t) = 0.5 \sum_{n=1}^{M} \frac{1}{n} \sin(2\pi n f t) \tag{10}$$

   avec $f = 440$Hz et $M = 20$ harmoniques. Le signal est-il périodique? L'écouter avec Audacity. Essayer avec d'autres valeurs de M. Que pensez vous du signal si $M \to \infty$. Le signal sera t-il continu?

2. Au signal précédent rajouter des phases $\varphi_n \in [0, 2\pi]$ aléatoires (avec loi uniforme). Observer le signal. Ecouter le signal. (On peut montrer que dans la limite $M \to \infty$, le graphe est continu, c'est un mouvement Brownien en dim. 1). Essayer les phases $\varphi_n = n \log n$ (série de Hardy-Littlewood, [1, p.33])

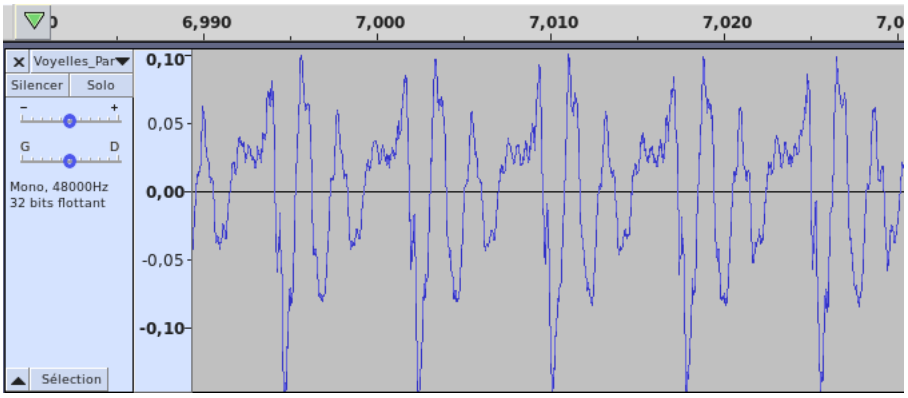
3. Essayer avec le signal

$$u(t) = 0.5 \sum_{n=1}^{N} \frac{1}{n} (1 - (-1)^n) \sin(2\pi n f t) \tag{11}$$

Le signal est-il périodique? que pensez vous du signal si $M \to \infty$? L'écouter avec Audacity. Ecrire $u(t)$ sous la forme (8) avec des amplitudes A_n et phases φ_n.

Exercise 9. (TP) "**Analyse de la voix**". Avec Audacity ou logiciel équivalent comme Lexis Audio Editor sous android.

1. Enregistrer sa voix prononçant une voyelle, parlée ou chantée. Ecouter. Observer que le signal est (localement) périodique, i.e. c'est une note de musique. Mesurer la période fondamentale T, déduire la fréquence f . Si besoin, voici un fichier déjà enregistré Voyelles_Par_Malik.wav.



2. Selectionner une période (wave-form) et la répéter avec audacity en faisant Ctrl-T (Edition/-suppresion spéciale/Rognage audio), puis Effets/Répéter. Quelle différence à l'écoute?
3. Inverser le sens du signal de départ (Effets/Inverser sens) et écouter. Essayer avec une phrase contenant des consonnes (attaques b,p,t etc) et sans consonnes mais voyelles seules (a,e,i,o,u).
4. Enregistrer sa voix prononçant une voyelle chuchotée. Ecouter. Observer le signal. Quelle différence observez vous avec le cas parlé?

References

[1] Yitzhak Katznelson. *An introduction to harmonic analysis*. Cambridge University Press, 2004.