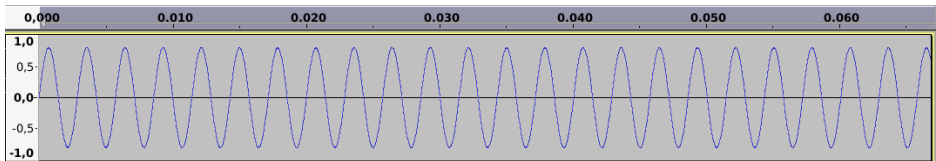


Groupe B

Exercice 1 — Intervalle de temps

Projet à 44 100... calculer l'intervalle de temps entre deux relevés.

Exercice 2 — Fréquence



Retrouver la fréquence, faire jouer dans GeoGebra et dans Audacity

Exercice 3 — Niveau sonore

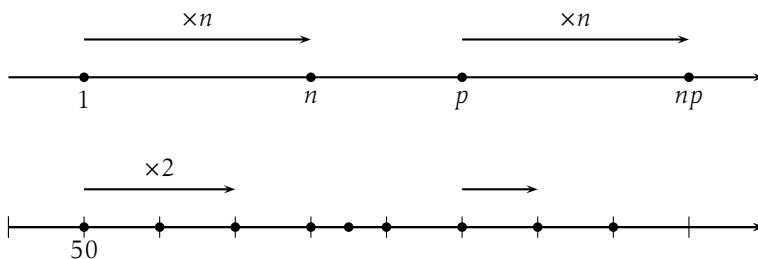
Trouver un tableau donnant des exemples de sons et leur niveau sonore en dB. Copier ce tableau dans votre support de cours en citant la source du document !

Dresser un tableau des niveaux sonores d'au moins 5 appareils utilisés au quotidien : aspirateur, réfrigérateur, machines à laver (linge, vaisselle...), tondeuses...

Exercice 4 — Échelles logarithmiques

Une graduation *logarithmique* permet de placer les très petites valeurs, comme les très grandes sur le même axe.

Pour placer le nombre $n \times p$ à partir des nombres n et p il faut procéder ainsi :

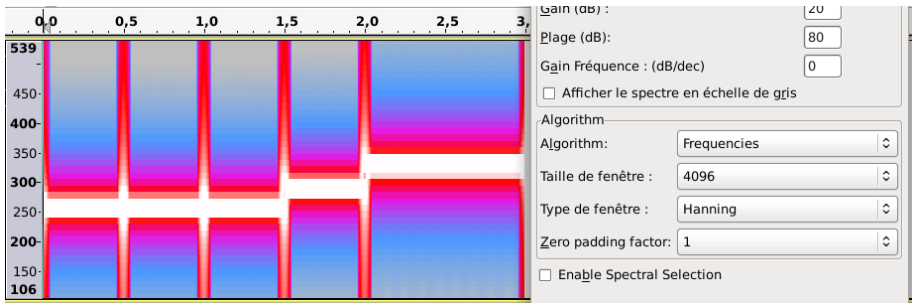


Exercice 5 — Faire chanter une fonction

Partie A – Une note

```
1 | pour obtenir SIMPLEMENT un morceau : écr
2 | nDo:=2^14*sin(2*pi*261*soundsec(0.5))
   | Done
3 | nRe:=2^14*sin(2*pi*293*soundsec(0.5))
   | Done
4 | nMi:=2^14*sin(2*pi*329*soundsec(1))
   | Done
5 | M:=concat(nDo,nDo,nDo,nRe,nMi)
   | Done
6 | playsnd(M)

1 | lire note à note ; les n comme note à cause des mots
2 | introduction d'une fonction Note(fréquence, durée)
3 | Note(f,d):=2^14*sin(2*pi*f*soundsec(d))
   | // Interprète Note
   | // Succès lors de la compilation Note
   | (f,d)>2^14*sin(2*pi*f*soundsec(d))
4 | nDo:=Note(261,0.5)
   | Done
5 | nRe:=Note(293,0.5)
   | Done
6 | nMi:=Note(329,1)
   | Done
7 | M:=concat(nDo,nDo,nDo,nRe,nMi)
   | Done
8 | playsnd(M)
   | Using sound parameters: channels, rate, bits, records
   | 1
9 | enregistrement en fichier .wav
10| writewav("xcas-clair3.wav",M)
   | 1
```



À l'aide du spectrogramme, retrouver la fréquence de chaque note et sa durée.

Partie B – Un accord - un ensemble de notes

```

7 Attenu(d,p):=concat(makelist(1,1,p*d*44100),makelist(0,1,(1-p)*d*44100))
// Interprète Attenu
// Succès lors de la compilation Attenu
(d, p )> concat(makelist(1,1,p*d*44100),makelist(0,1,(1-p)*d*44100))

8 NoteH(f,d):=214*(sin(2*pi*f*soundsec(d))+0.3*Attenu(d,.3).*sin(2*pi*3*f*
soundsec(d))+0.5*Attenu(d,0.6).*sin(2*pi*5*f*soundsec(d)))
// Interprète NoteH
Warning, une fonction algébrique définie par d'autres fonctions peut conduire à de
Vous voulez dire peut être NoteH:=unapply(214*(sin(2*pi*f*soundsec(d)))+(0.3*At
// Succès lors de la compilation NoteH
f, d )> 214*(sin(2*pi*f*soundsec(d))+0.3*(Attenu(d, 0.3 )).*sin(2*pi*3*f*so

9 nDo:=NoteH(261,0.5)
Done

10 nRe:=NoteH(293, 0.5)
Done

11 nMi:=NoteH(329,1)
Done

12 M:=concat(nDo,nDo,nDo,nRe,nMi)
Done

13 playsnd(M)
Using sound parameters: channels, rate, bits, records 1,44100,16,132300
1

14 enregistrement en fichier .wav

15 nom du fichier "xcas-clair4-coeff fond -coeff tierce-coeff quinte.wav"

16 writewav("xcas-clair4-1-03-05.wav",M)

```

La fonction de la ligne 8 est (sur une seule ligne !) :

$$\text{NoteH}(f, d) := 2^{14} * (\sin(2 * \pi * f * \text{soundsec}(d)) + 0.3 * \text{Attenu}(d, 0.3) * \sin(2 * \pi * 3 * f * \text{soundsec}(d)) + 0.5 * \text{Attenu}(d, 0.6) * \sin(2 * \pi * 5 * f * \text{soundsec}(d)))$$

À l'aide de Xcas, obtenir le fichier .wav ; puis à l'aide de Audacity :

- Obtenir le graphe de la fonction donnant l'intensité en fonction du temps ;
- Obtenir le spectrogramme de ce morceaux et retrouver les différentes fréquences des notes de l'accord et leur durée.