

# TP1 : Introduction au son

## 3A SRI

Julien Pinguier

Institut de Recherche en Informatique de Toulouse  
Université Paul Sabatier

2025-2026



# Présentation des TP

- 2 séances de TP de 2h
- Intervenant : Julien Pinquier (pinquier@irit.fr)
- Evaluation : questions sous Moodle

# Plan

## 1 Affichage et manipulation de la parole

- Introduction
- Charger un fichier audio
- Afficher un signal audio
- Spectre du signal
- Spectrogramme du signal
- Echelle MEL
- Analyse ceptrale

# Plan

## 1 Affichage et manipulation de la parole

- Introduction
- Charger un fichier audio
- Afficher un signal audio
- Spectre du signal
- Spectrogramme du signal
- Echelle MEL
- Analyse ceptrale

# Introduction

Nous allons aborder au cours de cette séance une introduction à la manipulation de signaux sonores. Les différentes manipulations présentées sur ce support vous permettront de faire le lien avec les notions que vous avez abordé en cours. L'utilisation de Python vous permettra d'être au plus près avec le signal et d'acquérir rapidement une autonomie au niveau de la manipulation et du traitement des signaux de parole. Ne vous contentez pas de suivre à la lettre les instructions présentes dans ce document, sachez en dégager le sens et n'hésitez pas à les utiliser en dehors des exemples indiqués. Pensez à visualiser et à comprendre les programmes Python (fichiers possédant le suffixe .py) fournis lors des manipulations.

## 1 Affichage et manipulation de la parole

- Introduction
- **Charger un fichier audio**
- Afficher un signal audio
- Spectre du signal
- Spectrogramme du signal
- Echelle MEL
- Analyse ceptrale

# Charger un fichier audio

```
import scipy.io.wavfile as wav  
fs, signal = wav.read('essai.wav')
```

- Décrire la variable `signal`.
- Quelle est la valeur de `fs` ?
- Afficher les 10 premières valeurs de `signal`.
- Quel est l'intervalle de variation des échantillons du signal ?

# Plan

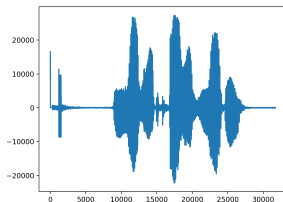
## 1 Affichage et manipulation de la parole

- Introduction
- Charger un fichier audio
- **Afficher un signal audio**
- Spectre du signal
- Spectrogramme du signal
- Echelle MEL
- Analyse ceptrale



# Afficher un signal audio

```
import matplotlib.pyplot as plt  
plt.figure(1)  
plt.plot(signal)  
plt.show()
```



- Calculer la durée du signal (en secondes).
- Afficher le signal avec l'axe des abscisses en secondes.

# Plan

## 1 Affichage et manipulation de la parole

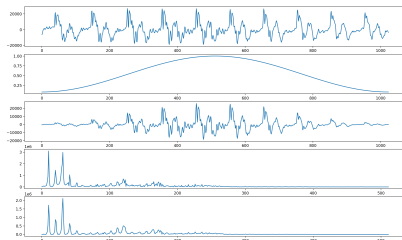
- Introduction
- Charger un fichier audio
- Afficher un signal audio
- **Spectre du signal**
- Spectrogramme du signal
- Echelle MEL
- Analyse ceptrale

# Signal et spectre

- Charger le fichier “note.wav”.
- Afficher le signal et son spectre (fonctions `abs` et `np.fft.fft`).
- Afficher l’abscisse du signal en seconde, et celle du spectre en Hertz.
- Quelle est la fréquence de cette note ? De quelle note s’agit-il ?
- Calculer la transformée de Fourier rapide d’un extrait de 1024 points du signal de parole (à partir de l’échantillon 11300).
- Afficher le résultat sur une demi-période, avec les abscisses en Hertz.

# Calcul du spectre et fenêtrage de Hamming

```
plt.subplot(5,1,1)
plt.plot(extrait)
plt.subplot(5,1,2)
ham1024 = np.hamming(len(extrait))
plt.plot(ham1024)
plt.subplot(5,1,3)
extrait_ham1024 =
np.multiply(extrait, ham1024)
plt.plot(extrait_ham1024)
plt.subplot(5,1,4)
spectre_f = abs(np.fft.fft(extrait))
plt.plot(spectre_f[:512])
plt.subplot(5,1,5)
spectre_f_ham1024 =
abs(np.fft.fft(extrait_ham1024))
plt.plot(spectre_f_ham1024[:512])
```



- Expliquer ces différents tracés.
- Qu'est-ce qu'une fenêtre de Hamming ? Quel est son intérêt ?
- Citez d'autres fenêtrages.

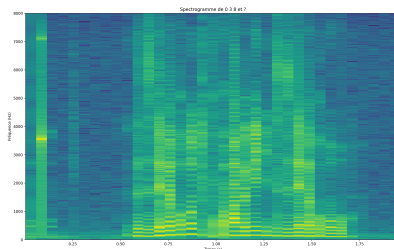
# Plan

## 1 Affichage et manipulation de la parole

- Introduction
- Charger un fichier audio
- Afficher un signal audio
- Spectre du signal
- **Spectrogramme du signal**
- Echelle MEL
- Analyse ceptrale

# Spectrogramme du signal

```
plt.specgram(signal, Fs=fs, window=ham1024, NFFT=1024)
plt.title('Spectrogramme de 0 3 8 et ?')
plt.ylabel('Fréquence (Hz)')
plt.xlabel('Temps (s)')
```



- Ce fichier est composé d'un bip, suivi de 4 chiffres : 0, 3, 8 et ?

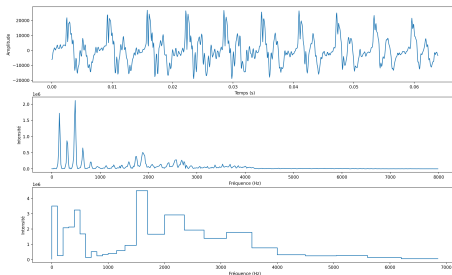
# Plan

## 1 Affichage et manipulation de la parole

- Introduction
- Charger un fichier audio
- Afficher un signal audio
- Spectre du signal
- Spectrogramme du signal
- Echelle MEL
- Analyse ceptrale

# Echelle MEL

```
import canaux as MEL
MEL.canaux(extrait, 16000, 26)
```



- Expliquer ces différents tracés.
- Qu'est-ce que l'échelle MEL ?



# Plan

## 1 Affichage et manipulation de la parole

- Introduction
- Charger un fichier audio
- Afficher un signal audio
- Spectre du signal
- Spectrogramme du signal
- Echelle MEL
- Analyse ceptrale

# Calcul du cepstre

- Réaliser le calcul du cepstre sur l'extrait.
- Afficher l'extrait, son spectre et son cepstre dans la même fenêtre.
- Les axes des abscisses doivent être en secondes ou en Hertz suivant le cas.
- Mettre un titre à chaque tracé et un nom aux axes.