

Tuto analyse fréquentielle de données sismiques avec Python sur JupyterLab (tuto_3_seismes.ipynb)

1) Importer les bibliothèques nécessaires

```
[1]: import obspy
from obspy import read

import numpy as np

import matplotlib
import matplotlib.pyplot as plt

from scipy.fft import fft, fftfreq
```

2) Importer les données des 3 séismes

Plus d'infos séisme du Teil: <https://www.emsc-csem.org/Earthquake/earthquake.php?id=804595>

Plus d'infos séisme à Chypre: <https://www.emsc-csem.org/Earthquake/earthquake.php?id=1085800>

Plus d'infos séisme aux Iles Sandwich: <https://www.emsc-csem.org/Earthquake/earthquake.php?id=1023153>

```
[2]: # Importer Les données des séismes: en france (Le Teil), à Chypre et aux Iles Sandwich
st_teil = read(r"votre_chemin\FR.CALF.00.HHZ_teil.sac")
print(st_teil)

st_chypre = read(r"votre_chemin\FR.CALF.00.HHZ_chypre.sac")
print(st_chypre)

st_sandwich = read(r"votre_chemin\FR.CALF.00.HHZ_sandwich.sac")
print(st_sandwich)
```

Shift + entrée

```
1 Trace(s) in Stream:
FR.CALF.00.HHZ | 2019-11-11T10:50:00.000500Z - 2019-11-11T10:59:59.990500Z | 100.0 Hz, 60000 samples
1 Trace(s) in Stream:
FR.CALF.00.HHZ | 2022-01-11T01:00:00.000501Z - 2022-01-11T01:49:59.990501Z | 100.0 Hz, 300000 samples
1 Trace(s) in Stream:
FR.CALF.00.HHZ | 2021-08-12T18:10:00.000501Z - 2021-08-12T21:29:59.990501Z | 100.0 Hz, 1200000 samples
```

3) Premier aperçu des données

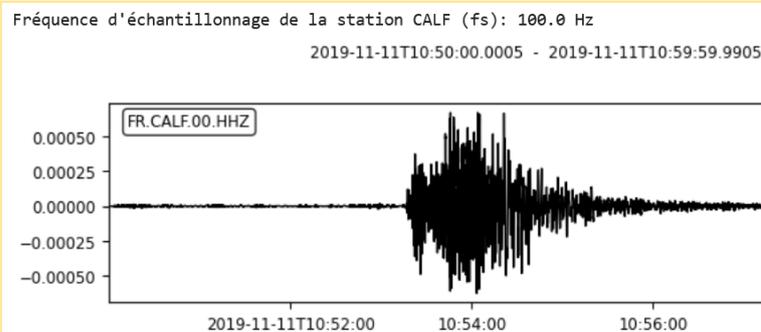
```
[3]: # Informations sur Les données:
print("Fréquence d'échantillonnage de la station CALF (fs): " + str(st_teil[0].stats.sampling_rate) + " Hz")

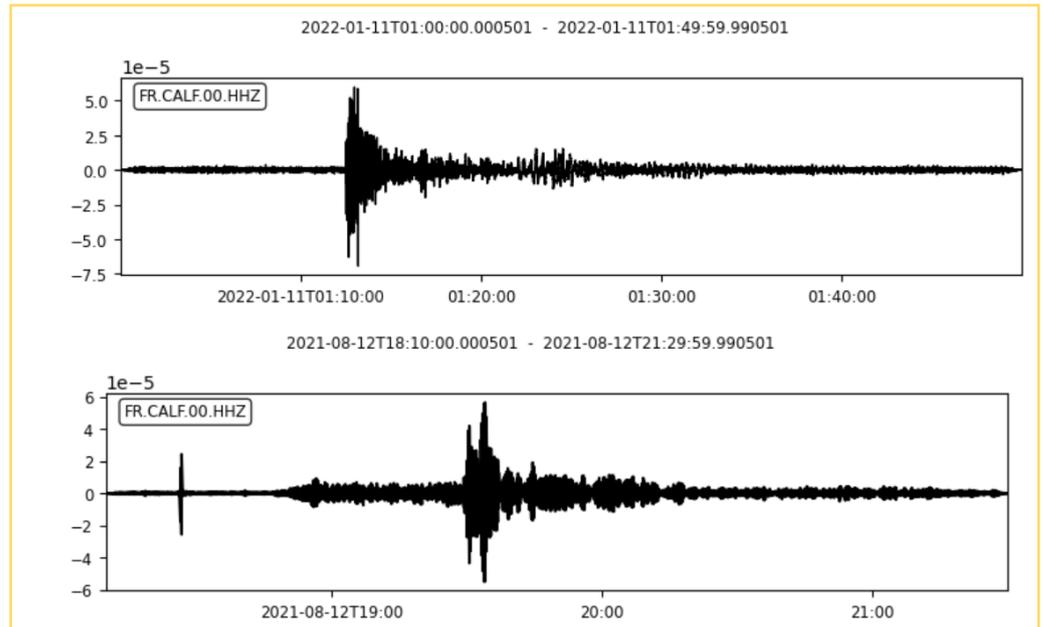
# Afficher Le contenu des données:
st_teil[0].plot()
plt.show()

st_chypre[0].plot()
plt.show()

st_sandwich[0].plot()
plt.show()
```

Shift + entrée





4) Pré-processing

detrend (polynomial): enlever la tendance du signal, ici de type polynomiale d'ordre 3

detrend (demean): enlever la moyenne du signal

taper: application d'un taper (atténuation des amplitudes aux 2 extrémités sur signal): peut fausser l'analyse fréquentielle

filter: filtrage des très basses fréquences du signal: élimination des fréquences d'origine naturelles (longues et petites oscillations)

```
[4]: # Pré-traitement des données + affichage:
# Teil
st_teil_corr = st_teil.copy()
st_teil_corr.detrend(type='polynomial', order=3)
st_teil_corr.detrend('demean')
st_teil_corr.taper(max_percentage=0.05, type="hann")
st_teil_corr.filter('highpass', freq=0.05)

st_teil_corr[0].plot()
plt.show()

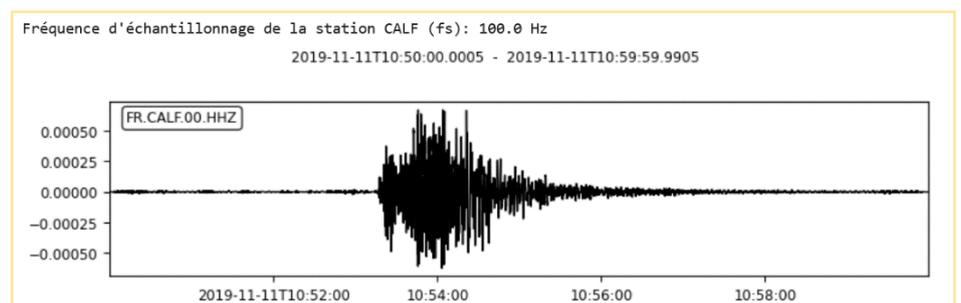
# Chypre
st_chypre_corr = st_chypre.copy()
st_chypre_corr.detrend(type='polynomial', order=3)
st_chypre_corr.detrend('demean')
st_chypre_corr.taper(max_percentage=0.05, type="hann")
st_chypre_corr.filter('highpass', freq=0.05)

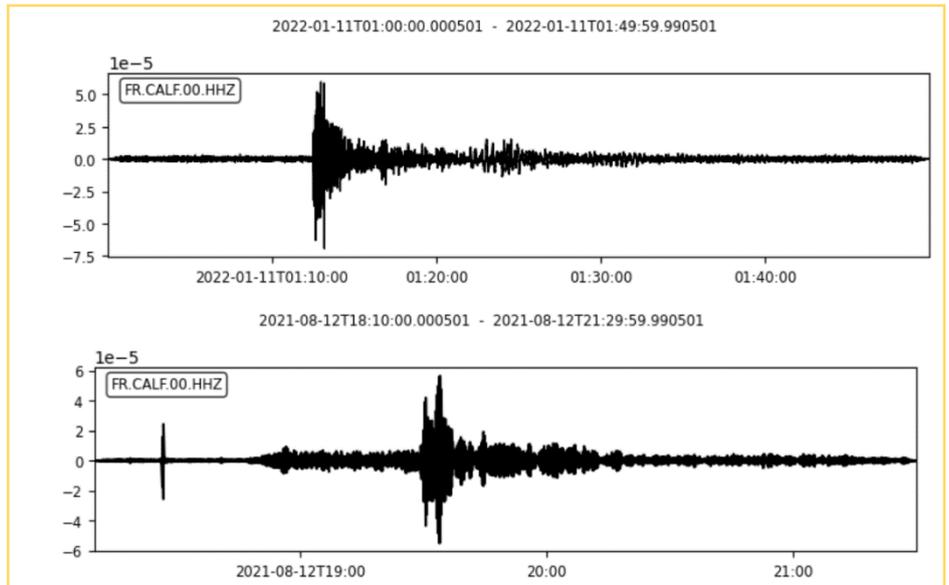
st_chypre_corr[0].plot()
plt.show()

# Iles Sandwich
st_sandwich_corr = st_sandwich.copy()
st_sandwich_corr.detrend(type='polynomial', order=3)
st_sandwich_corr.detrend('demean')
st_sandwich_corr.taper(max_percentage=0.05, type="hann")
st_sandwich_corr.filter('highpass', freq=0.05)

st_sandwich_corr[0].plot()
plt.show()
```


 Shift + entrée





5) Calcul du spectre de Fourier (fréquences d'intérêt sismologique entre 0.5 et 20 Hz)

```
[5]: # Calcul des spectres de Fourier:
fs = st_teil[0].stats.sampling_rate
dt = 1.0 / fs

# Nombre d'échantillons (de points) dans Les sismogrammes
N_teil = st_teil_corr[0].stats.npts
N_chypre = st_chypre_corr[0].stats.npts
N_sandwich = st_sandwich_corr[0].stats.npts

# Calcul de La transformée de Fourier (fast fourier transform)
FFT_teil = fft(st_teil_corr[0].data)
FFT_chypre = fft(st_chypre_corr[0].data)
FFT_sandwich = fft(st_sandwich_corr[0].data)

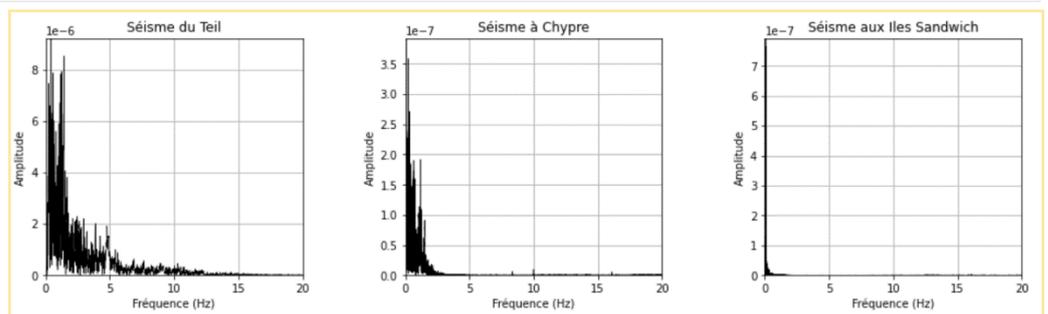
# On garde uniquement Les fréquences positives
# Valeurs en abscisses: xf
xf_teil = fftfreq(N_teil, dt)[:N_teil//2]
xf_chypre = fftfreq(N_chypre, dt)[:N_chypre//2]
xf_sandwich = fftfreq(N_sandwich, dt)[:N_sandwich//2]

# On prend La valeur absolue de L'amplitude uniquement pour Les fréquences positives et normalisation
# Valeurs en ordonnées: yf
yf_teil = np.abs(FFT_teil[0:N_teil//2]) * 2.0/N_teil
yf_chypre = np.abs(FFT_chypre[0:N_chypre//2]) * 2.0/N_chypre
yf_sandwich = np.abs(FFT_sandwich[0:N_sandwich//2]) * 2.0/N_sandwich

# Afficher Les spectres de Fourier des 3 séismes:
fig, axs = plt.subplots(1, 3, figsize=(16, 4))
plt.subplots_adjust(wspace=0.4)
axs[0].plot(xf_teil, yf_teil, linewidth=0.5, color='black')
axs[0].set_ylim(0, max(yf_teil))
axs[0].set_title("Séisme du Teil")
axs[1].plot(xf_chypre, yf_chypre, linewidth=0.5, color='black')
axs[1].set_ylim(0, max(yf_chypre))
axs[1].set_title("Séisme à Chypre")
axs[2].plot(xf_sandwich, yf_sandwich, linewidth=0.5, color='black')
axs[2].set_ylim(0, max(yf_sandwich))
axs[2].set_title("Séisme aux Iles Sandwich")

for num in range(3):
    axs[num].grid(True)
    axs[num].set_xlabel("Fréquence (Hz)")
    axs[num].set_ylabel("Amplitude")
    axs[num].set_xlim(0, 20)
    # Pour mieux analyser Le contenu fréquentiel (avec set_xlim):
    # -> essayez de recentrer Les graphiques entre [0-20] Hz, [0-5] Hz et [0-1] Hz
```

Shift + entrée



6) Figure spectrogramme

Paramètres du spectrogramme avec specgram:

- **NFFT**: nombre d'échantillons utilisés dans chaque bloc pour la FFT
- **Fs**: fréquence d'échantillonnage
- **window**: type de fenêtre utilisée (par défaut: hanning window) d'une longueur égale à celle du signal
- **noverlap**: nombre d'échantillons similaires repris pour le calculs de la FFT de bloc à bloc
- **cmap**: code couleur du spectrogramme
- **scale_by_freq**: donne les valeurs de densité (couleurs) en Hz^{-1}
- **vmin et vmax**: valeurs minimales et maximales de l'amplitude du spectre (intensité min et max des couleurs)

```
[6]: # Spectrogrammes
nfft = 400
VMIN = -190
VMAX = -70
ylim = 20

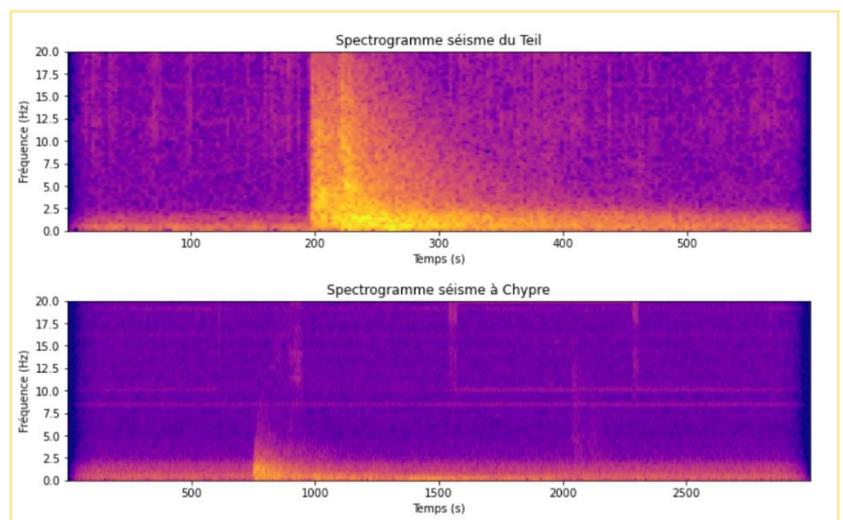
# Séisme du Teil
plt.figure(figsize=(12, 3))
plt.specgram(st_teil_corr[0], NFFT=nfft, Fs=fs, window=None, noverlap=None, cmap='plasma', scale_by_freq=True, vmin=VMIN, vmax=VMAX)
plt.ylim(0, ylim)
plt.margins(0,0)
plt.xlabel('Temps (s)')
plt.ylabel('Fréquence (Hz)')
plt.title('Spectrogramme séisme du Teil')
plt.show()

# Séisme à Chypre
plt.figure(figsize=(12, 3))
plt.specgram(st_chypre_corr[0], NFFT=nfft, Fs=fs, window=None, noverlap=None, cmap='plasma', scale_by_freq=True, vmin=VMIN, vmax=VMAX)
plt.ylim(0, ylim)
plt.margins(0,0)
plt.xlabel('Temps (s)')
plt.ylabel('Fréquence (Hz)')
plt.title('Spectrogramme séisme à Chypre')
plt.show()

# Séisme à Chypre
plt.figure(figsize=(12, 3))
plt.specgram(st_chypre_corr[0], NFFT=nfft, Fs=fs, window=None, noverlap=None, cmap='plasma', scale_by_freq=True, vmin=VMIN, vmax=VMAX)
plt.ylim(0, ylim)
plt.margins(0,0)
plt.xlabel('Temps (s)')
plt.ylabel('Fréquence (Hz)')
plt.title('Spectrogramme séisme à Chypre')
plt.show()

# Séisme aux Iles Sandwich
plt.figure(figsize=(12, 3))
plt.specgram(st_sandwich_corr[0], NFFT=nfft, Fs=fs, window=None, noverlap=None, cmap='plasma', scale_by_freq=True, vmin=VMIN, vmax=VMAX)
plt.ylim(0, ylim)
plt.margins(0,0)
plt.xlabel('Temps (s)')
plt.ylabel('Fréquence (Hz)')
plt.title('Spectrogramme séisme aux Iles Sandwich')
plt.show()
```

Shift + entrée



7) Rendre visibles les ondes qui nous intéressent

Après avoir identifié la gamme de fréquences du séisme sur le spectrogramme, nous pouvons filtrer le signal pour rendre visible les différentes ondes (P, S, de surface, etc).

Différents filtres sont possibles :

- **filtre 'highpass' ou 'passe haut'**: permet de garder les fréquences supérieures à une fréquence donnée (seules les fréquences inférieures sont filtrées)
- **filtre 'lowpass' ou 'passe bas'**: permet de garder les fréquences inférieures à une fréquence donnée (seules les fréquences supérieures sont filtrées)
- **filtre 'bandpass' ou 'passe bande'**: permet de garder les fréquences se situant dans un intervalle donné (entre des fréquences minimale et maximale)

```
[7]: # Filtrer Le signal pour faire ressortir Les fréquences qui nous intéressent:
# 'highpass' avec freq = ?
# 'Lowpass' avec freq = ?
# 'bandpass' avec freqmin = ? et freqmax = ?

# Séisme du Teil
st_teil_filt = st_teil_corr.copy()
st_teil_filt.filter('lowpass', freq=7.5)

st_teil_filt[0].plot()
plt.show()

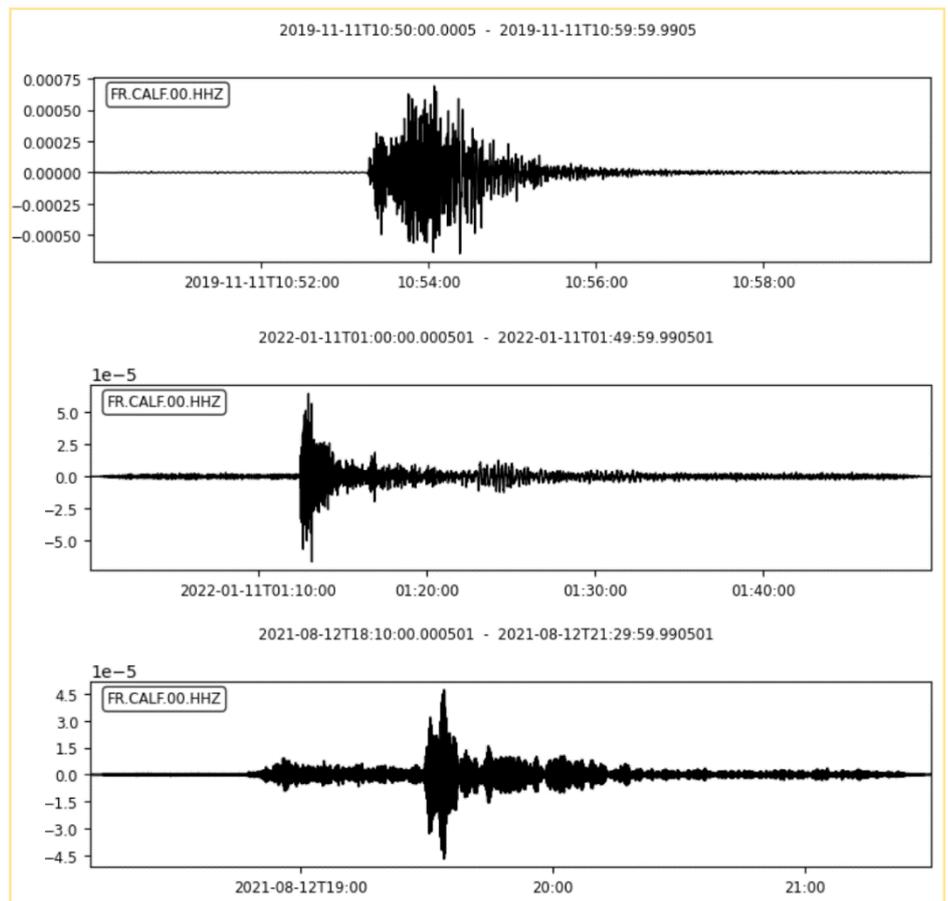
# Séisme à Chypre
st_chypre_filt = st_chypre_corr.copy()
st_chypre_filt.filter('lowpass', freq=2.5)

st_chypre_filt[0].plot()
plt.show()

# Séisme aux Iles Sandwich
st_sandwich_filt = st_sandwich_corr.copy()
st_sandwich_filt.filter('lowpass', freq=1)

st_sandwich_filt[0].plot()
plt.show()
```

↪
Shift + entrée



8) Figure groupée : sismogramme et spectrogramme

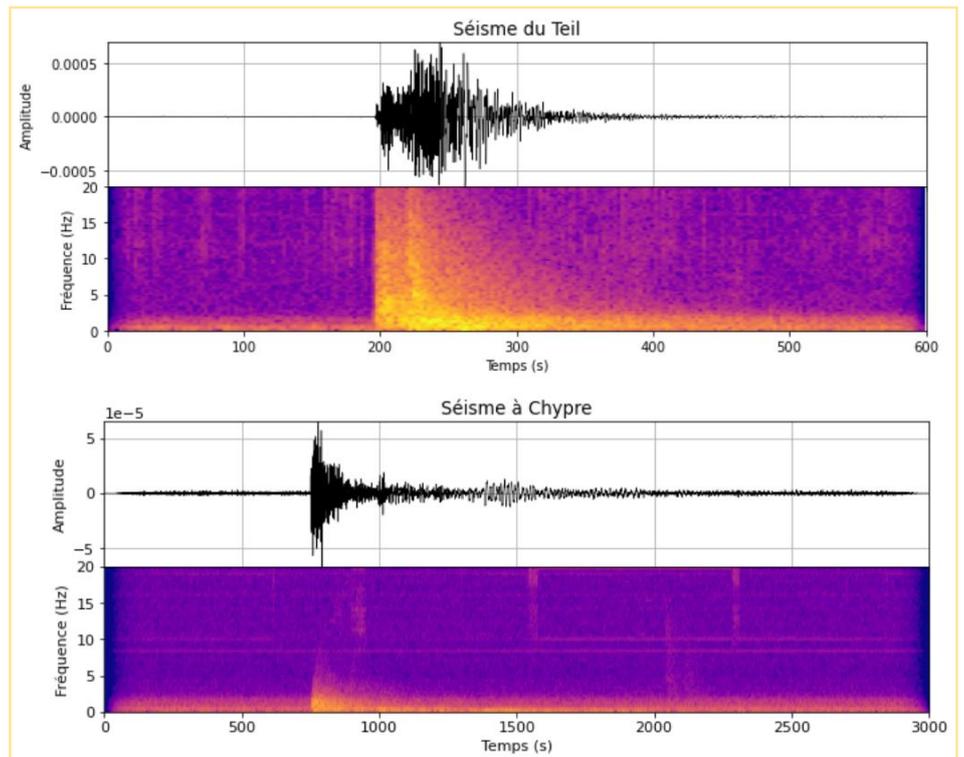
```
[8]: # Figures groupées: sismogramme et spectrogramme
# Vecteur de temps du signal
tt_teil = np.linspace(0, dt * N_teil, N_teil)
tt_chypre = np.linspace(0, dt * N_chypre, N_chypre)
tt_sandwich = np.linspace(0, dt * N_sandwich, N_sandwich)

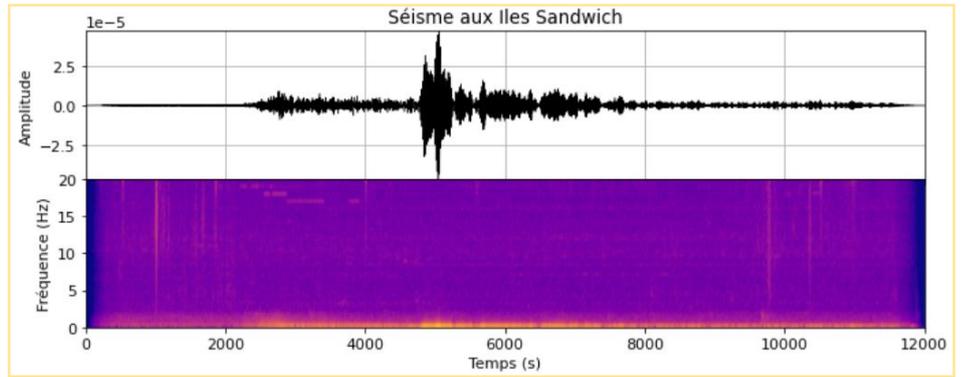
# Séisme du Teil
fig1, axes = plt.subplots(2, sharex=True, sharey=False, figsize=(11,4))
plt.subplots_adjust(wspace=0, hspace=0)
axes[0].plot(tt_teil, st_teil_filt[0].data, color='black', linewidth=0.5)
axes[0].grid()
axes[0].margins(0, 0)
axes[0].set_ylabel('Amplitude')
axes[1].specgram(st_teil_corr[0], NFFT=nfft, Fs=fs, window=None, noverlap=None, cmap='plasma', scale_by_freq=True, vmin=VMIN, vmax=VMAX)
axes[1].margins(0, 0)
axes[1].set_ylim(0, 20)
axes[1].set_xlabel('Temps (s)')
axes[1].set_ylabel('Fréquence (Hz)')
matplotlib.rcParams.update({'font.size': 11})
axes[0].set_title("Séisme du Teil")

# Séisme à Chypre
fig2, axes = plt.subplots(2, sharex=True, sharey=False, figsize=(11,4))
plt.subplots_adjust(wspace=0, hspace=0)
axes[0].plot(tt_chypre, st_chypre_filt[0].data, color='black', linewidth=0.5)
axes[0].grid()
axes[0].margins(0, 0)
axes[0].set_ylabel('Amplitude')
axes[1].specgram(st_chypre_corr[0], NFFT=nfft, Fs=fs, window=None, noverlap=None, cmap='plasma', scale_by_freq=True, vmin=VMIN, vmax=VMAX)
axes[1].margins(0, 0)
axes[1].set_ylim(0, 20)
axes[1].set_xlabel('Temps (s)')
axes[1].set_ylabel('Fréquence (Hz)')
matplotlib.rcParams.update({'font.size': 11})
axes[0].set_title("Séisme à Chypre")

# Séisme aux Iles Sandwich
fig3, axes = plt.subplots(2, sharex=True, sharey=False, figsize=(11,4))
plt.subplots_adjust(wspace=0, hspace=0)
axes[0].plot(tt_sandwich, st_sandwich_filt[0].data, color='black', linewidth=0.5)
axes[0].grid()
axes[0].margins(0, 0)
axes[0].set_ylabel('Amplitude')
axes[1].specgram(st_sandwich_corr[0], NFFT=nfft, Fs=fs, window=None, noverlap=None, cmap='plasma', scale_by_freq=True, vmin=VMIN, vmax=VMAX)
axes[1].margins(0, 0)
axes[1].set_ylim(0, 20)
axes[1].set_xlabel('Temps (s)')
axes[1].set_ylabel('Fréquence (Hz)')
matplotlib.rcParams.update({'font.size': 11})
axes[0].set_title("Séisme aux Iles Sandwich")
```

↙
Shift + entrée





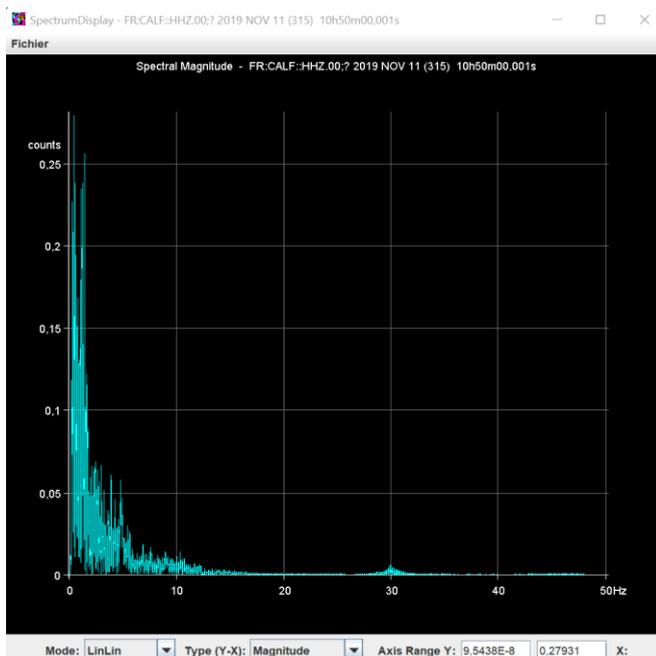
Tuto analyse fréquentielle de données sismiques avec SeisGram2K

Exemple d'analyse sur un fichier : séisme du Teil composante HHZ.

- Ouvrir SeisGram2K.
- Voici les boutons d'actions que vous avez à votre disposition :



- Importer le fichier contenant les données sismiques : **Fichier > Sélectionner Fichier**
- Pour filtrer le signal cliquez sur **Filtrer**, entrez les valeurs que vous voulez pour la fréquence minimum et la fréquence maximum puis cliquez sur **Appliquer**.
- Pour calculer le spectre de Fourier cliquez sur **Spectrum**, une nouvelle fenêtre s'ouvre.



Vous avez la possibilité de changer l'échelle de l'axe des abscisses avec le dérouleur **Mode** en bas à gauche de la fenêtre et vous pouvez aussi changer les limites des deux axes avec **Axes Range**.

- Pour calculer le spectrogramme, cliquez sur **Spectrogram**, une nouvelle fenêtre s'ouvre. Vous avez la possibilité de changer la couleur, l'intensité, le mode, la méthode et le type de graphique.

