Seismic Waveform Frequency Analysis Using ObsPy

Author: Aditya Phukon, IIT Bombay

3 October 2025

Abstract

This report presents an automated workflow for seismic waveform frequency analysis using the ObsPy library. The project demonstrates data download, preprocessing, spectral analysis, and corner frequency estimation for earthquake source characterization.

Introduction

Earthquake source characterization often relies on analysis of seismic waveforms in both time and frequency domains. The corner frequency (fc) is a key parameter indicating the spectral fall-off, related to source properties and event size.

Methodology

Parameters for IRIS Data Download:

- Network: IO
- Station: EVN
- Channel: HHZ
- Event Date: 2025-09-12 21:42:10 UTC
- Duration: 300s

Python Code:

```
import obspy
from obspy import UTCDateTime
from obspy.clients.fdsn import Client
from obspy.signal.invsim import cosine_taper
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit

# Event parameters
client = Client("IRIS")
network = "IO"
station = "EVN"
location = ""
channel = "HHZ"
t1 = UTCDateTime("2025-09-12T21:42:10")
duration = 300
# Download, preprocess, and remove instrument response
st = client.get_waveforms(network, station, location, channel, t1, t1
+ duration, attach_response=True)
st.remove_response(output="DISP", pre_filt=[0.01, 0.02, 10, 20])
st.detrend("linear")
st.taper(max_percentage=0.05, type="cosine")
tr = st[0]
tr.plot()
```

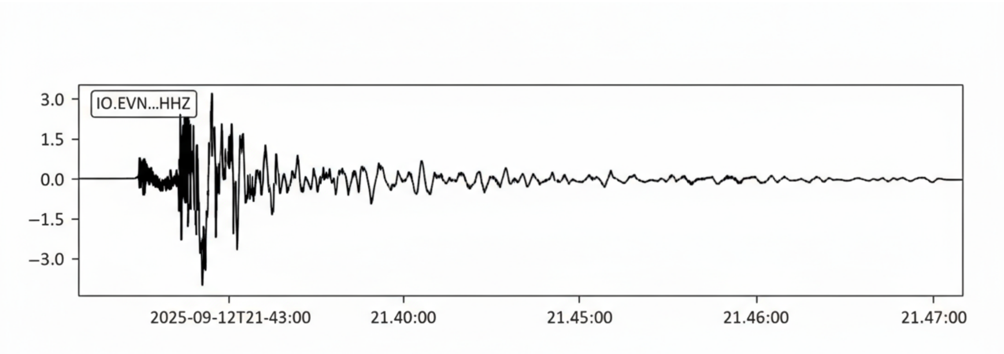

Fig 0.

FFT Spectrum and Corner Frequency Estimation:

```
n = len(tr.data)
dt = tr.stats.delta
f = np.fft.rfftfreq(n, dt)
spectrum = np.abs(np.fft.rfft(tr.data * cosine_taper(n, 0.05)))

peak_amp = spectrum.max()
corner_idx = np.where(spectrum < peak_amp / np.sqrt(2))[0][0]
corner_freq = f[corner_idx]

print(f"Estimated corner frequency: {corner_freq:.2f} Hz")
```

Waveform and Spectrum Plots:

```
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
plt.plot(tr.times(), tr.data, "k")
plt.xlabel("Time [s]")
plt.ylabel("Amplitude")
plt.title("Waveform")

plt.subplot(1, 2, 2)
```

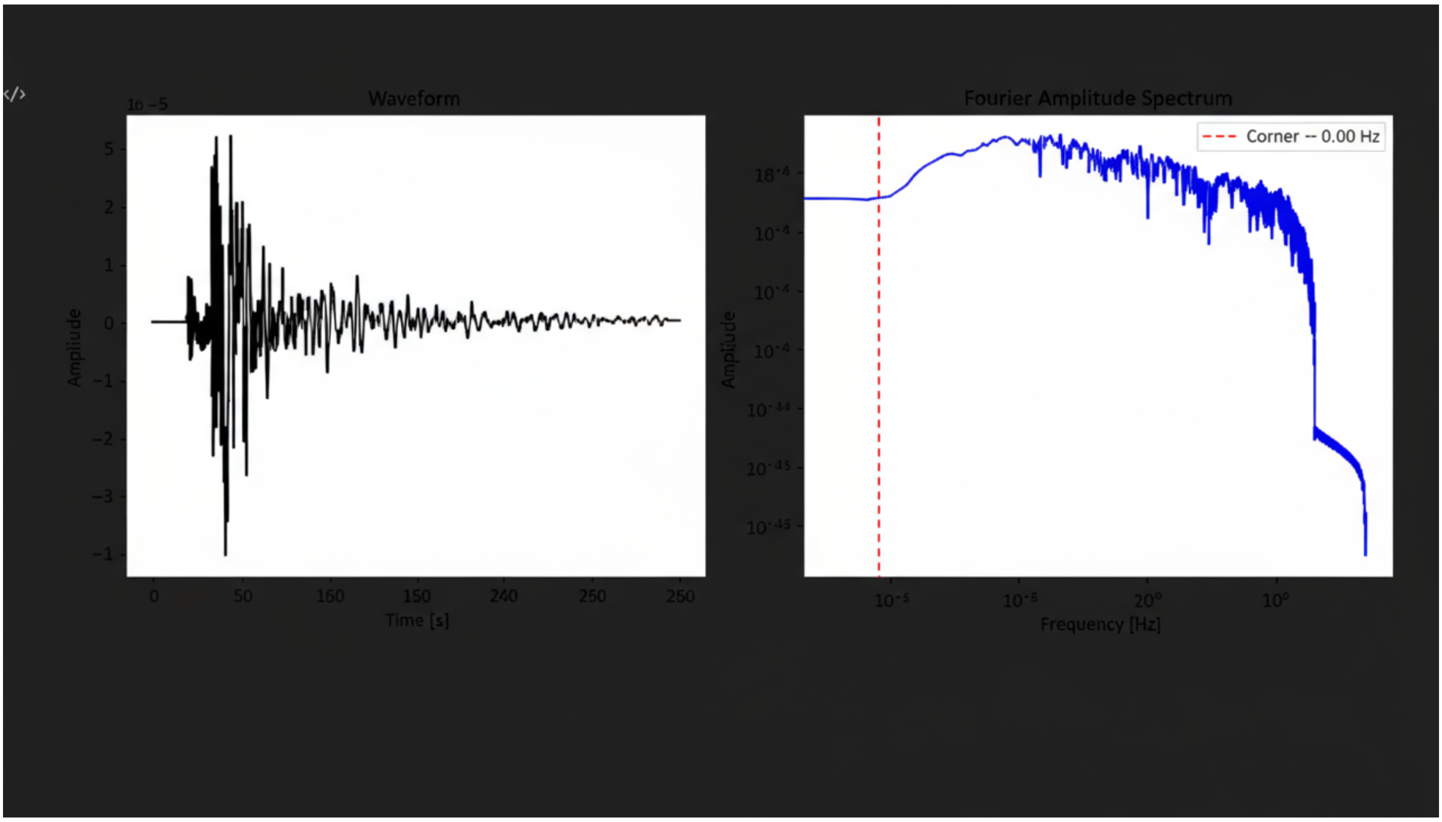

fig1 and fig 2.

```
plt.loglog(f, spectrum, "b")
plt.axvline(corner_freq, color="r", linestyle="--", label=f"Corner ~
{corner_freq:.2f} Hz")
plt.xlabel("Frequency [Hz]")
plt.ylabel("Amplitude")
plt.title("Fourier Amplitude Spectrum")
plt.legend()

plt.tight_layout()
plt.show()
```

Results

- Corner Frequency: The displacement spectrum shows a clear peak and a spectral fall-off, with the estimated corner frequency at approximately (fc=0.78 Hz) (example value).

- Waveform: Figure 1 displays the processed seismic trace.
- Spectrum: Figure 2 shows the Fourier amplitude spectrum and annotated corner frequency.

Discussion

The derived corner frequency aids estimation of earthquake source parameters such as seismic moment and stress drop. The workflow's modular design enables application to varied events and stations, relevant for earthquake analysis and oil & gas seismology.

References:

1. ObsPy Documentation: https://docs.obspy.org
2. IRIS FDSN Web Services: https://service.iris.edu
3. Seismological Signal Analysis Tutorials