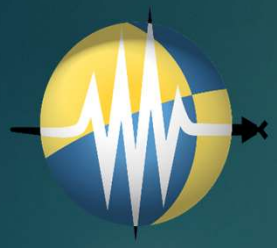


ObsPy

A Python Framework for Seismology

05-Filtering Seismograms

IMAN KAHBASI

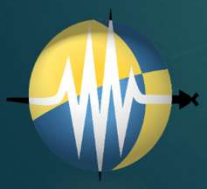
MASTERS STUDENT AT IIEES

1398/12/17

Method of stream and trace

>>> trace.filter(TypeOfFilter, freq or freqs, corners, zerophase=False)

>>> stream.filter(TypeOfFilter, freq or freqs, corners, zerophase=False)

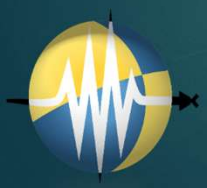


ObsPy

A Python Framework for Seismology

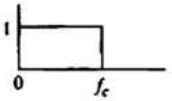
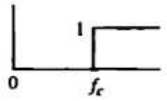
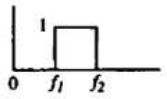
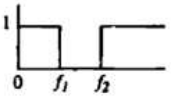
Import function


```
>>> data = trace.data
>>> df = trace.stats.sampling_rate
>>> from obspy.signal.filter import lowpass, highpass, bandpass, bandstop
>>> lowpass(data, freq, df, corners=4, zerophase=False)
>>> highpass(data, freq, df, corners=4, zerophase=False)
>>> bandpass(data, freqmin, freqmax, df, corners=4, zerophase=False)
>>> bandstop(data, freqmin, freqmax, df, corners=4, zerophase=False)
```

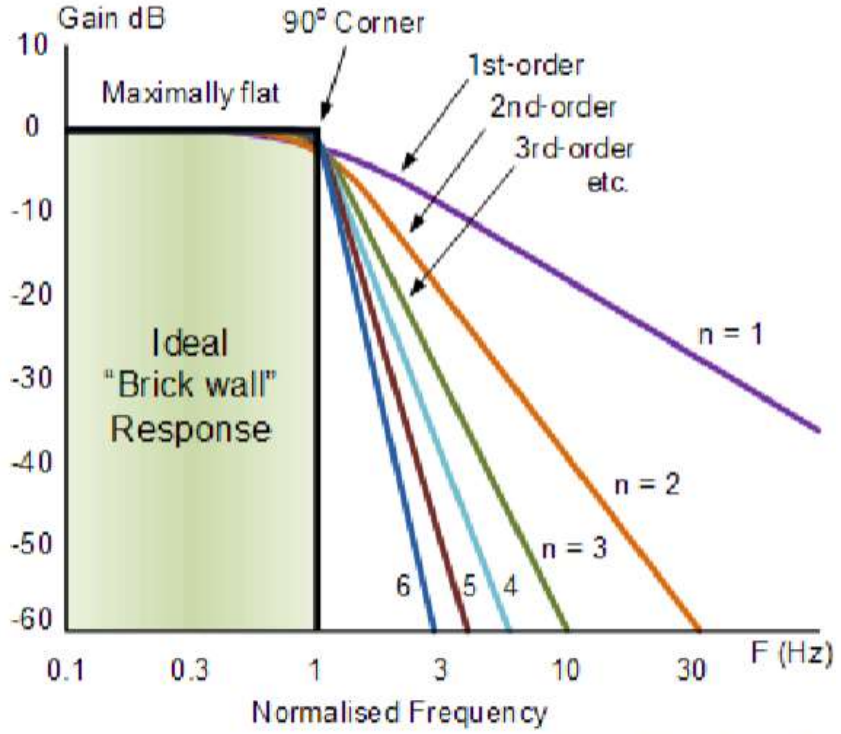


ObsPy

A Python Framework for Seismology

Type	Typical Ideal $\|H(f)\|$	Description	Example Uses
Lowpass		removes all frequency information above f_c	noise removal, interpolation, data smoothing
Highpass		removes all frequency information below f_c	removing DC or low freq drift, edge detection or enhancement
Bandpass		removes all frequency information outside of $f_1 \rightarrow f_2$	tuning in to one radio station, andio graphic equalizers
Notch		removes all frequency information between $f_1 \rightarrow f_2$	removing noise at a particular frequency, e.g. 60 Hz

Butterworth filter



The sampling theorem

- The shanon or nyquist sampling theorem indicates that a continuous signal can be properly sampled, only if it does not contain frequency components above one-half of the sampling rate.
- The key point to remember is that a digital signal cannot contain frequencies above one-half the sampling rate (i.e., the Nyquist frequency)