**yf = np.abs(fft(accel)[:N // 2])**

**What is happening here?**

fft(accel):

* This computes the **Fast Fourier Transform (FFT)** of your time-domain acceleration signal (accel).
* The FFT converts your signal from the time domain (acceleration vs. time) to the frequency domain (amplitude vs. frequency).
* The result is a complex-valued array of length N (where N is the number of samples).

[:N // 2]:

* The FFT of a real-valued signal is **symmetric** about the midpoint (Nyquist frequency).
* Only the first half (N // 2) contains the unique frequency information for real signals.
* So, you take only the first half to get the **one-sided spectrum**.

np.abs(...):

* The FFT output is complex (has real and imaginary parts).
* np.abs() computes the **magnitude** (length) of each complex number, which gives the amplitude of each frequency component.

**Summary:** This line gives you the amplitude (magnitude) of each frequency component, for frequencies from 0 Hz up to just below the Nyquist frequency (half the sampling rate).

**2. xf = fftfreq(N, 1 / FS)[:N // 2]**

**What is happening here?**

fftfreq(N, 1 / FS):

* This generates an array of frequency bins corresponding to the FFT output.
* N is the number of samples.
* 1 / FS is the time step between samples (sampling interval).
* The result is an array of length N with the frequency (in Hz) for each FFT bin.

[:N // 2]:

* Again, you only need the first half for the one-sided spectrum (unique frequencies for real signals).

**Summary:**This line gives you the actual frequency values (in Hz) that correspond to each amplitude in yf.

**How do these work together?**

* xf is the x-axis (frequency in Hz).
* yf is the y-axis (amplitude of each frequency component).
* When you plot plt.plot(xf, yf), you get a spectrum showing which frequencies are present in your original signal and how strong they are.

**Example**

Suppose:

* FS = 50000 (sampling frequency)
* N = 100000 (number of samples)

Then:

* xf will go from 0 Hz up to just below 25,000 Hz (the Nyquist frequency).
* yf will have the amplitude for each of those frequencies.

**Visualization**

* **xf**: [0 Hz, 0.5 Hz, 1 Hz, ..., 24,999.5 Hz] (if N is even)
* **yf**: [amplitude at 0 Hz, amplitude at 0.5 Hz, ..., amplitude at 24,999.5 Hz]

**In summary:**

* The first line computes the amplitude spectrum (one-sided) of your signal.
* The second line computes the corresponding frequency values for each amplitude.
* Together, they let you analyze and visualize the frequency content of your vibration data.