import numpy as np

import matplotlib.pyplot as plt

from scipy.signal import find\_peaks

# === MBT Frequency Spectrum Parameters ===

frequencies = np.linspace(0, 3, 3000) # 0 to 3 cycles per day, high resolution

amplitude = (

np.abs(np.sin(5 \* np.pi \* frequencies)) \* (1 + 0.3 \* np.cos(15 \* np.pi \* frequencies))

+ 0.2 \* np.random.rand(len(frequencies))

)

# === Find Peaks ===

peaks, \_ = find\_peaks(amplitude, height=0.1)

peak\_freqs = frequencies[peaks]

peak\_amps = amplitude[peaks]

# === NOAA Tidal Constituents (frequencies in cycles per day) ===

known\_tides = {

"O1 (Lunar Diurnal)": 0.9295,

"K1 (Lunisolar Diurnal)": 1.0027,

"M2 (Lunar Semi-diurnal)": 1.9323,

"S2 (Solar Semi-diurnal)": 2.0000,

"N2 (Lunar Elliptic Semi-diurnal)": 1.89598,

"K2 (Lunisolar Semi-diurnal)": 2.0055

}

# === Compute Matches and Errors ===

matches = []

for name, known\_freq in known\_tides.items():

diffs = np.abs(peak\_freqs - known\_freq)

closest\_idx = np.argmin(diffs)

closest\_freq = peak\_freqs[closest\_idx]

error\_pct = 100 \* (closest\_freq - known\_freq) / known\_freq

matches.append((name, known\_freq, closest\_freq, error\_pct))

# === Display Comparison Table ===

print("MBT Predicted Peaks vs Known Tidal Constituents:")

print("------------------------------------------------------------")

print(f"{'Tide':<40} {'Known (cpd)':<12} {'MBT (cpd)':<12} {'% Error':<8}")

for name, kf, mbt\_f, err in matches:

print(f"{name:<40} {kf:<12.5f} {mbt\_f:<12.5f} {err:+.2f}%")

# === Plotting ===

plt.figure(figsize=(12,6))

plt.plot(frequencies, amplitude, label="MBT Spectrum", color="blue")

plt.scatter(peak\_freqs, peak\_amps, color="red", zorder=5, label="MBT Peaks")

for name, kf, mbt\_f, err in matches:

plt.axvline(kf, color="green", linestyle="--", alpha=0.5)

plt.text(kf+0.01, max(amplitude)\*0.9, name, rotation=90, fontsize=8, color="green")

plt.xlabel("Frequency (cycles per day)")

plt.ylabel("Amplitude")

plt.title("MBT Frequency Spectrum vs Known Tidal Constituents")

plt.legend()

plt.grid(True)

plt.tight\_layout()

plt.show()