import numpy as np

import matplotlib.pyplot as plt

# Settings

T = 200 # total timesteps

collapse\_point = 100 # midpoint of collapse/revival

S\_quantum = 2.828 # Tsirelson's bound

S\_classical = 2.0 # Local realism limit

# Collapse: 0 (order) → 1 (full collapse), then healing 1 → 0

C = np.concatenate([

np.linspace(0, 1, collapse\_point), # Collapse

np.linspace(1, 0, T - collapse\_point) # Healing

])

# MBT memory law constants (tweak as you like)

P = 1.0

c\_echo = 1.0

# MBT memory law: M(t) = (1 - C(t)) \* P \* c\_echo^2

M = (1 - C) \* P \* c\_echo\*\*2

# Bell S (order parameter): S = S\_classical + (1 - C) \* (S\_quantum - S\_classical)

S\_MBT = S\_classical + (1 - C) \* (S\_quantum - S\_classical)

# Plot both on dual axes

fig, ax1 = plt.subplots(figsize=(8,5))

color1 = 'tab:blue'

color2 = 'tab:green'

ax1.plot(S\_MBT, label='Bell S (MBT)', color=color1, linewidth=2)

ax1.axhline(S\_classical, color='gray', linestyle='--', label='Classical Limit')

ax1.axhline(S\_quantum, color='purple', linestyle='--', label='Quantum Limit')

ax1.set\_xlabel("Time (collapse → healing)")

ax1.set\_ylabel("Bell S (Order Parameter)", color=color1)

ax1.tick\_params(axis='y', labelcolor=color1)

ax2 = ax1.twinx()

ax2.plot(M, label='MBT Memory', color=color2, linewidth=2, alpha=0.6)

ax2.set\_ylabel("MBT Memory", color=color2)

ax2.tick\_params(axis='y', labelcolor=color2)

# Legends

lines1, labels1 = ax1.get\_legend\_handles\_labels()

lines2, labels2 = ax2.get\_legend\_handles\_labels()

ax1.legend(lines1 + lines2, labels1 + labels2, loc='upper center')

plt.title("Emergence & Revival of Quantum Order and MBT Memory Law")

plt.tight\_layout()

plt.show()