import numpy as np

import matplotlib.pyplot as plt

# Create a synthetic CMB cold spot field

def generate\_cmb\_cold\_spot(grid\_size=256, cold\_center=(-57, 209), depth=1.0, spread=500):

    lat = np.linspace(-90, 90, grid\_size)

    lon = np.linspace(0, 360, grid\_size)

    Lon, Lat = np.meshgrid(lon, lat)

    r = (Lat - cold\_center[0])\*\*2 + (Lon - cold\_center[1])\*\*2

    cmb\_field = -depth \* np.exp(-r / spread)

    return cmb\_field, Lat, Lon

# Create a synthetic MBT curvature field with a bell root

def generate\_mbt\_curvature(grid\_size=256, root\_center=(-70, 180), height=1.0, steepness=300):

    lat = np.linspace(-90, 90, grid\_size)

    lon = np.linspace(0, 360, grid\_size)

    Lon, Lat = np.meshgrid(lon, lat)

    r = (Lat - root\_center[0])\*\*2 + (Lon - root\_center[1])\*\*2

    curvature\_field = height / (1 + steepness \* r / 10000)

    return curvature\_field, Lat, Lon

# Generate fields

cmb, Lat, Lon = generate\_cmb\_cold\_spot()

mbt, \_, \_ = generate\_mbt\_curvature()

# Plot side-by-side

plt.figure(figsize=(10, 4))

plt.subplot(1, 2, 1)

plt.contourf(Lon, Lat, cmb, levels=50, cmap='coolwarm')

plt.title("Synthetic CMB Cold Spot")

plt.xlabel("Longitude (°)")

plt.ylabel("Latitude (°)")

plt.grid(True)

plt.subplot(1, 2, 2)

plt.contourf(Lon, Lat, mbt, levels=50, cmap='plasma')

plt.title("MBT Bell Curvature Field")

plt.xlabel("Longitude (°)")

plt.ylabel("Latitude (°)")

plt.grid(True)

plt.tight\_layout()

plt.show()

I used that to get this