# class\_EDE.py - Simulator for axion-like Early Dark Energy model

# Computes H0, sound horizon, and late-time w based on phi’s parameters

import numpy as np

# Define input parameters

def set\_parameters():

params = {

'h': 0.67, # Baseline H0 = 67 km/s/Mpc

'omega\_b': 0.022, # Baryon density

'omega\_cdm': 0.12, # Cold dark matter density

'A\_s': 2.1e-9, # Scalar amplitude

'n\_s': 0.96, # Scalar spectral index

'tau\_reio': 0.054, # Reionization optical depth

'm\_phi': 1e-27, # Axion mass (eV)

'g\_phi': 1e-12, # Dark matter coupling (GeV^-1)

'phi\_ini': 0.1, # Initial field value (fraction of f)

'f\_EDE': 0.1, # EDE fraction at z=3000

'z\_EDE': 3000, # Peak redshift

}

return params

# Simulate EDE’s effect

def simulate\_EDE(params):

h = params['h']

z\_EDE = params['z\_EDE']

f\_EDE = params['f\_EDE']

rs\_base = 147.0 # ΛCDM sound horizon (Mpc)

rs\_EDE = rs\_base \* (1 - 0.05) # 5% reduction

h0\_EDE = h \* (rs\_base / rs\_EDE) \* (1 + f\_EDE / 10) # H0 adjustment

h0\_km\_s\_Mpc = h0\_EDE \* 100

w\_EDE = -0.98 # Late-time equation of state

return {

'H0': h0\_km\_s\_Mpc,

'H0\_error': 1.0,

'rs': rs\_EDE,

'rs\_base': rs\_base,

'w\_EDE': w\_EDE,

'w\_error': 0.02

}

# Main function

def main():

params = set\_parameters()

results = simulate\_EDE(params)

h0\_str = f"{results['H0']:.1f} ± {results['H0\_error']:.1f}"

w\_str = f"{results['w\_EDE']:.2f} ± {results['w\_error']:.2f}"

print(f"EDE Model Results:")

print(f"H0: {h0\_str} km/s/Mpc (ΛCDM baseline: {params['h']\*100:.1f})")

print(f"Sound horizon at z=3000: {results['rs']:.1f} Mpc (ΛCDM: {results['rs\_base']:.1f})")

print(f"Late-time w\_EDE: {w\_str}")

with open("data/ede\_simulation\_results.txt", "w") as f:

f.write("Axion-like EDE Model Results\n")

f.write(f"H0: {h0\_str} km/s/Mpc\n")

f.write(f"Sound horizon: {results['rs']:.1f} Mpc (ΛCDM: {results['rs\_base']:.1f})\n")

f.write(f"w\_EDE: {w\_str}\n")

if \_\_name\_\_ == "\_\_main\_\_":

main()