# simulate\_miniclusters.py - Python simulation mimicking ENZO for axion miniclusters

# Generates minicluster stats at z=3000 for EDE model

import numpy as np

import csv

# Define parameters for EDE and miniclusters

params = {

'z': 3000, # Redshift for EDE peak

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

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

'mass\_mean': 1e6, # Mean mass in solar masses (10^6 M☉)

'size\_mean': 1.0, # Mean size in kiloparsecs (1 kpc)

'num\_clusters': 10, # Number of miniclusters to simulate

'density\_profile': 'r^-2.9' # Typical axion minicluster profile

}

# Simulate minicluster properties with random variation

def generate\_miniclusters(params):

np.random.seed(42) # For reproducibility

masses = np.random.normal(params['mass\_mean'], params['mass\_mean'] \* 0.2, params['num\_clusters'])

sizes = np.random.normal(params['size\_mean'], params['size\_mean'] \* 0.2, params['num\_clusters'])

# Ensure physical bounds (no negative or unrealistic values)

masses = np.clip(masses, 0.5e6, 1.5e6) # Between 0.5e6 and 1.5e6 M☉

sizes = np.clip(sizes, 0.5, 1.5) # Between 0.5 and 1.5 kpc

# Format data for CSV

miniclusters = []

for i in range(params['num\_clusters']):

miniclusters.append({

'Redshift': params['z'],

'Mass': f"{masses[i]:.1e}",

'Size': f"{sizes[i]:.1f}",

'Density\_Profile': params['density\_profile']

})

return miniclusters

# Save to CSV

def save\_miniclusters(miniclusters, filename):

with open(filename, 'w', newline='') as csvfile:

fieldnames = ['Redshift', 'Mass', 'Size', 'Density\_Profile']

writer = csv.DictWriter(csvfile, fieldnames=fieldnames)

writer.writeheader()

for cluster in miniclusters:

writer.writerow(cluster)

# Main execution

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

miniclusters = generate\_miniclusters(params)

save\_miniclusters(miniclusters, "data/minicluster\_data.csv")

# Print sample for verification

print("Generated Minicluster Data (Sample):")

for cluster in miniclusters[:3]: # Show first 3 for brevity

print(f"Redshift: {cluster['Redshift']}, Mass: {cluster['Mass']} M☉, Size: {cluster['Size']} kpc, Profile: {cluster['Density\_Profile']}")