"""

Refined Cluster Dynamics Framework

===============================

"""

class ClusterDynamics:

def \_\_init\_\_(self):

self.cluster\_params = {

'mass\_threshold': 1e14, # Solar masses

'concentration\_param': 4.0,

'gas\_fraction': 0.17

}

def compute\_cluster\_evolution(self, initial\_state, time\_span):

"""

Compute detailed cluster evolution including all components

"""

# Initialize components

dark\_matter = self.initialize\_dark\_matter(initial\_state)

gas = self.initialize\_gas(initial\_state)

galaxies = self.initialize\_galaxies(initial\_state)

# Evolution with temporal flow effects

evolution = self.evolve\_cluster\_system(

dark\_matter, gas, galaxies, time\_span)

return evolution

def compute\_merger\_dynamics(self, cluster1, cluster2):

"""

Compute cluster merger dynamics with temporal flow effects

"""

# Initial conditions

relative\_velocity = self.compute\_relative\_velocity(cluster1, cluster2)

impact\_parameter = self.compute\_impact\_parameter(cluster1, cluster2)

# Merger evolution

merger\_evolution = self.evolve\_merger(

cluster1, cluster2, relative\_velocity, impact\_parameter)

return merger\_evolution

def compute\_mass\_distribution(self, cluster):

"""

Compute detailed mass distribution including temporal flow effects

"""

# Dark matter profile with temporal modification

dm\_profile = self.modified\_nfw\_profile(cluster)

# Gas distribution with pressure support

gas\_profile = self.compute\_gas\_distribution(cluster)

# Galaxy distribution

galaxy\_profile = self.compute\_galaxy\_distribution(cluster)

return {

'dark\_matter': dm\_profile,

'gas': gas\_profile,

'galaxies': galaxy\_profile

}

def modified\_nfw\_profile(self, cluster):

"""

NFW profile modified by temporal flow effects

"""

def nfw(r):

# Standard NFW

x = r/cluster.rs

standard\_nfw = cluster.rho\_s / (x \* (1 + x)\*\*2)

# Temporal flow modification

temporal\_mod = 1.0 + self.compute\_temporal\_modification(r)

return standard\_nfw \* temporal\_mod

return nfw

def compute\_gas\_distribution(self, cluster):

"""

Compute gas distribution with temporal flow effects

"""

def gas\_density(r):

# Beta model with temporal modification

beta\_model = self.compute\_beta\_model(r, cluster)

# Additional temporal flow effects

temporal\_mod = self.compute\_gas\_temporal\_mod(r)

return beta\_model \* temporal\_mod

return gas\_density