Simulation of The Evolution of Primordial Fluctiations into The Cosmic Web

María Constanza Lepe Sabando

Abstract

In this report, we present the implementation of a 3D cosmological simulation using the Leapfrog integration method and the Poisson equation in Fourier space. We analyze the evolution of matter density in the universe, incorporating cosmic expansion through the scale factor $a(t)$. We compare the results with and without cosmic expansion.

1 Introduction

The formation of structures in the universe is a fundamental problem in cosmology. The evolution of primordial density fluctuations is governed by gravity and cosmic expansion. In this work, we simulate the behavior of a density field on a 3D grid, compute the gravitational potential by solving the Poisson equation in Fourier space, and evolve the system using the Leapfrog numerical integration method.

2 Grid Generation and Gaussian Random Field

To model the early universe, we generate a 3D grid of size N^3. With **grid size = N**. This value could be change, but depend of the computer RAM.

```
# size of the grill

grid_size= 128

kx = np.fft.fftfreq(grid_size)*grid_size
ky = np.fft.fftfreq(grid_size)*grid_size
kz = np.fft.fftfreq(grid_size)*grid_size

# np.meshgrid created a coordinate system with kx, ky, kz (frequency grid)

kx, ky, kz = np.meshgrid(kx, ky, kz)

# Wavenumber magnitude k
k = np.sqrt(kx**2 + ky**2 + kz**2)
```

Then, we distributed an initial density field based on a Gaussian random field with a power spectrum $P(k)$.

$$P(k) \propto k^{-3} \tag{1}$$

Where k is the wave number. The real-space density field is obtained by applying the inverse Fourier transform:

$$\delta(x) = \mathcal{F}^{-1}(\delta_k) \tag{2}$$

The Gaussian Random Field is a powerful tool for generating random realizations of stochastic processes that follow a Gaussian probability distribution function. Due to the Central Limit Theorem,

a random variable resulting from the sum of multiple independent processes tends to follow a normal distribution (cita).

Thus, we use it to construct our density field. Here, the power spectrum plays a crucial role, as it determines that larger scales have a greater influence.

```
# Power Spectrum k**-3 (That's mean that larger object in the universe have more
    influence)


P_k = np.where(k > 0, k**-3, 0)


# Creating a 3D Gaussian Random field and delta_k

random_field= np.random.normal(size=(grid_size,grid_size,grid_size))
delta_k=random_field*np.sqrt(P_k)

delta_real=np.fft.ifftn(delta_k).real

#density field in regular space

density_field = np.fft.ifftn(delta_k).real
```

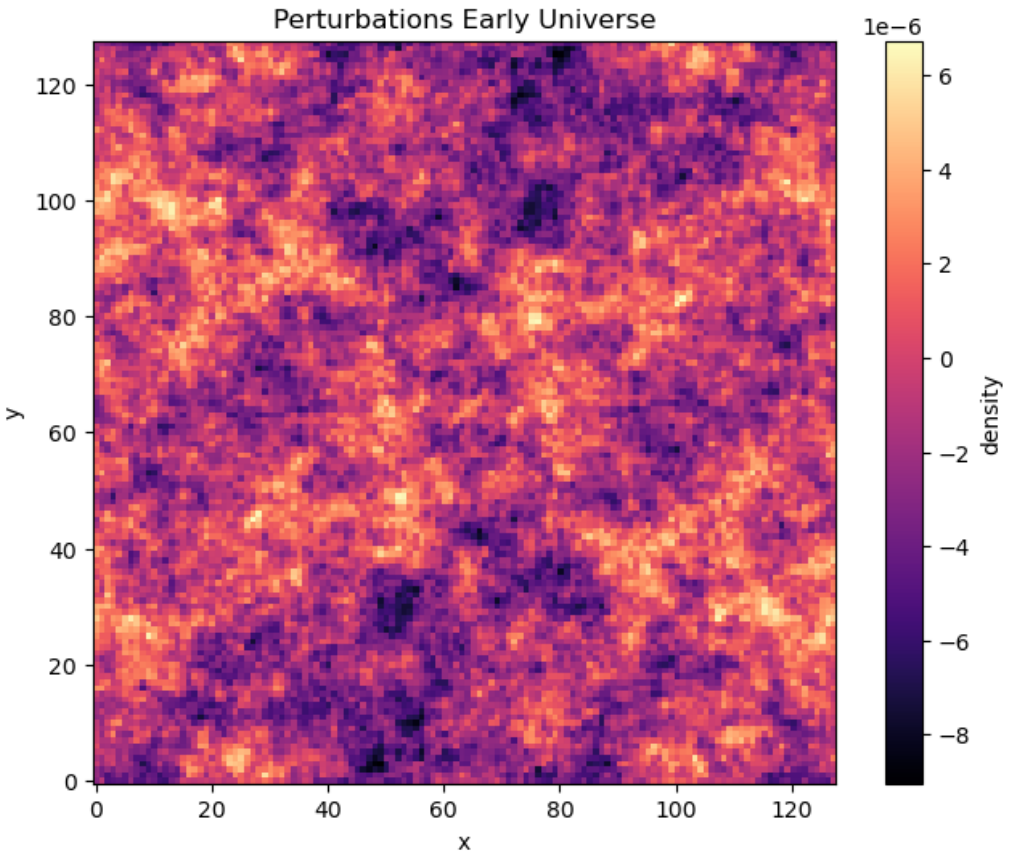

Figure 1: Density $\delta_k(x)$ in a primitive universe.

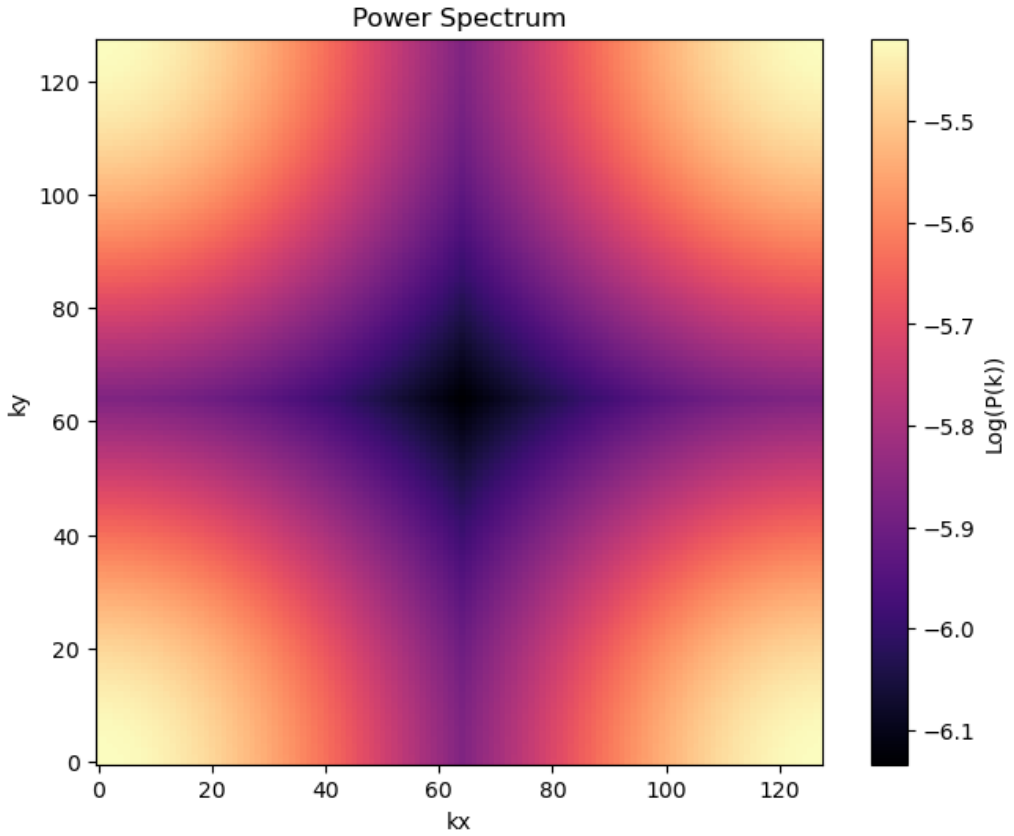

Figure 2: Power spectrum.

3 Gravitational Potential Calculation

The system's evolution is governed by the Poisson equation, which relates matter density to the gravitational potential:

$$\nabla^2 \Phi = 4\pi G \rho \tag{3}$$

In Fourier space, this equation transforms into:

$$\tilde{\Phi}(k) = -\frac{4\pi G \rho_k}{k^2} \tag{4}$$

```
G = 1 # Gravitational constant (generic value)
rho_bar = 1 # universe density

#phi_k= (4* np.pi * G * rho_bar * delta_k) / k**2
phi_k = np.where(k > 0, (-4*np.pi* G * rho_bar * delta_k) / k**2+ 1e-6, 0)

phi_real= np.fft.ifftn(phi_k).real

phi_real -= np.mean(phi_real)
```

4 Numerical Integration with Leapfrog

To evolve the system over time, we use the Leapfrog method, a second-order numerical integration method suitable for gravitational systems. The steps involved are:

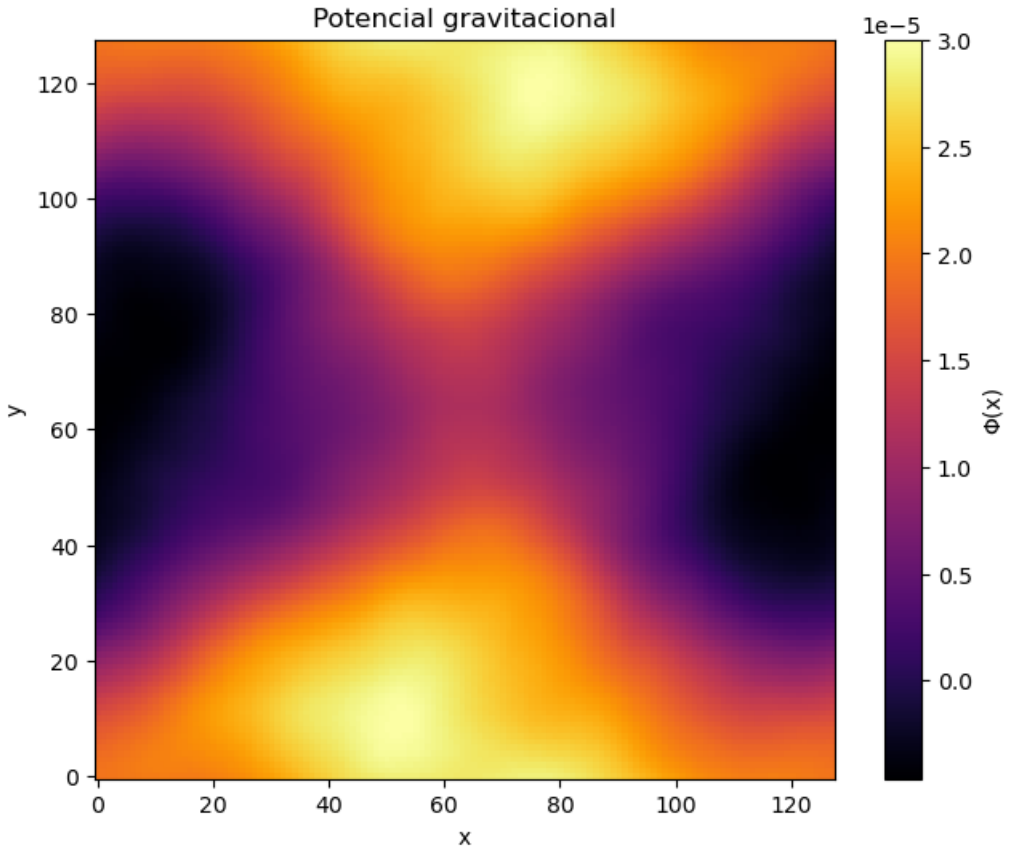

Figure 3: Gravitational potential obtained from the Poisson equation in Fourier space.

1. Update velocities at half-step:

$$v_{t+\frac{\Delta t}{2}} = v_t + \frac{1}{2} a_t \Delta t \tag{5}$$

2. Update positions:

$$x_{t+\Delta t} = x_t + v_{t+\frac{\Delta t}{2}} \Delta t \tag{6}$$

3. Recompute accelerations from the new density field.
4. Update velocities to full step:

$$v_{t+\Delta t} = v_{t+\frac{\Delta t}{2}} + \frac{1}{2} a_{t+\Delta t} \Delta t \tag{7}$$

4.1 Acceleration Computation in Fourier Space

To evolve the density perturbations in our simulation, we need to compute the gravitational acceleration from the potential field. The gravitational potential $\Phi(x)$ is determined by solving Poisson's equation:

$$\nabla^2 \Phi(x) = 4\pi G \rho_{\text{bar}} \delta(x), \tag{8}$$

where $\delta(x)$ represents the density fluctuations, G is the gravitational constant, and ρ_{bar} is the mean density of the universe.

Since computing the Laplacian ∇^2 directly in real space is computationally expensive, we apply the Fourier transform to Poisson's equation. In Fourier space, the Laplacian operator ∇^2 becomes a simple multiplication by $-k^2$, leading to:

$$\hat{\Phi}(k) = -\frac{4\pi G \rho_{\text{bar}} \hat{\delta}(k)}{k^2}. \tag{9}$$

From the gravitational potential, the acceleration field is obtained via:

$$a(x) = -\nabla \Phi(x). \tag{10}$$

Taking the Fourier transform of the gradient, we use the property:

$$\mathcal{F}\left[\frac{\partial \Phi}{\partial x_i}\right] = (ik_i)\hat{\Phi}(k), \tag{11}$$

which gives the acceleration in Fourier space as:

$$\hat{a}_i(k) = -ik_i \hat{\Phi}(k). \tag{12}$$

Finally, the inverse Fourier transform is applied to return the acceleration to real space:

$$a_i(x) = \mathcal{F}^{-1}[\hat{a}_i(k)]. \tag{13}$$

This method allows an efficient computation of the gravitational acceleration using Fast Fourier Transforms (FFT), significantly reducing computational cost compared to real-space finite difference methods.

```

for step in range(num_steps):
    # 1. Update the velocities at half-step (v = v + 0.5 a dt)
    vx += 0.5 * ax_real * dt
    vy += 0.5 * ay_real * dt
    vz += 0.5 * az_real * dt

    # 2. Update positions (x = x + v dt)
    delta_real += vx * dt
    delta_real += vy * dt
    delta_real += vz * dt

    # 3. Recalculate aceleration (as if the density had changed)
    delta_k_new = np.fft.fftn(delta_real)
    phi_k_new = np.where(k > 0, (4 * np.pi * G * rho_bar * delta_k_new) / k**2, 0)
    phi_real_new = np.fft.ifftn(phi_k_new).real

    ax_k_new = np.where(k > 0, -1j * kx * phi_k_new, 0)
    ay_k_new = np.where(k > 0, -1j * ky * phi_k_new, 0)
    az_k_new = np.where(k > 0, -1j * kz * phi_k_new, 0)

    ax_real_new = np.fft.ifftn(ax_k_new).real
    ay_real_new = np.fft.ifftn(ay_k_new).real
    az_real_new = np.fft.ifftn(az_k_new).real

    # 4. Update velocities at full-step (v= v+0.5 a_new dt)
    vx += 0.5 * ax_real_new * dt
    vy += 0.5 * ay_real_new * dt
    vz += 0.5 * az_real_new * dt
```

5 Incorporating Cosmic Expansion

To include cosmic expansion, we introduce the scale factor $a(t)$, which evolves according to the Friedmann equation. In a matter-dominated universe, the scale factor follows:

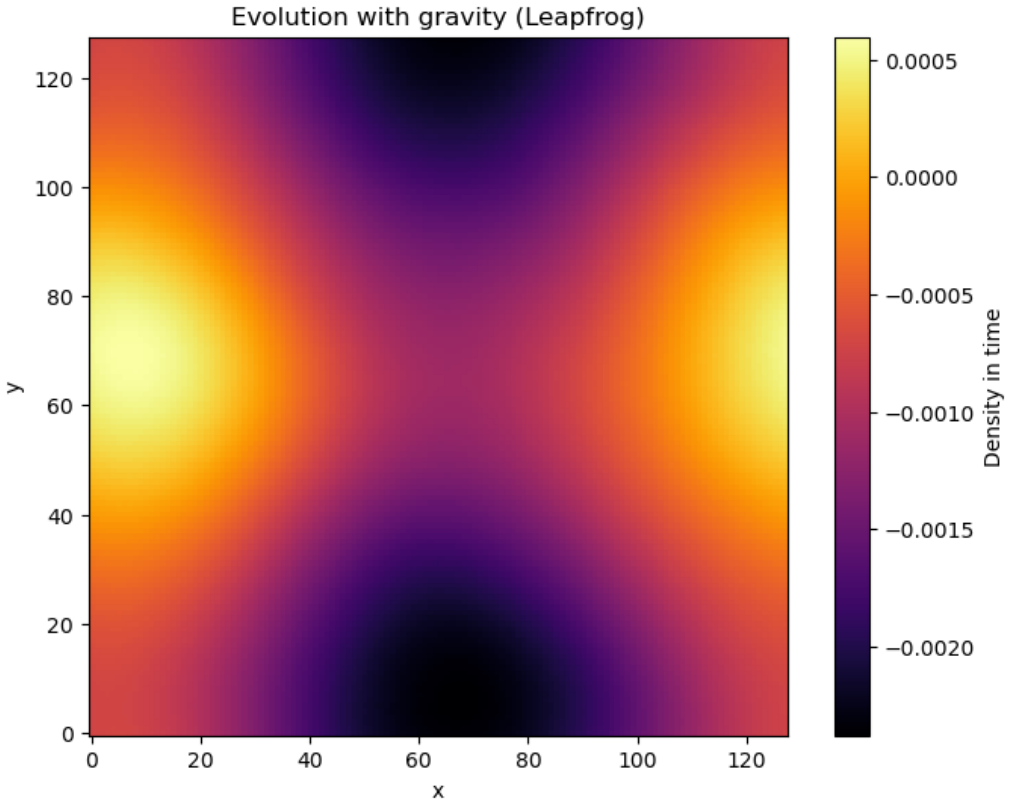

Figure 4: Density evolution over time using Leapfrog integration.

$$a(t) \propto t^{2/3} \tag{14}$$

This implies that the Hubble parameter H is defined as:

$$H = \frac{\dot{a}}{a} \tag{15}$$

In our code, we incorporate these terms as corrections in the velocity and position updates.

```
1
2  # Leapfrog with expansion
3
4   for step in range(num_steps):
5      a =  (1 + H0 * step * dt)**(2/3)
6      H = H0 * np.sqrt(a)
7
8
9      # Half-step velocity update with expansion
10     vx += 0.5 * ax_real * dt * a
11     vy += 0.5 * ay_real * dt * a
12     vz += 0.5 * az_real * dt * a
13
14
15     # Cosmological friction
16     vx *= 1- H*dt
17     vy *= 1- H*dt
18     vz *= 1- H*dt
19
20     # Position update with expansion factor
21     delta_real += (vx * dt * a)  # Es la densidad la que se mueve! no en si las
           particulas
```

```
    delta_real += (vy * dt * a)
    delta_real += (vz * dt * a)


    # Recalculate gravitational potential in Fourier space
    delta_k_new = np.fft.fftn(delta_real)
    phi_k_new = np.where(k > 0, (-4 * np.pi * G * rho_bar * delta_k_new) / (k**2 ),
        0)
    print(phi_k_new)
    phi_real_new = np.fft.ifftn(phi_k_new).real

    # Recalculate acceleration
    ax_k_new = np.where(k > 0, -1j * kx * phi_k_new, 0)
    ay_k_new = np.where(k > 0, -1j * ky * phi_k_new, 0)
    az_k_new = np.where(k > 0, -1j * kz * phi_k_new, 0)

    ax_real_new = np.fft.ifftn(ax_k_new).real
    ay_real_new = np.fft.ifftn(ay_k_new).real
    az_real_new = np.fft.ifftn(az_k_new).real


    # Second half-step with velocities expansion
    vx += 0.5 * ax_real_new * dt * a
    vy += 0.5 * ay_real_new * dt * a
    vz += 0.5 * az_real_new * dt * a
```

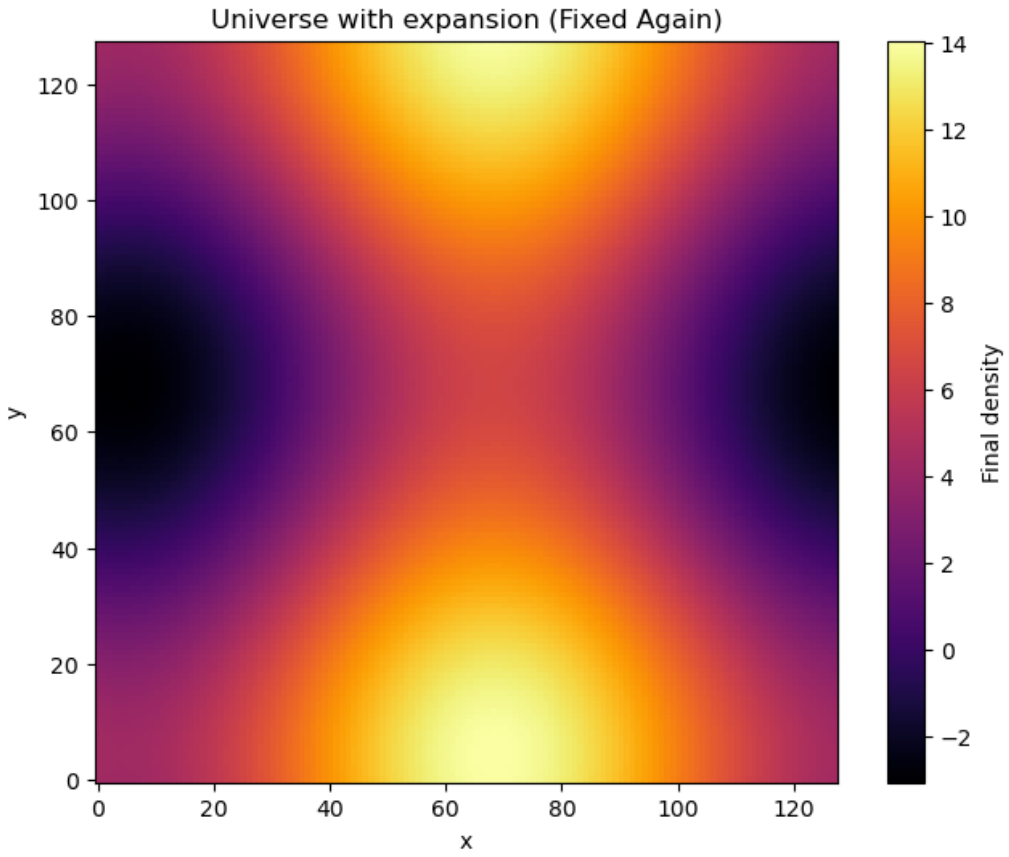

Figure 5: Density evolution including cosmic expansion.

6 SkyServer SDSS

To obtain a dataset of galaxies, the SkyServer SDSS website was used, selecting objects within the region defined by RA min: 49.4°, RA max: 249.6°, DEC min: -1.2°, DEC max: 1.3°. The Pandas library was employed to process the data table efficiently. The resulting galaxy distribution can be observed in Figure 6, where it is evident that galaxies tend to cluster together rather than being randomly distributed.

To further analyze these structures and compare them with real astronomical images, the web-based tool Aladin (Centre de Données astronomiques de Strasbourg - CDS) was utilized Figure 7.

He we show the code use in Skyserver SDSSS:

```
SELECT TOP 1000
  p.objid, p.ra, p.dec, s.z
FROM
  PhotoObj AS p
JOIN
  SpecObj AS s ON p.objid = s.bestobjid
WHERE
  s.class = 'GALAXY' AND s.z BETWEEN 0.01 AND 0.1
```

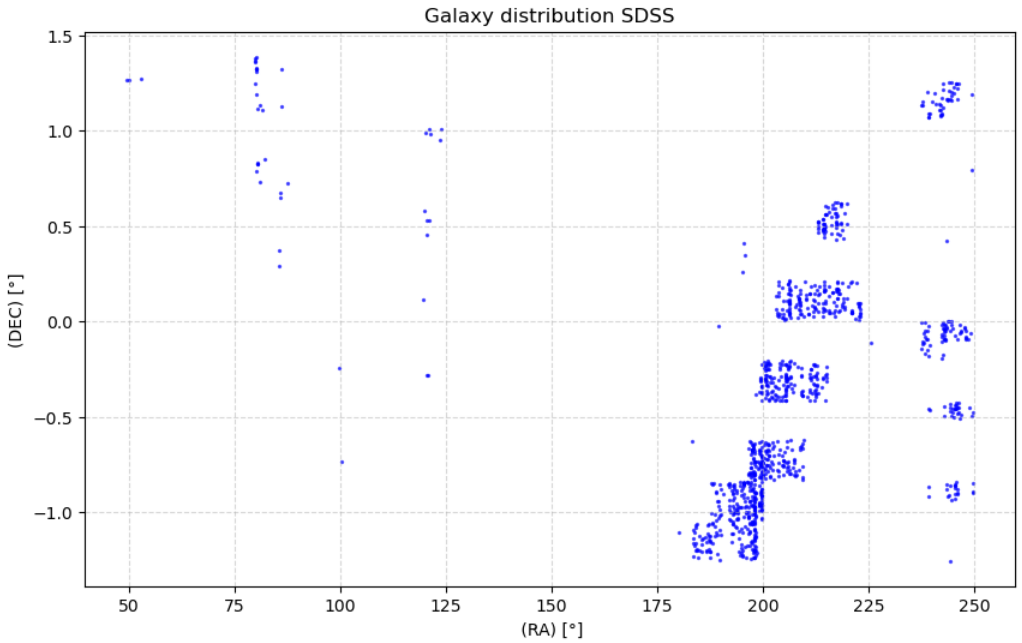

Figure 6: Galaxy distribution SDSS

7 Results and Conclusion

We observe that in the early universe, matter density exhibited a tendency for certain regions to have slightly higher concentrations than others, as shown in Figure 1. In Figure 2, we visualize the gravitational potential corresponding to the density field in Figure 1, where the darker regions represent attractive areas that, over time, should accumulate more matter. This behavior is confirmed through the Leapfrog integration method, illustrated in Figure 4, where matter tends to cluster in these attractive regions uniformly.

However, one aspect that remains unresolved in this project is a numerical inconsistency in the implementation of cosmic expansion. Specifically, the color scale appears to be inverted, suggesting

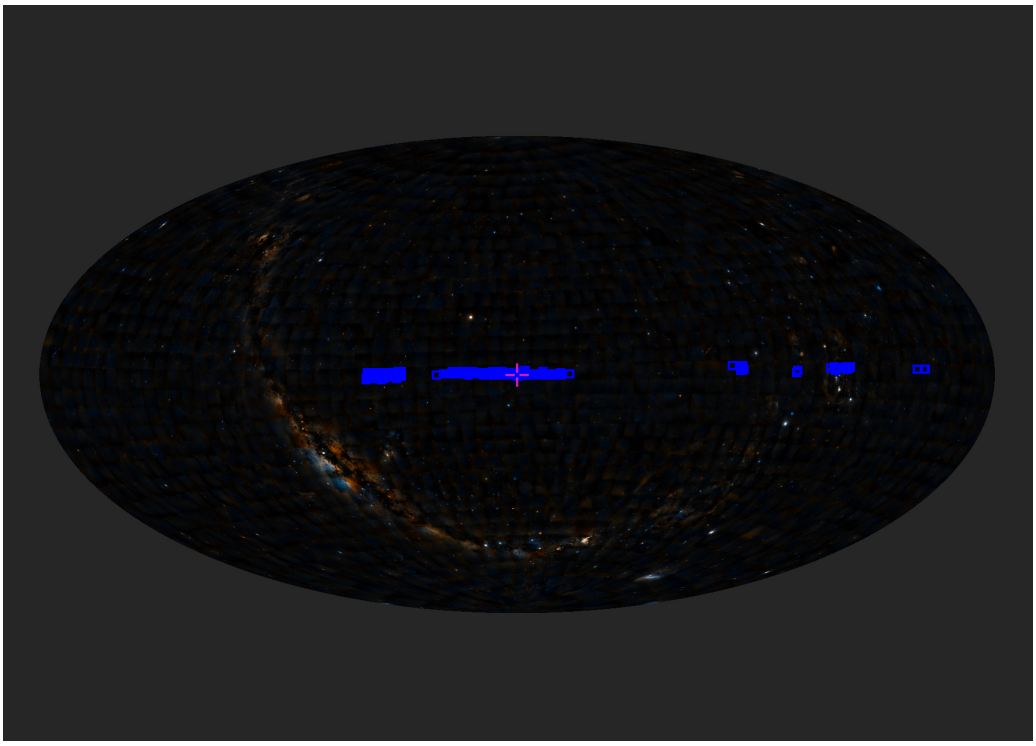

Figure 7: Map galaxy distribution, web-based tool Aladin (Centre de Données astronomiques de Strasbourg - CDS)

that matter is distributed in the opposite regions than expected. Additionally, instead of decreasing due to the expansion of the universe, the matter density shows a slight increase, which contradicts theoretical expectations.