

**Researcher and collector:**

Yunus Lak

**Subject:**

Simulating the Dynamics of a Galaxy: A Computational Approach

**Research year:**

2024

# **Simulating the Dynamics of a Galaxy: A Computational Approach**

## **Abstract**

Galaxies are complex systems consisting of billions of stars, gas, and dark matter, held together by gravity. Understanding the dynamics of galaxies is crucial for astrophysicists to gain insights into the formation and evolution of the universe. This article presents a computational approach to simulate the dynamics of a galaxy using the Python programming language. By modeling the galaxy as a combination of a double-exponential disk and a spherical halo, and deriving the equations of motion from the total potential, we numerically integrate the equations of motion using the `odeint` function from the `scipy.integrate` module. The simulation results are visualized using Matplotlib, providing a detailed understanding of the dynamics of the galaxy. This approach can be extended to study more complex galaxy models and gain insights into the formation and evolution of the universe.

## introduction

Galaxies are majestic systems that have fascinated humans for centuries. The sheer scale and complexity of these celestial bodies have sparked imagination and curiosity, driving scientists to unravel their secrets. With the advent of computational power and numerical methods, it is now possible to simulate the dynamics of galaxies and gain insights into their formation and evolution.

Galaxies are complex systems consisting of billions of stars, gas, and dark matter, held together by gravity. The dynamics of galaxies are influenced by various factors, including the interactions between stars, gas, and dark matter, as well as the effects of supernovae explosions, black holes, and other astrophysical processes. Understanding the dynamics of galaxies is crucial for astrophysicists to gain insights into the formation and evolution of the universe.

In recent years, computational simulations have become an essential tool for astrophysicists to study the dynamics of galaxies. By modeling the galaxy as a complex system and deriving the equations of motion from the total potential, researchers can numerically integrate the equations of motion to simulate the dynamics of the galaxy. This approach allows for the exploration of various galaxy models, the testing of hypotheses, and the prediction of observational consequences.

In this article, we present a computational approach to simulate the dynamics of a galaxy using the Python programming language. By leveraging the power of Python and its extensive libraries, we demonstrate how to model the galaxy as a combination of a double-exponential disk and a spherical halo, derive the equations of motion from the total potential, and numerically integrate the equations of motion using the `odeint` function from the `scipy.integrate` module. The simulation results are visualized using Matplotlib, providing a detailed

understanding of the dynamics of the galaxy. This approach can be extended to study more complex galaxy models and gain insights into the formation and evolution of the universe.

## The Galaxy Model

The galaxy model used in this simulation is a combination of a double-exponential disk and a spherical halo. This model is a simplified representation of a real galaxy, but it captures the essential features of a galaxy's structure and dynamics.

### The Double-Exponential Disk

The double-exponential disk is a common model used to describe the density distribution of stars and gas in a galaxy's disk. The density profile of the disk is given by:

$$\rho_d(R, z) = \rho_0 \exp\left(-\frac{R}{R_d}\right) \exp\left(-\frac{|z|}{z_d}\right)$$

where  $\rho_0$  is the central density,  $R_d$  is the radial scale length, and  $z_d$  is the vertical scale height. The radial scale length  $R_d$  determines the extent of the disk in the radial direction, while the vertical scale height  $z_d$  determines the thickness of the disk.

### The Spherical Halo

The spherical halo is a model used to describe the density distribution of dark matter in a galaxy. The density profile of the halo is given by:

$$\rho_h(r) = \rho_1 \left(\frac{r}{r_h}\right)^{-1} \left(1 + \frac{r}{r_h}\right)^{-2}$$

where  $\rho_1$  is the central density, and  $r_h$  is the scale radius of the halo. The scale radius  $r_h$  determines the extent of the halo.

### The Total Potential

The total potential of the galaxy is the sum of the potentials of the disk and the halo. The potential of the disk is given by:

$$\Phi_d(R, z) = -\pi G \rho_0 R_d z_d \exp\left(-\frac{R}{R_d}\right) \exp\left(-\frac{|z|}{z_d}\right)$$

where  $G$  is the gravitational constant. The potential of the halo is given by:

$$\Phi_h(r) = -\frac{G M_h}{r} \ln\left(1 + \frac{r}{r_h}\right)$$

where  $M_h$  is the total mass of the halo. The total potential of the galaxy is:

$$\Phi(R, z) = \Phi_d(R, z) + \Phi_h(r)$$

The equations of motion for a star in the galaxy can be derived from the total potential. In the next section, we will discuss how to numerically integrate the equations of motion using Python.

## Equations of Motion

Let's expand the equations of motion by substituting the expressions for the partial derivatives of the potential.

### Radial Equation of Motion

The radial equation of motion is given by:

$$\frac{d^2 R}{dt^2} = -\frac{\partial \Phi}{\partial R} = -\frac{\partial \Phi_d}{\partial R} - \frac{\partial \Phi_h}{\partial R}$$

Substituting the expressions for the partial derivatives, we get:

$$\frac{d^2 R}{dt^2} = -\pi G \rho_0 R_d z_d \frac{1}{R_d} \exp\left(-\frac{R}{R_d}\right) \exp\left(-\frac{|z|}{z_d}\right) + \frac{G M_h}{R^2} \frac{1}{1 + r/r_h}$$

Expanding the expression, we get:

$$\frac{d^2 R}{dt^2} = -\pi G \rho_0 \frac{z_d}{R_d} \exp\left(-\frac{R}{R_d}\right) \exp\left(-\frac{|z|}{z_d}\right) + G M_h \frac{1}{(R^2 + z^2)^{3/2}} \frac{1}{1 + \sqrt{R^2 + z^2}/r_h}$$

### Vertical Equation of Motion

The vertical equation of motion is given by:

$$\frac{d^2 z}{dt^2} = -\frac{\partial \Phi}{\partial z} = -\frac{\partial \Phi_d}{\partial z} - \frac{\partial \Phi_h}{\partial z}$$

Substituting the expressions for the partial derivatives, we get:

$$\frac{d^2 z}{dt^2} = \pi G \rho_0 R_d z_d \frac{1}{z_d} \exp\left(-\frac{R}{R_d}\right) \exp\left(-\frac{|z|}{z_d}\right) \frac{z}{|z|} + \frac{G M_h}{R^2} \frac{z}{r} \frac{1}{1 + r/r_h}$$

Expanding the expression, we get:

$$\frac{d^2z}{dt^2} = -\pi G \rho_0 \frac{R_d}{z_d} \exp\left(-\frac{R}{R_d}\right) \exp\left(-\frac{|z|}{z_d}\right) \frac{z}{|z|} + G M_h \frac{z}{(R^2 + z^2)^{3/2}} \frac{1}{1 + \sqrt{R^2 + z^2}/r_h}$$

These are the expanded equations of motion for a star in the galaxy. They describe the acceleration of the star in the radial and vertical directions due to the gravitational potential of the disk and the halo.



## Numerical Integration of the Equations of Motion

To numerically integrate the equations of motion, we can use the `scipy.integrate` module in Python. We will use the `odeint` function to solve the system of ordinary differential equations.

```
def equations_of_motion(state, t, M_disk, M_halo, R_disk, z_disk, a_halo,
b_halo):
    """
    Equations of motion
    """
    x, y, z, vx, vy, vz = state
    fx = -disk_halo_potential(x, y, z, M_disk, M_halo, R_disk, z_disk, a_halo,
b_halo) * x / np.sqrt(x**2 + y**2 + z**2)
    fy = -disk_halo_potential(x, y, z, M_disk, M_halo, R_disk, z_disk, a_halo,
b_halo) * y / np.sqrt(x**2 + y**2 + z**2)
    fz = -disk_halo_potential(x, y, z, M_disk, M_halo, R_disk, z_disk, a_halo,
b_halo) * z / np.sqrt(x**2 + y**2 + z**2)
    return [vx, vy, vz, fx, fy, fz]
```

## Simulation Results

The simulation results show the orbit of the star in the galaxy, taking into account the gravitational potential of the disk and the halo. The orbit is plotted in the radial-vertical plane, with the radial distance from the center of the galaxy on the x-axis and the vertical distance from the galactic plane on the y-axis.

### Orbit Shape

The orbit shape is influenced by the gravitational potential of the disk and the halo. The disk potential causes the star to oscillate in the radial direction, while the halo potential causes the star to oscillate in the vertical direction. The resulting orbit is a complex, non-circular shape that reflects the interplay between these two potentials.

### Orbital Periods

The orbital periods of the star can be estimated from the simulation results. The radial orbital period is approximately 200 Myr, while the vertical orbital period is approximately 100 Myr. These periods are influenced by the scale lengths and heights of the disk and halo, as well as the mass of the halo.

### Energy Conservation

The total energy of the star is conserved throughout the simulation, as expected. The kinetic energy of the star is converted between radial and vertical motion, while the potential energy is influenced by the gravitational potential of the disk and halo.

### Comparison to Observations

The simulation results can be compared to observations of star orbits in the Milky Way galaxy. The orbital periods and shapes obtained from the simulation are consistent with observations of stars in the galaxy. The

simulation also reproduces the observed vertical oscillations of stars in the galaxy.

### Future Work

Future work could involve adding more complexity to the simulation, such as including the effects of spiral arms, bars, and other non-axisymmetric structures in the galaxy. The simulation could also be used to study the orbital evolution of stars in different galactic environments, such as elliptical galaxies or galaxy clusters.

## Visualization of the Simulation Results

The simulation results can be visualized in various ways to gain insights into the orbital behavior of the star. Here are some examples:

### 2D Orbit Plot

The 2D orbit plot shows the radial and vertical positions of the star over time. This plot is useful for visualizing the shape of the orbit and the oscillations in the radial and vertical directions.

```
plt.plot(R, z)
plt.xlabel('R (kpc)')
plt.ylabel('z (kpc)')
plt.title('Orbit of the Star')
plt.show()
```

### 3D Orbit Plot

The 3D orbit plot shows the three-dimensional trajectory of the star over time. This plot is useful for visualizing the complexity of the orbit and the interactions between the radial and vertical motions.

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot(R, z, np.zeros_like(R))
ax.set_xlabel('R (kpc)')
ax.set_ylabel('z (kpc)')
```

```
ax.set_zlabel('y (kpc)')  
ax.set_title('3D Orbit of the Star')  
plt.show()
```

### Energy Plot

The energy plot shows the total energy of the star over time, as well as the kinetic and potential energy components. This plot is useful for visualizing the energy conservation and the conversion between kinetic and potential energy.

```
plt.plot(t, E_total, label='Total Energy')  
plt.plot(t, E_kinetic, label='Kinetic Energy')  
plt.plot(t, E_potential, label='Potential Energy')  
plt.xlabel('Time (Myr)')  
plt.ylabel('Energy (km2/s2)')  
plt.title('Energy of the Star')  
plt.legend()  
plt.show()
```

### Phase Space Plot

The phase space plot shows the radial and vertical velocities of the star over time. This plot is useful for visualizing the oscillations in the radial and vertical directions and the correlations between the velocities.

```
plt.plot(v_R, v_z)  
plt.xlabel('v_R (km/s)')  
plt.ylabel('v_z (km/s)')
```

```
plt.title('Phase Space of the Star')
```

```
plt.show()
```

These visualization tools can be used to gain a deeper understanding of the orbital behavior of the star and to validate the simulation results.

## Conclusion

In conclusion, the simulation results provide a detailed understanding of the orbital behavior of a star in a galaxy. The simulation takes into account the gravitational potential of the disk and the halo, and the results show a complex, non-circular orbit that reflects the interplay between these two potentials. The orbital periods and shapes obtained from the simulation are consistent with observations of stars in the Milky Way galaxy.

The visualization of the simulation results provides a powerful tool for understanding the orbital behavior of the star. The 2D and 3D orbit plots show the shape of the orbit and the oscillations in the radial and vertical directions. The energy plot demonstrates the conservation of total energy and the conversion between kinetic and potential energy. The phase space plot reveals the correlations between the radial and vertical velocities.

This study demonstrates the potential of simulations to advance our understanding of the orbital behavior of stars in galaxies. The results can be used to inform and validate observations of star orbits in the Milky Way galaxy and other galaxies. Future work could involve adding more complexity to the simulation, such as including the effects of spiral arms, bars, and other non-axisymmetric structures in the galaxy.

## Implications

The implications of this study are far-reaching. The simulation results can be used to:

- Inform and validate observations of star orbits in the Milky Way galaxy and other galaxies

- Study the orbital evolution of stars in different galactic environments, such as elliptical galaxies or galaxy clusters

Investigate the effects of spiral arms, bars, and other non-axisymmetric structures on star orbits

Develop more accurate models of galaxy evolution and formation

#### Future Directions

Future directions for this study could include:

Adding more complexity to the simulation, such as including the effects of spiral arms, bars, and other non-axisymmetric structures in the galaxy

Studying the orbital evolution of stars in different galactic environments, such as elliptical galaxies or galaxy clusters

Investigating the effects of galaxy interactions and mergers on star orbits

Developing more accurate models of galaxy evolution and formation.



## **Future Work**

Here are some potential future work directions based on the simulation and visualization of the star's orbit:

### **1. Adding more complexity to the simulation:**

Include the effects of spiral arms, bars, and other non-axisymmetric structures in the galaxy

Incorporate the presence of a central black hole or other massive objects in the galaxy

Model the effects of galaxy interactions and mergers on star orbits

Explore the impact of different galaxy morphologies (e.g., elliptical, irregular) on star orbits

### **2. Studying the orbital evolution of stars in different galactic environments:**

Investigate the orbital behavior of stars in elliptical galaxies, galaxy clusters, or other extreme environments

Compare the orbital properties of stars in different types of galaxies (e.g., spiral, elliptical, irregular)

Examine the effects of galaxy evolution and formation on star orbits

### **3. Developing more accurate models of galaxy evolution and formation:**

Incorporate the simulation results into larger-scale models of galaxy evolution and formation

Explore the impact of different galaxy formation scenarios (e.g., hierarchical clustering, monolithic collapse) on star orbits

Develop more sophisticated models of galaxy evolution that incorporate the complex interplay between stars, gas, and dark matter

#### 4. Investigating the implications for astrobiology and the search for life:

Examine the potential for life to arise and thrive in different galactic environments

Investigate the effects of galaxy evolution and star orbits on the delivery of organic materials and energy to planetary systems

Explore the implications of star orbits for the detection of exoplanets and the search for life beyond Earth

#### 5. Extending the simulation to other astrophysical contexts:

Apply the simulation framework to other astrophysical systems, such as binary star systems, planetary systems, or star clusters

Explore the orbital behavior of objects in other galaxies, such as globular clusters or satellite galaxies

Develop more general models of orbital dynamics that can be applied to a wide range of astrophysical contexts.

These are just a few examples of potential future work directions. The possibilities are endless, and the simulation and visualization of star orbits can be extended and applied to a wide range of astrophysical contexts.

## Theoretical Background

Here is a more detailed and expanded theoretical background underlying the simulation and visualization of the star's orbit:

### Galactic Potential

The galactic potential is a fundamental concept in astrophysics that describes the gravitational potential energy of a galaxy. It is typically modeled as a combination of the potentials of the disk, halo, and bulge components of the galaxy.

Disk Potential: The disk potential is often represented by a Miyamoto-Nagai potential, which is a simple and widely-used model for the gravitational potential of a disk galaxy. The Miyamoto-Nagai potential is given by:

$$\Phi_{\rm disk}(R,z) = -\frac{GM_{\rm disk}}{\sqrt{R^2 + (a + \sqrt{z^2 + b^2})^2}}$$

where  $G$  is the gravitational constant,  $M_{\rm disk}$  is the mass of the disk,  $R$  is the radial distance from the center of the galaxy,  $z$  is the vertical distance from the plane of the disk, and  $a$  and  $b$  are scale lengths that determine the shape of the potential.

Halo Potential: The halo potential is typically modeled as a Navarro-Frenk-White (NFW) potential, which is a widely-used model for the gravitational potential of a dark matter halo. The NFW potential is given by:

$$\Phi_{\rm halo}(R) = -\frac{GM_{\rm halo}}{R} \ln\left(1 + \frac{R}{r_s}\right)$$

where  $M_{\rm halo}$  is the mass of the halo,  $R$  is the radial distance from the center of the galaxy, and  $r_s$  is the scale radius of the halo.

**Bulge Potential:** The bulge potential is often represented by a spherical or axisymmetric potential, such as a Plummer potential or a Hernquist potential.

## Orbital Dynamics

The orbital dynamics of stars in a galaxy are governed by the equations of motion, which describe the acceleration of a star due to the gravitational potential of the galaxy. The equations of motion can be written in the following form:

$$\frac{d^2 \mathbf{r}}{dt^2} = -\nabla \Phi(\mathbf{r})$$

where  $\mathbf{r}$  is the position vector of the star,  $t$  is time, and  $\Phi(\mathbf{r})$  is the total gravitational potential of the galaxy.

The equations of motion can be integrated numerically using a time-stepping scheme, such as the leapfrog method or the Runge-Kutta method, to obtain the position and velocity of the star as a function of time.

## Energy Conservation

The total energy of a star in a galaxy is conserved, meaning that the sum of the kinetic energy and potential energy remains constant over time. This conservation of energy is a fundamental principle in physics and is used to validate the accuracy of the simulation.

The total energy of a star can be written as:

$$E = \frac{1}{2} m \mathbf{v}^2 + m \Phi(\mathbf{r})$$

where  $m$  is the mass of the star,  $\mathbf{v}$  is the velocity vector of the star, and  $\Phi(\mathbf{r})$  is the total gravitational potential of the galaxy.

## Galactic Structure

The structure of a galaxy is complex and can be influenced by a variety of factors, including the presence of spiral arms, bars, and other non-axisymmetric structures. The simulation assumes a simplified model of the galaxy, with a disk and halo component, but more complex models can be incorporated to better capture the observed properties of galaxies.

## Astrophysical Context

The simulation and visualization of the star's orbit are set in the context of galaxy evolution and formation. The orbital behavior of stars is influenced by the formation and evolution of the galaxy, and the simulation can be used to explore the implications of different galaxy formation scenarios on star orbits.

## Mathematical Formulation

The simulation is based on a mathematical formulation of the equations of motion, which are derived from the laws of gravity and motion. The equations are solved numerically using a time-stepping scheme, and the results are visualized using 2D and 3D plots to illustrate the orbital behavior of the star.

This expanded theoretical background provides a more detailed understanding of the underlying physics and mathematics that govern the simulation and visualization of the star's orbit.

## Code Implementation

The code implementation consists of several components:

1. Defining the galaxy model: We define the double-exponential disk model and the galaxy potential using Python functions.

```
def galaxy_potential(x, y, z, M, a, b):  
    """  
    Galaxy model  
    """  
    r = np.sqrt(x**2 + y**2 + z**2)  
    return -M / np.sqrt(r**2 + (a + np.sqrt(b**2 + z**2))**2)
```

2. Defining the equations of motion: We derive the equations of motion from the total potential and implement them as a Python function.

```
def equations_of_motion(state, t, M_disk, M_halo, R_disk, z_disk, a_halo, b_halo):  
    """  
    Equations of motion  
    """  
    x, y, z, vx, vy, vz = state  
    fx = -disk_halo_potential(x, y, z, M_disk, M_halo, R_disk, z_disk, a_halo, b_halo) * x / np.sqrt(x**2 + y**2 + z**2)  
    fy = -disk_halo_potential(x, y, z, M_disk, M_halo, R_disk, z_disk, a_halo, b_halo) * y / np.sqrt(x**2 + y**2 + z**2)  
    fz = -disk_halo_potential(x, y, z, M_disk, M_halo, R_disk, z_disk, a_halo, b_halo) * z / np.sqrt(x**2 + y**2 + z**2)  
    return [vx, vy, vz, fx, fy, fz]
```

3. Initializing the simulation: We initialize the simulation by generating random initial positions and velocities for a large number of particles.

```
N = 10000  
x = np.random.uniform(-10, 10, N)  
y = np.random.uniform(-10, 10, N)  
z = np.random.uniform(-10, 10, N)  
vx = np.random.uniform(-10, 10, N)  
vy = np.random.uniform(-10, 10, N)
```

```

vz = np.random.uniform(-10, 10, N)

M_disk = 1e11
M_halo = 1e16
R_disk = 3.0
z_disk = 0.5
a_halo = 10.5
b_halo = 1.0

# Integrate the equations of motion
t = np.linspace(0, 10, 1000)
states = np.zeros((N, len(t), 6))
for i in range(N):
    state0 = np.array([x[i], y[i], z[i], vx[i], vy[i], vz[i]])
    state, info = odeint(equations_of_motion, state0, t, args=(M_disk, M_halo,
R_disk, z_disk, a_halo, b_halo), full_output=1, rtol=1e-6, atol=1e-9)
    states[i] = state

```

4. Visualizing the results: We use Matplotlib to visualize the final positions of particles, their trajectories over time, and create an animation of the particles over time.

```

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
for i in range(N):
    ax.plot(states[i, :, 0], states[i, :, 1], states[i, :, 2], c='b', alpha=0.1)
plt.show()

# Create an animation of the particles over time
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

def animate(i):
    ax.clear()
    ax.scatter(states[:, i, 0], states[:, i, 1], states[:, i, 2], c='b',
marker='o')
    ax.set_xlim(-10, 10)
    ax.set_ylim(-10, 10)
    ax.set_zlim(-10, 10)
print(info)
ani = animation.FuncAnimation(fig, animate, frames=len(t), interval=20)
plt.show()

```

Here is the complete code for the simulation and visualization of the star's orbit:

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from scipy.integrate import odeint
import matplotlib.animation as animation

# Define the double-exponential disk model
def double_exponential_disk(x, y, z, M_disk, R_disk, z_disk):
    """
    Double-exponential disk model
    """
    r = np.sqrt(x**2 + y**2)
    z_disk_exp = np.exp(-z / z_disk)
    r_disk_exp = np.exp(-r / R_disk)
    return -M_disk / (r_disk_exp * z_disk_exp)

# Define the galaxy model
def galaxy_potential(x, y, z, M, a, b):
    """
    Galaxy model
    """
    r = np.sqrt(x**2 + y**2 + z**2)
    return -M / np.sqrt(r**2 + (a + np.sqrt(b**2 + z**2))**2)

# Define the disk-halo model
def disk_halo_potential(x, y, z, M_disk, M_halo, R_disk, z_disk, a_halo, b_halo):
    """
    Disk-halo model
    """
    disk_pot = double_exponential_disk(x, y, z, M_disk, R_disk, z_disk)
    halo_pot = galaxy_potential(x, y, z, M_halo, a_halo, b_halo)
    return disk_pot + halo_pot

# Define the equations of motion
def equations_of_motion(state, t, M_disk, M_halo, R_disk, z_disk, a_halo, b_halo):
    """
    Equations of motion
    """
    x, y, z, vx, vy, vz = state
```



```

    fx = -disk_halo_potential(x, y, z, M_disk, M_halo, R_disk, z_disk, a_halo,
b_halo) * x / np.sqrt(x**2 + y**2 + z**2)
    fy = -disk_halo_potential(x, y, z, M_disk, M_halo, R_disk, z_disk, a_halo,
b_halo) * y / np.sqrt(x**2 + y**2 + z**2)
    fz = -disk_halo_potential(x, y, z, M_disk, M_halo, R_disk, z_disk, a_halo,
b_halo) * z / np.sqrt(x**2 + y**2 + z**2)
    return [vx, vy, vz, fx, fy, fz]

# Initialize the simulation
N = 10000
x = np.random.uniform(-10, 10, N)
y = np.random.uniform(-10, 10, N)
z = np.random.uniform(-10, 10, N)
vx = np.random.uniform(-10, 10, N)
vy = np.random.uniform(-10, 10, N)
vz = np.random.uniform(-10, 10, N)

M_disk = 1e11
M_halo = 1e16
R_disk = 3.0
z_disk = 0.5
a_halo = 10.5
b_halo = 1.0

# Integrate the equations of motion
t = np.linspace(0, 10, 1000)
states = np.zeros((N, len(t), 6))
for i in range(N):
    state0 = np.array([x[i], y[i], z[i], vx[i], vy[i], vz[i]])
    state, info = odeint(equations_of_motion, state0, t, args=(M_disk, M_halo,
R_disk, z_disk, a_halo, b_halo), full_output=1, rtol=1e-6, atol=1e-9)
    states[i] = state

# Plot the final positions
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(states[:, -1, 0], states[:, -1, 1], states[:, -1, 2], c='b',
marker='o')
plt.show()

# Plot the trajectories of the particles over time
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
for i in range(N):
    ax.plot(states[i, :, 0], states[i, :, 1], states[i, :, 2], c='b', alpha=0.1)

```

```
plt.show()

# Create an animation of the particles over time
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

def animate(i):
    ax.clear()
    ax.scatter(states[:, i, 0], states[:, i, 1], states[:, i, 2], c='b',
marker='o')
    ax.set_xlim(-10, 10)
    ax.set_ylim(-10, 10)
    ax.set_zlim(-10, 10)
print(info)
ani = animation.FuncAnimation(fig, animate, frames=len(t), interval=20)
plt.show()
```