ASTR 400B Homework 7: Orbit Integration

Due: March 27 2025

For this assignment you are going to write a code that will predict the future trajectory of M33 in the center of mass frame of M31. You will need the COM orbit files for M33 and M31 that you computed in Homework 6 so that you can plot their relative positions and velocities as a function of time and compare them to the analytic solutions.

See Patel, Besla, Sohn 2017a (https://arxiv.org/abs/1609.04823) for more details and background on M33's past orbital trajectory.

Note that $G = 4.498768\text{e}{-6}$ in units of kpc^3/M$_\odot$/Gyr^2 (define this as a global variable, but let's forget about storing the units and use values in this homework).

1 M33AnalyticOrbit

Create a Class called `M33AnalyticOrbit`.

In this Class, we will create a series of functions that will determine the acceleration M33 feels from M31 and integrate its current position and velocity forwards in time.

Initialize the Class (`def __init__`), taking as input a filename for the file in which you will store the integrated orbit. At the beginning of the class, also initialize the following quantities:

- The relative position and velocity of M33 to M31
  - `self.r` : the current relative COM position vector of M33 to M31 calculated using the disk particles of both galaxies at Snapshot 0 (location of M33 today) from Assignment 4. This is a *vector*, not the magnitude. So the x component would be self.r[0].
  - `self.v` : the current relative COM velocity vector of M33 to M31 at Snapshot 0.
- The scale lengths and masses for each component in M31
  - `self.rdisk` and `self.Mdisk` : set `self.rdisk`=5 kpc and use the disk mass computed in Assignment 3 (the value of mass should be in units of M$_\odot$, same for below)

- `self.rbulge` and `self.Mbulge` : set `self.rbulge`=1 kpc and use the bulge mass from Assignment 3
- `self.rhalo` and `self.Mhalo` : for `self.rhalo`, use the Hernquist scale length (a) computed in Assignment 5 and use the halo mass from Assignment 3

2 Define the Acceleration Terms

Define functions that will compute the gravitational acceleration **vectors** from 3 components of the M31 galaxy: Halo, Bulge and Disk.

2.1 Halo and Bulge Acceleration

1. The gravitational acceleration induced by a Hernquist profile is given by :

$$\boldsymbol{a} = -\nabla\Phi_{\text{Hernquist}} = -\frac{GM}{r_{mag}(r_a + r_{mag})^2}\boldsymbol{r} = \begin{pmatrix} -\frac{GM}{r_{mag}(r_a+r_{mag})^2}x \\ -\frac{GM}{r_{mag}(r_a+r_{mag})^2}y \\ -\frac{GM}{r_{mag}(r_a+r_{mag})^2}z \end{pmatrix} \tag{1}$$

   where M is the total halo or bulge mass, r_a is the corresponding scale length, r_{mag} is the magnitude of the relative position vector, where the vector is $\boldsymbol{r}$.

2. Define a function **HernquistAccel** that takes 3 inputs: `self, M, r_a, r`. Again **r** is a vector.

3. This function returns the acceleration **vector** from a Hernquist potential.

4. This function will be used for both the halo and the bulge of M31 (set by the input `M` and `r_a`).

2.2 Disk Acceleration

1. For the disk of M31, we will use an approximation that mimics the exponential disk profile at distances far from the disk. It is called a Miyamoto-Nagai 1975 profile. The potential for this profile is:

$$\Phi_{\text{MN}}(r) = \frac{-GM_{\text{disk}}}{\sqrt{R^2 + B^2}}, \quad \text{where} \quad R = \sqrt{x^2 + y^2} \quad \text{and} \quad B = r_d + \sqrt{z^2 + z_d^2}, \tag{2}$$

   where r_d is `self.rdisk` in our code and z_d can be defined as `self.rdisk/5.0` in this homework. The acceleration vector is thus given by:

$$\boldsymbol{a} = -\nabla\Phi_{\text{MN}} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = \begin{pmatrix} -\frac{GM_{\text{disk}}}{(R^2 + B^2)^{1.5}}x \\ -\frac{GM_{\text{disk}}}{(R^2 + B^2)^{1.5}}y \\ -\frac{GM_{\text{disk}}B}{(R^2 + B^2)^{1.5}\sqrt{z^2 + z_d^2}}z \end{pmatrix}. \tag{3}$$

Note the z component of the acceleration vector is a bit more complicated.

Equivalently the acceleration can be written in terms of the vector $\mathbf{r}$ as

$$\boldsymbol{a} = -\nabla\Phi_{\rm MN} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = -\frac{GM_{\rm disk}}{(R^2+B^2)^{1.5}}\boldsymbol{r}\begin{pmatrix} 1 \\ 1 \\ \dfrac{B}{\sqrt{z^2+z_d^2}} \end{pmatrix}. \tag{4}$$

Where the right hand side of the equation can be written as an array: `np.array( [ 1, 1,` $B/np.sqrt(z^2 + z_d^2)$

2. Define a function **MiyamotoNagaiAccel** that takes 4 inputs: `self, M, r_d, r`.
3. This function returns the acceleration **vector** from a Miyamoto-Nagai profile.

2.3 M31Acceleration

Define a new function **M31Accel** that sums all acceleration vectors from each galaxy component (be careful about the vector summation; you can always take advantage of the array operations provided by `numpy`). This function takes as input the 3D position vector (`x,y,z`) and returns a 3D vector of the total acceleration.

3 Build an Integrator

We need to solve the orbit of M33 by integrating the following equation of motion forwards in time

$$\ddot{\mathbf{r}} = -\nabla\Phi_{\rm total} = -\nabla\Big[\Phi_{\rm halo}(\mathbf{r}) + \Phi_{\rm bulge}(\mathbf{r}) + \Phi_{\rm disk}(\mathbf{r})\Big] \tag{5}$$

$$\Rightarrow \ddot{\mathbf{r}} = \boldsymbol{a}_{\rm halo} + \boldsymbol{a}_{\rm bulge} + \boldsymbol{a}_{\rm disk}, \tag{6}$$

where Φ represents the potential for each galaxy component. To do this, we will treat M33 as a point mass and adopt a variant of the "Leap Frog" integration scheme given that $\boldsymbol{a}$ is a pure function of $\boldsymbol{r}$. In your code, define a function **LeapFrog** that takes as input:

- a time interval for integration `dt` (Δt)
- a starting position vector $\mathbf{r}$ for the M33 COM position relative to the M31
- a starting velocity vector $\mathbf{v}$ for the M33 relative to M31

In each integration step, your code will update the positions and velocities using standard kinematic equations, according to the following (see also Figure 1):

1. Predict the 3D position vector of M33's center of mass (COM) at the middle of the timestep Δt using the current COM velocity and position vectors according to

$$\boldsymbol{r}_{n+\frac{1}{2}} = \boldsymbol{r}_n + \boldsymbol{v}_n \frac{\Delta t}{2}, \tag{7}$$

   storing $\boldsymbol{r}_{n+\frac{1}{2}}$ as a new variable `rhalf`.

2. The COM velocity vector is then advanced a full time step using the acceleration at the 1/2 timestep (calculated based on the $\boldsymbol{r}_{n+\frac{1}{2}}$) according to

$$\boldsymbol{v}_{n+1} = \boldsymbol{v}_n + \boldsymbol{a}_{n+\frac{1}{2}} \Delta t. \tag{8}$$

   The acceleration is determined by calling the function `self.M31Accel`, using the half-step position vector `rhalf` as the input. Store $\boldsymbol{v}_{n+1}$ as a new variable `vnew`.

3. Lastly, the position vector is advanced to the full time step using the average velocity during the time step, with

$$\boldsymbol{r}_{n+1} = \boldsymbol{r}_n + \frac{1}{2}\Big[\boldsymbol{v}_n + \boldsymbol{v}_{n+1}\Big]\Delta t, \tag{9}$$

   which is equivalent to

$$\boldsymbol{r}_{n+1} = \boldsymbol{r}_{n+\frac{1}{2}} + \boldsymbol{v}_{n+1}\frac{\Delta t}{2}. \tag{10}$$

   You may use either one. Store $\boldsymbol{r}_{n+1}$ as a new variable `rnew`.

4. Return rnew and vnew.

Note that Δt can be positive or negative because Leap Frog integrators are symplectic, meaning they can be used for calculations that run both forward and backward in time. For this assignment, you'll want positive values since we are calculating future orbits.

4 Integrate the Orbit

Now we will loop over the LeapFrog integrator to solve the equations of motion and compute the future orbit of M33 for 10 Gyr into the future.

- Define a function **OrbitIntegrator** that takes as input: `self`, a starting time t_o, a time interval Δt and a final time $t_{\max}$.

- The initial conditions of the integration are the starting COM position and velocity of M33 **relative to M31**, which you defined in the `__init__` function earlier.

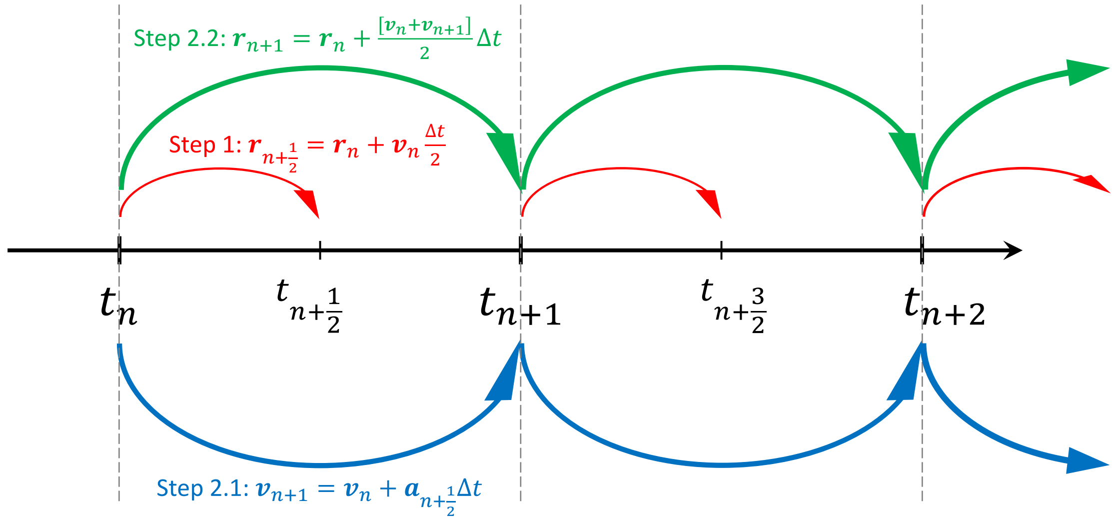

Figure 1: This figure illustrates the LeapFrog integration scheme to evolve the equation of motion of M33. The subscript (n, $n+1$, etc.) denotes the steps along the time axis.

- Define a variable `t` and start the integration from t_o. Continue looping (i.e. a while loop) over **LeapFrog** until you reach $t_{\rm max}$, updating the time, positions and velocities along the way. Store the results in an array that you initialize outside the loop (as you did when you calculated the COM orbits from the simulations). Don't forget to store the initial positions and velocities before you begin the loop.

- Once the loop is complete, store the array into a file, like in Homework 6.

5 Analysis

1. Create a plot of your predicted M33 orbit from $t_o = 0$ Gyr to $t_{\rm max} = 10$ Gyr. Start with 0.5 Gyr intervals for Δt and refine (for example, to 0.1 Gyr) once you know the code is working. Overplot the solution to Assignment 6 for M33's orbit **with respect to M31** from the simulation. Do this for both the total position and total velocity as a function of time.

2. How do the plots compare?

3. What missing physics could make the difference?

4. The MW is missing in these calculations. How might you include its effects?