ASTR 400B Homework 6

Due: Feb 27th 2025, 5 PM

In this assignment you will store the separation and relative velocities of the center of mass of the simulated MW, M33 and M31 over the entire simulation and plot the corresponding orbits.

Please use files located in the directory `/home/astr400b/VLowRes` on nimoy. They are smaller files and will be easier to handle. If you are using nimoy, create a symbolic link to this directory rather than copying the files over to your own directory.

If you want to use the files on your own computer there is a tar.gz file for each galaxy in that directory. You can download the file using sftp. Note this might take a bit of time.

sftp username@nimoy.as.arizona.edu
*get *.tar.gz*

and then untar the files using the below in the command line:
tar -xvzf filename.tar.gz

You will need to import *ReadFile.py* and *CenterOfMass.py* (along with numpy, astropy, matplotlib).

If you run into errors or not sure what part of the code is not working following your design, `print` function is always your friend. Try to `print` the variables out to see if the content is what you want.

We have provided a template file to help with the assignment.

1 Modify existing Code

A modification to *CenterOfMass.py*: in Homework4 we decreased RMAX by 2 to refine the volume. Create a new version of CenterOfMass.py, where you modify COM_P so that it now takes an input *volDec* that defines the amount by which RMAX is decreased. Replace "RMAX/2" with "RMAX/volDec" everywhere it appears.

2 Looping over all Files

Create a function called *OrbitCOM* that takes the following as input:

- *galaxy* the name of the galaxy, e.g. "MW"
- *start* the number of the first snapshot to be read in.
- *end* the number of the last snapshot to be read in.
- *n* an integer indicating the intervals over which you will return the COM.

For this assignment you will compute the orbits up to Snapshot 800 (∼12 Gyr). BUT we will output values in intervals of n=5.

This function will compute the time and COM position and velocity vectors of a given galaxy in each snapshot and save that output into a file (we don't want to run this code repeatedly).

1. Define a string for the filename for the file that will store the orbit: *fileout = "Orbit_galaxyname.txt"*.
2. Set *delta* and *volDec* to be used with a *CenterOfMass* object. Use *delta = 0.1* and *volDec = 2*. For M33, *volDec* needs to be larger because it will be severely tidally stripped towards the end of the simulation. *volDec=4* is a good choice for M33.
3. Define an array *snap_ids* that stores the snapshot id sequence using np.arange (from *start* to *end*). **Himansh: Note that we only want snapshots at an interval of n.** Bonus: check if the input is eligible - make the code stop if the array is empty.
4. Define an array called *orbit* using *np.zeros* with the same number of rows as snap_ids and 7 columns (usage: *np.zeros([number of rows, number of columns])*). This array will store the time, x, y, z, vx, vy, vz of the COM of the galaxy at each snapshot.
5. Set up a **for loop** to compute the COM position and velocities at each snapshot of the simulation. For the convenience of generating the filenames, use the following:

   for i, snap_id in enumerate(snap_ids):

   In this way, if *start* is set to 0, *end* is 800, and *n* is 5, then *i* will be 0, 1, 2, 3, ..., 160, and *snap_id* will be 0, 5, 10, 15, ..., 800. Now do the following in the loop:

   - Define the filename for the galaxy you are going to read in. See instructions in Homework 5 (ilbl = '000' + str(snap_id), etc).
   - Create a *CenterOfMass* object using disk particles.
   - Use the *COM_P* and *COM_V* functions in your *CenterOfMass* object to store the position and velocity of the center of mass (just as you did in Homework 4).
   - In the first column of *Orbit*, store the time **in Gyr** (divide by 1000).

- Recall that in *CenterOfMass*, you initialize the class with a *member* variable called *self.time*. This means it is a property of the instance of the class. So if you called *COM= CenterOfMass(...)* etc, then you can use *COM.time* to retrieve the time of the snapshot.
- The row index of the *Orbit* array (that we use to save the time) is given by i
- **NOTE:** Since time, x, y, z, vx, vy, vz have different units, it would be quite complicated to store them with units in one numpy array. Instead, we will just store their values. For any variable with astropy units, you can obtain their pure values by `var.value`. For example, if the variable `x = 10 kpc`, then `x.value` will be `10`.

- Store the COM Position and Velocity in the *Orbit* array. Remember to use `var.value` to only store their pure values.
- print the counter of the for loop to the screen so you know where the code is at.

6. Save the array *Orbit* to a file: *np.savetxt(fileout, orbit, fmt = "%11.3f"*7, comments='#', header="{: 10s}{: 11s}{: 11s}{: 11s}{: 11s}{: 11s}{: 11s}" .format('t', 'x', 'y', 'z', 'vx', 'vy', 'vz'))*

To test your code, start with a small range of snapshots.

3 Plotting

1. Use your code to compute the time and center of mass position and velocity vectors (3D coordinates) for each galaxy (MW, M31, M33) from SnapNumber 0 to SnapNumber 800 in intervals of n=5, generating three files storing the COM properties over time. Note: SnapNumber 0= 0 Gyr and Snap 800 = ∼12 Gyr in the future. Depending on your research project you might need to re-run this code with a smaller n later.

2. Read in the COM data files you just created for each galaxy. Note the file has a header that starts with #, which yields the column headings.

3. Create a function that computes the difference between two vectors and returns the magnitude of that vector. Use this function to compute the magnitude of the relative separation and velocity of : 1) MW and M31; and 2) M33 and M31.

4. Plot the magnitude of the separation between: 1) MW and M31 and 2) M33 and M31 as a function of time. To do this, you will need to subtract the X,Y,Z positions of the MW from those of the M31. The same for M33 and M31.

5. Plot the magnitude of the relative velocity between: 1) MW and M31 and 2) M33 and M31 as a function of time. Recall that again you need to subtract each component of the velocity vectors. An example is shown in Figure 1.

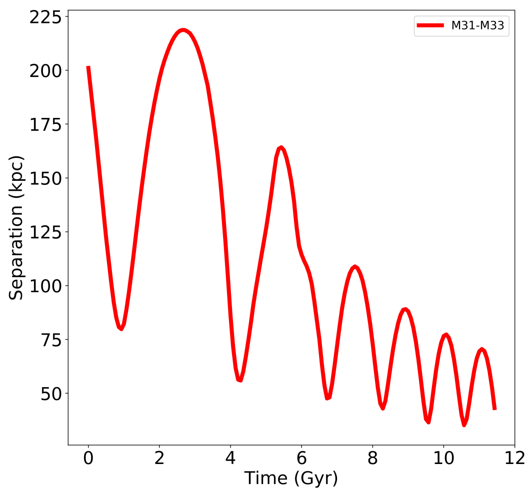

Figure 1: Example figure for M33-M31 orbit

4 Questions

1. How many close encounters will the MW and M31 experience in the future?

2. How is the time evolution of the separation and relative velocity related?

3. When do M31 and the MW merge? (you might need to zoom in on the plot - try a log y axis). What happens to M33's orbit when they merge?

4. BONUS: what is roughly the decay rate of M33's orbit after 6 Gyr (ratio of the difference between two successive apocenters and the orbital period; you don't need to be precise). If this rate is constant, how long will it take M33 to merge with the combined MW+M31 remnant if it is at a distance of 75 kpc?

5 Submit

Submit your code on GitHub with all plots. Make sure to include a screen shot of your answers to the above questions if you write them within your .py script. DO NOT include the snapshot files in your submission because it will get too large !