ASTR 660 HOMEWORK1

110060033 LI, WEI-WEN

October, 2023

Question 1

1. The particles needed is

$$\frac{500 \times (100Mpc)^3 \times 1.36 \times 10^{11} M_\odot\ Mpc^{-3}}{10^6 M_\odot} = 6.8 \times 10^{13} \approx (32768)^3$$

   which is not a technically achievable simulation judging from the lecture slide, where it stated that the highest amount of particles used in a simulation is currently below 10^{13}

2. Power et al. (2003) suggested that we choose

$$a_{max} = \frac{Gm}{\epsilon^2} < a_{min} \approx \frac{GM_{200}}{r_{200}^2}$$

   However, we do not have a simulated halo to obtained the required parameters. So I simply take the condition of the problem, stating that we should resolve 500 particles in a density of $1.36 \times 10^{11} M_\odot\ Mpc^{-3}$. We don't want any two particles have acceleration greater than the acceleration caused by 500 particles under the initial condition. Therefore, we take

$$a_{max} = \frac{Gm}{\epsilon^2} < \frac{500Gm}{R_{bound}^2}$$

   where

$$R_{bound} = \sqrt[3]{\frac{10^6 M_\odot}{1.36 \times 10^{11} M_\odot\ Mpc^{-3}}} = 1.94 \times 10^{-2} Mpc$$

   and thus we obtain $\epsilon = 3.89 \times 10^{-5} Mpc$

3. Since our simulation crosses 7 magnitude (ϵ to 100Mpc), the required precision is roughly $10^7 \approx 2^{23}$ where 23 bits is the precision bits of float32. We choose float32 although it is close to the its capacity of precision bits. Since we need 6 parameters (x, y, z and v_x, v_y, v_z) for each particle, and the memory required for each parameters is 4 bytes, we obtained the minimum memory requirement:

$$(32768)^3 \times 6 \times 4 = 2^{49} bytes \approx 256Tb$$

   Using the CPU nodes c05 to c17 of the CICA cluster for calculation, each of them provides 256Gb RAM. Therefore we need at least 1000 nodes in order to run NTHU BOX.

4. The time required would grow longer for the later stage of simulation, since TreePM code utilize Particle Mesh for long distance calculation and Tree code for short distance calculation. When the particles stack together, Tree code would have to open the cell in order to calculate neighboring particles.

5. The mass resolution of Aq-A-1 is $1.712 \times 10^3 M_\odot$ and the spatial resolution is $\epsilon = 20.5pc$ which is better than NTHU Box. The amount of particles in Aq-A-1 is 1.5 billion. Using similar calculation, we obtained the minimum memory requirement is

$$1.5 \times 10^9 \times 6 \times 4byte = 36Gb$$

   We may therefore conclude that CICA is sufficient enough to run Aq-A-1.

6. The main difference between NTHU Box and Aquarius is the amount of particles. This is caused by the method of the simulation. Even though Aquarius had a better mass and spatial resolution, the size (boundary) of each simulation is smaller. The total size of Aquarius is $(137Mpc)^3$, but it didn't calculate every halo at the same time, it zoomed in to each halo and calculate them several times. For example there are five Aq-A with different resolutions and size. The size of each halo is hence significantly smaller than that of NTHU Box.

Question 2

1. We simulated M_*, M_{cold} and M_{hot} using both numerical and analytical calculations with initial values

$$M_* = 1.0 \times 10^{10} M_\odot \ , \ M_{cold} = 1.0 \times 10^{10} M_\odot \ , \ M_{hot} = 5.0 \times 10^{10} M_\odot$$

and the adjust parameters are set as

$$\dot{M}_{cool} = 100 M_\odot \ yr^{-1}, \ V_{hot} = 500 \times kms^{-1}, \ \alpha_{hot} = 1.5$$

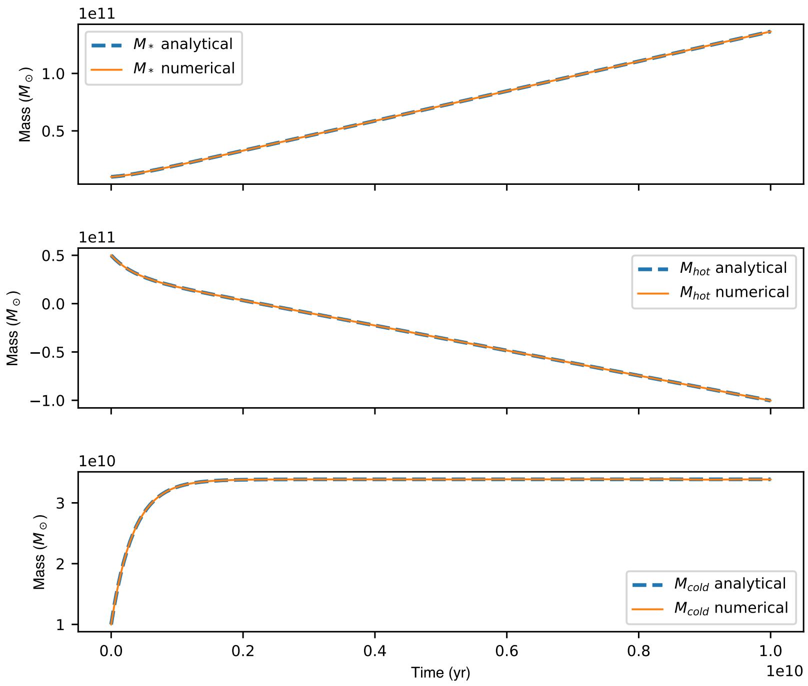

Figure 1: Comparisons of numerical and analytical solutions for different reservoirs of a galaxy. From upper to lower are M_*, M_{hot} and M_{cold}

From the results, we may observe that the numerical solution closely follows the path of analytical solution and M_{cold} is stagnant after around $10^9 yr$ while the value of M_{hot} goes to negative.

2. In order to solve the issue of M_{hot} we set the $\dot{M}_{cooling}$ to be a function of M_{hot} We set

$$\dot{M}_{cooling} = C_{cooling} M_{hot}, \ where \ C_{cooling} = 3 \times 10^{-9} yr^{-1}$$

From this we obtained the following evolution path:

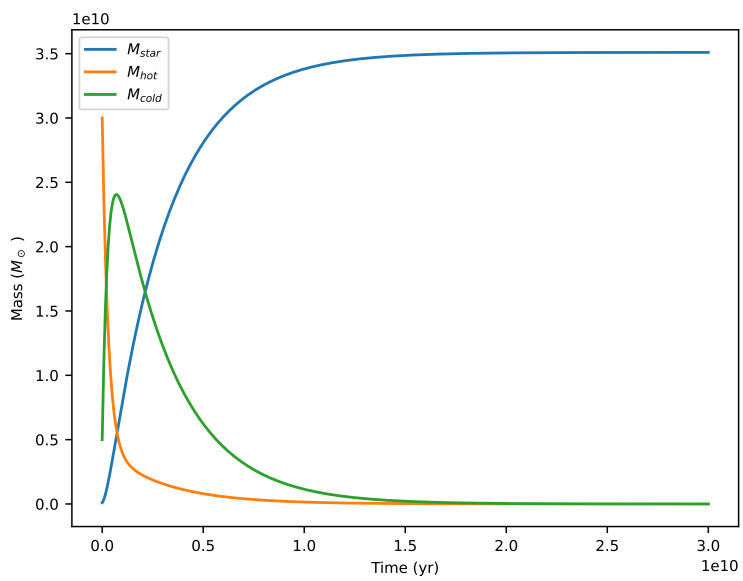

Figure 2: The evolution of M_*, M_{hot} and M_{cold}

From the figure, we may observe that M_{hot} drop rapidly initially and M_{cold} increase rapidly since they are affected by $\dot{M}_{cooling}$ which is proportional to M_{hot}. After the value of M_{cold} rises, M_* begin to rise, since star formation rate is proportional to the amount of M_{cold}. And after M_{hot} and M_{cold} are depleted, the growth of M_* stagnates.

I tried to find whether there could be a fixed point other than what we obtained. Therefore I calculated the nullspace of the following matrix A.

$$A = \begin{bmatrix} 0 & 0 & (1-R)/\tau_* \\ 0 & -C_{cooling} & \beta/\tau_* \\ 0 & C_{cooling} & -(1-R+\beta)/\tau_* \end{bmatrix}, N(A) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

It turned out that the only fixed point is the same as what we obtained. Therefore we may infer that the galaxy tend to consume all its reservoir of M_{hot} and M_{cold} and turn them into stars.

However, another questions is whether this fixed point is really stable? I then followed the approach given in **Matplotlib: lotka volterra tutorial** and constructed the Jacobian matrix and noticed that the Jacobian is the same with A. I then calculate the eigenvalue of A, resulting with

$$\lambda_1 = 0, \lambda_2 = -3.38 \times 10^{-9} \; and \; \lambda_3 = -3.39 \times 10^{-10}$$

Since there is no imaginary value, it should not oscillate, but follows a exponential decreasing path.