Computational Methods HW- 5

Aasim Z Jan (aj3008@g.rit.edu)

October 13, 2020

https://github.com/AasimZahoor/Comp_methods.git

Question 1.

Given

This problem asks to evolve the system of 655 galaxies in a box of 10*10 Mpc with time. We have been given two files; *galaxies0.npy* and *galaxies1.npy*; which contain the x and y coordinate of the galaxies separated by a time of 1000 years. We have been given the freedom to choose our own time step.

Approach

For this problem I made three functions.

- scan(data,limits):

  This function scans for the points in the given quadrant.

  Parameters-
  data : the points you want to get scanned,
  limits : Limits of the quadrant i.e a<x<b and c<y<d
  Returns-
  j= number of points in the quadrant,
  newdata= the points in the given quadrant

- com(data):

  This function finds the Centre of mass for the given points

  Parameters-
  data : Points you want to find centre of mass for. Usually it is an output of scan function

  Returns-
  comx= x position of COM
  comy= y position of COM
  n= number of points for which COM is being calculated. I am taking n and not mass since all galaxies have same mass

- accel(point1,point2):
  This function gives the acceleration on point1 due to point2. Force smoothing has been added as shown in paper Dehnen 2001.

  Parameters-
  point1 : point1 coordinates (x,y).
  point2 : point2 coordinates and number of points it represents if it is a COM.(x,y,n). You need n i.e the number of point it represents to find the mass of point two when finding acceleration of point1 due to point 2

  Returns-
  Xacceleration, Yacceleration.

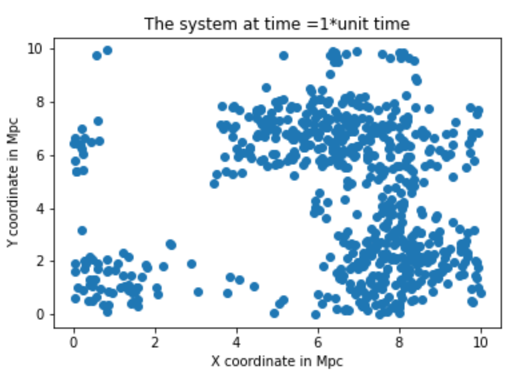
So after defining the functions I defined the Barnes-hut algorithm. To define the algorithm I used:

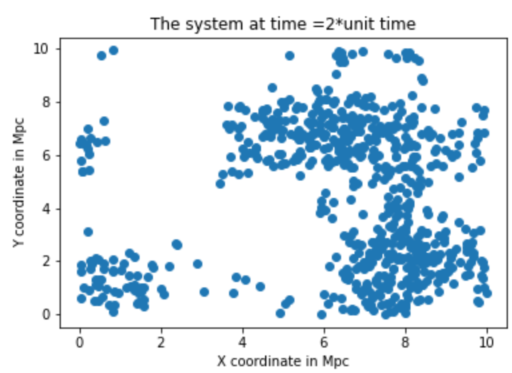
- **while loop** - This loop is for time, it goes on till it reaches the end of time defined by us
- **for loop-** For each time, this for loop focuses on each of the 655 galaxies
- **while loop and four pairs of if-else statements-** the while loop has "i" as a switch so that when "i" takes a value of 1, it turns off and allows the for loop to move on to next galaxy. i=1 only when the galaxy in question is the only galaxy in a quadrant.
  The four pair of if-else statements check if the point(galaxy) in question is in one of 4 quadrant. If it isn't, else statements finds out the acceleration of the galaxy in question due to galaxies in that quadrant. If the galaxy is in the quadrant then if statements focuses on that quadrant and scans the quadrant if there are more than one galaxies. If there are then it divides it again into 4 quadrants else, it allows the main for loop to move on to next galaxy.
  Also, l acts as a switch so that when l=> 4, the algorithm switches from finding COM to calculating forces between individual galaxies.

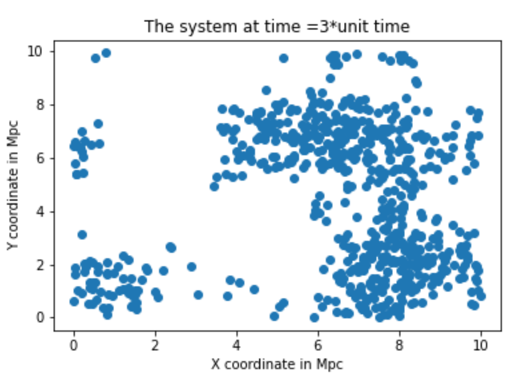
What I am doing here is making my code go through each of the 655 galaxies, for each galaxy find the quadrant it is in and keep on dividing that quadrant till it is the only galaxy in that quadrant. Each time there will be 3 quadrants where it is not present, so the code either finds COM and calculate acceleration or finds acceleration due to individual acceleration if the 3 quadrants are too close the galaxy in question.After finding the acceleration we use the Verlet symplectic integrator to find the position for the next time step.

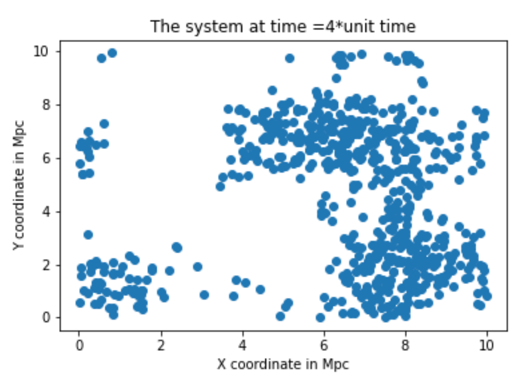
$$x_{i+1} = 2 * x_i - x_{i-1} + h^2 * a_i$$

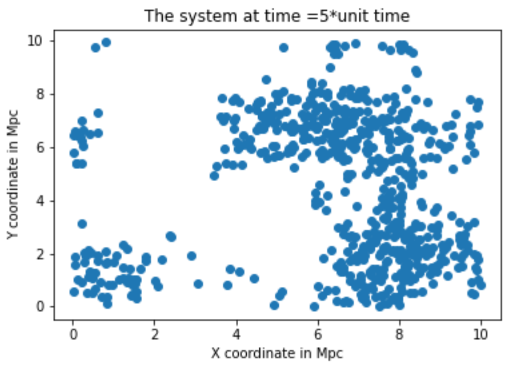
This goes on for each of the 655 galaxy. This code ends when it reach the end of the time period.
Results

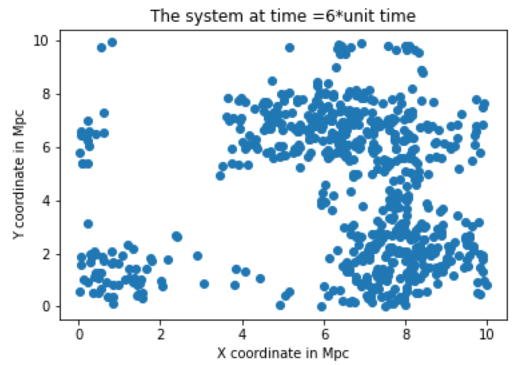


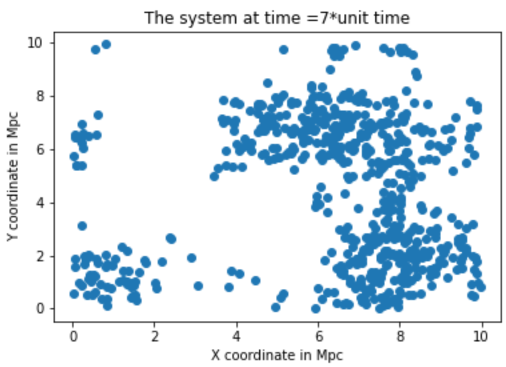


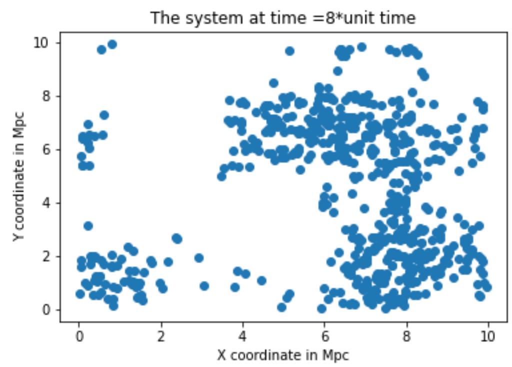


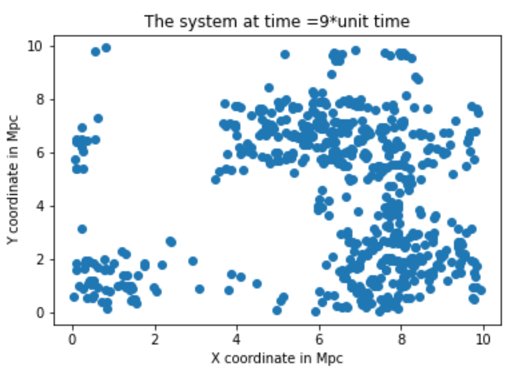


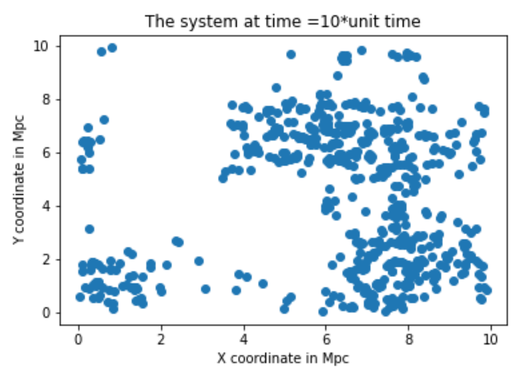












Here the unit time is 40000000 years. If you look at bottom right and bottom left corner, galaxies come together over time. The change is more visible if you look at the .png files in the Images folder.
I had run the code at larger time steps and many galaxies "merged", however I wasn't sure taking such big time steps and lowkey assuming constant gravitational force is right, so I took the smallest time step where the change is visible.

Question 2.

Given

This problem asked to find out potential at different points in a 10*10 Mpc box. The box has 655 galaxies, 400 uniformly distributed in a disk of radius 3Mpc centered at (7,7) and 255 uniformly distributes in a dish of radius 2Mpc centered at (8,2). The boundary condition is that potential at boundaries is zero.

Approach

I created a mesh with a step size of 0.1 Mpc in both x and y direction. I calculated potential at each point and plotted it.
To find out the potential I divided the code in three if statements to find out density:

- one which checks if it is inside the disk at (7,7) and gives out density $= (455 * 10^{12})/(pi * 9)$ if it is.
- one which checks if it is inside the disk at (8,2) and gives out density $= (200 * 10^{12})/(pi * 4)$ if it is.
- one which check if it outside both disks and gives out density to be zero.

These checks are done by using the inequality

$$(x-a)^2 + (y-b)^2 <= r^2$$

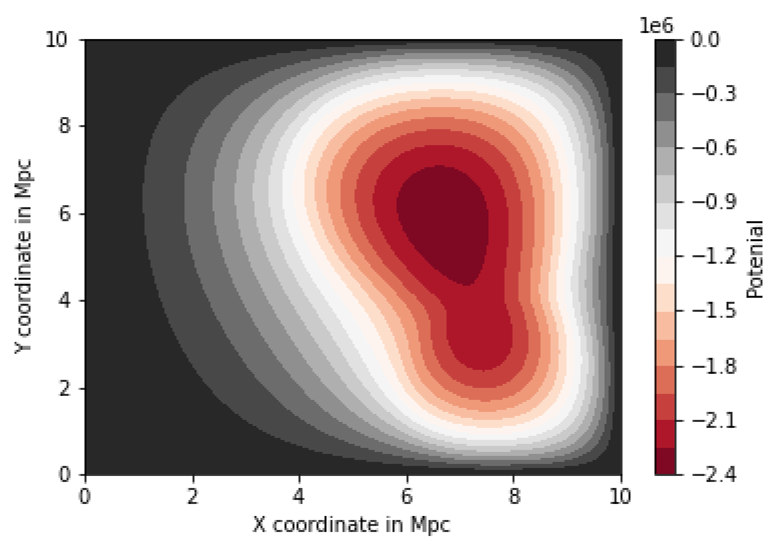
After finding the densities I used the three point method to evaluate the Laplacian and found out the value of potential at each point iteratively using Successive Over Relaxation method (SOR).
The equation used are:

$$R_{ij}^k = 0.25 * (p_{[]i+1][j]}^k + p_{[i-1][j]}^k + p_{[i][j-1]}^k + p_{[i][j+1]}^k - h^2 * 4 * pi * G * density) - p_{[i][j]}^k$$

$$p_{[i][j]}^{k+1} = R_{[i][j]}^k + p_{[i][j]}^k$$

Here, p is an array of potentials, G is the Gravitational constant, R is the residual array and density is found at the centre of cell.

Results

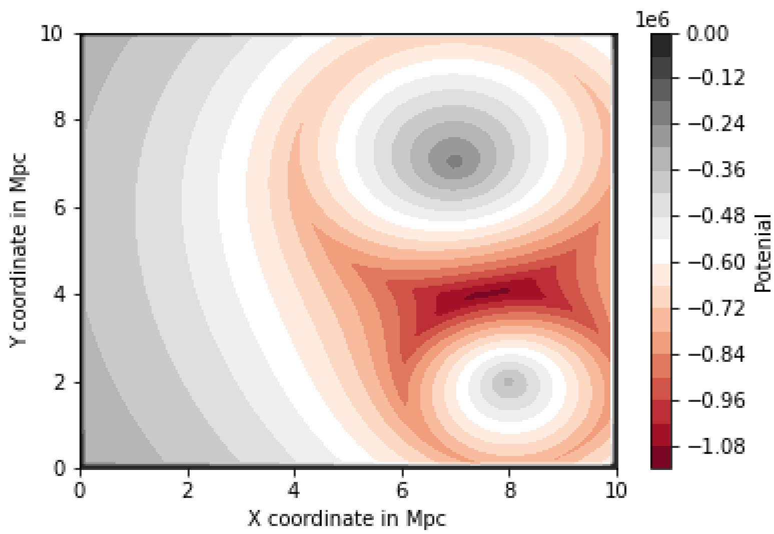


This figure shows the potential due to two disks using SOR.

Earlier approach(Just kept to show how analytical solution works)

To find out the potential I divided the code in three if statements to find out tje location of the point and hence potential:

- one which checks if it is inside the disk at (7,7) and finds potential if it is. If it was, I considered the disk at (8,2) to be a point particle of mass of 255 galaxies situated at (8,2) and found the mass inside the disk $(x-7)^2+(y-7)^2 <= r^2$ where r is the distance from the (7,7) to the point. The mass between r and 3 has no contribution due to Gauss law and is ignored.
- one which checks if it is inside the disk at (8,2) and finds potential if it is. If it was, I considered the disk at (7,7) to be a point particle of mass of 400 galaxies situated at (7,7) and found the mass inside the disk $(x-8)^2+(y-8)^2 <= r^2$ where r is the distance between (8,2) and the point. The mass between r and 2 has no contribution due to Gauss law and is ignored.
- one which checks if the point is outside the two disks and finds potential if it is. In this case I considered the disks to to be two point particles, one with mass of 255 galaxies at (8,2) and one mass of 400 galaxies at (7,7).



This figure shows the potential due to two disks analytically. However, the boundary conditions were superficially imposed. So the code isn't exactly right for this case. The code for that is in pb2m if you are interested