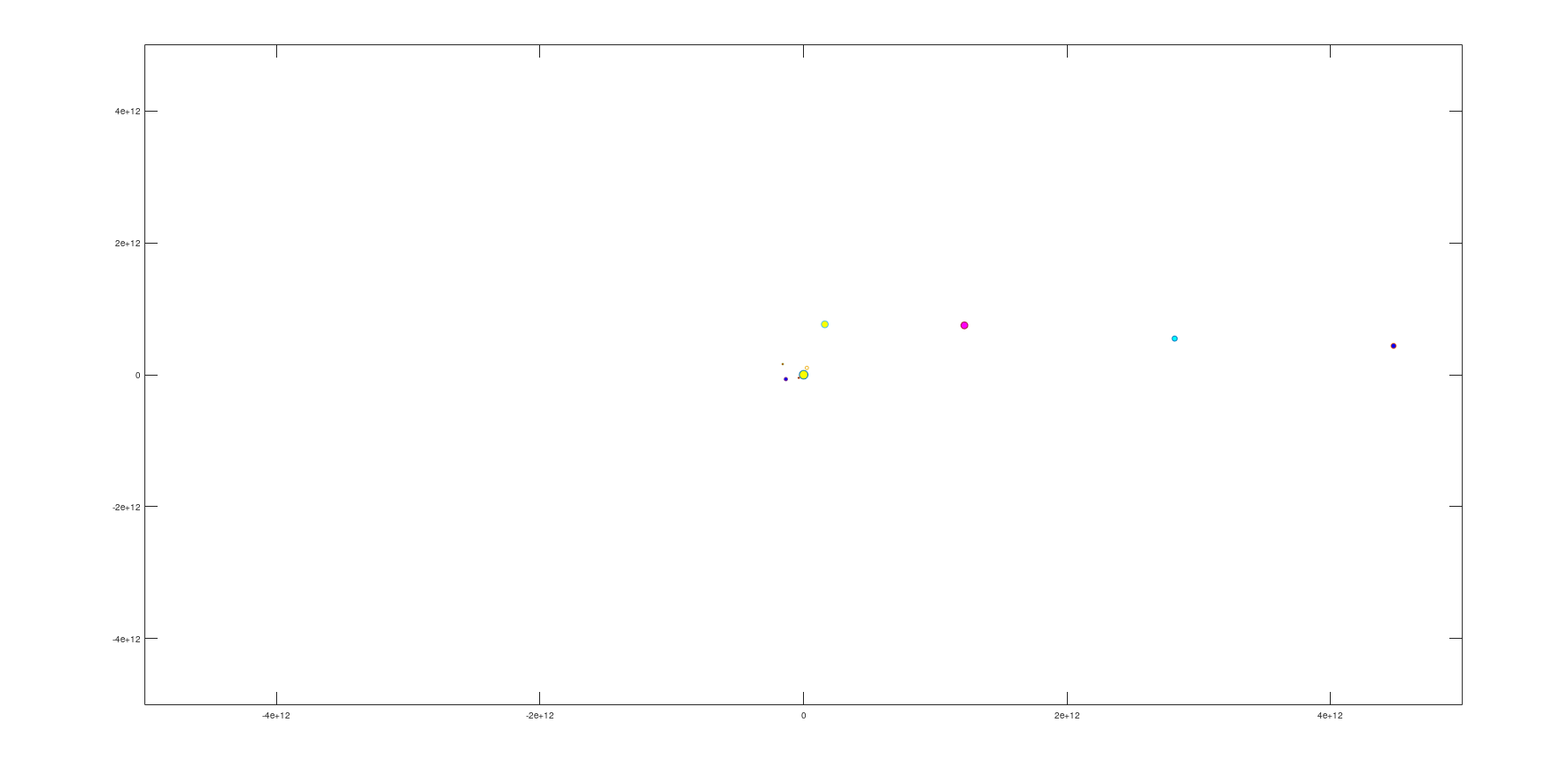
I wrote code to animate the solar system (the 8 planets and the sun) using the gravitational forces they apply on each other. When 2 planets collide with each other (or a planet with the sun) they cause a non-elastic collision.



The gravitational forces were calculated using the formula:

Where m is the mass of the planet and r is the distance between the two planets (or the sun).

The distances between the planets were calculated using the formula:

Where x1, x2, y1, y2 are the positions of the two planets.

When the elastic collisions occurred between 2 entities, their masses were added and their new velocities were calculated using the formula:

Where VP is the velocity of the planet and M is the mass. This is done for both of the velocities in the x and y direction.

**Animation:**

For the animation, each iteration in the for-loop is equivalent to 1 day on earth-time.

Firstly, values inside an nxn (where n is the number of entities that have not collided) matrix for the distances between each planet and the sun was created using the distance formula (2). After that, an nxn matrix for the angles between the planets was calculated using the atan2 function in Octave.

The forces from each entity onto the other were calculated and stored inside an nxn matrix using formula (1). These forces were used to calculate their projected values on the x-axis and on the y-axis using these formulas:

These directional forces were then multiplied by a column vector with the size of nx1. The purpose of this matrix multiplication is to add all of the forces on one entity on the x-axis and on the y-axis (to get the resultant forces).

These directional forces were used to calculate the velocities of the entities in the x and y direction using this formula:

These velocities were stored in an nx2 matrix, where each column corresponds to the velocity in the x and the y direction for that particular entity. Next, the same thing was done to calculate the distance covered using these calculated velocities using the formula:

The distance in the x-direction and the y-direction corresponded to the entity’s x and y coordinates. Finally, these coordinates were used to animate this solar system.

In the case of collisions, an if-statement was used to check if the distances between the entities were smaller than their combined radii. This was checked after the distances were calculated. If two entities got too close to each other, then we assumed that a non-elastic collision has occurred between them. The masses and the velocities of the 2 entities were combines using equation (3). The rest of the matrixes were modified by removing an entire row or column for one of the entities that collided (the other pace was used to keep the new values.

This triggered 2 break statements which would have made us repeat the calculations again with the new conditions. This if-statement was placed after the distances were calculated so that the values of the other matrixes would not be changed. All the conditions would have been the same except for the change done for the two planets that collided with each other.

Other matrixes were used for keeping properties of the planets and some constants such as the color of the planets, their sizes, their distance from the sun, their velocities, and the planet’s radius.

4 test runs were included in the zip folder with different starting conditions.

1. Normal/ real conditions.
2. To show how collisions look like when they happen.
3. To show how the first 4 planets interact with the sun if they got really close to it.
4. To show how the first 3 planets would interact with the sun if they were close to the sun but not as close to trial no. 3 (Similar interaction to what would happen if alpha particles were shot onto a thin gold foil) and the forth planet heading directly towards the sun.