Object Oriented Simulation of the Solar System

Colin, Spencer and Kristen[1]

[1]*Department of Physics and Astronomy, Michigan State University*

We present a beautiful model of the Solar system. This model utilizes algorithms that numerically solve for the orbits of the planets based on the gravitational forces between all the main bodies of mass in the system. Those bodies include the sun and all eight planets. We hope to find the orbits stable. otherwise we are in for a surprise.

INTRODUCTION

What we are trying to do here is to numerically create Issac Newton's model of the solar system. Way back in newton's day he only could get the two body problem to work. a hundred years later, perturbation theory was formulated which gives a better approximation for the system. now with computers we can use high level precision algorithms to calculate a near exact solution to the orbits.

THEORY

Basic Theory

For this project we are using newton's law of gravity on the planets. To calculate the acceleration, we take one planet and calculate the force between the other planets and the planet of interest (in this equation below object j is the interest).

$$\sum_{i\neq j} \frac{-GM_iM_j}{(r_i - r_j)^3}(\overrightarrow{r_i} - \overrightarrow{r_j}) = M_j \frac{d^2\overrightarrow{r_j}}{dt^2} \tag{1}$$

.

METHODS

Differential Solver

So with initial conditions given to us by NASA we can then perform an integration method to solve this second order ordinary differential equation. The method of integration we choose to use is the Verlet method. For this algorithm, we use the planet's initial velocity, position and acceleration to calculate the new position. After the new position is calculated we then calculate the new acceleration. We calculate the new acceleration before the velocity since in the Verlet method uses an average acceleration to determine the new velocity. The algorithm is written out as:

$$\overrightarrow{x}_{i+1} = \overrightarrow{x_i} + h\overrightarrow{v_i} + \frac{h^2}{2}\overrightarrow{a_i} \tag{2}$$

$$\overrightarrow{v}_{i+1} = \overrightarrow{v_i} + \frac{h}{2}(\overrightarrow{a}_{i+1} + \overrightarrow{a_i}) \tag{3}$$

where $\overrightarrow{x_i}$ vector is the discretized position vector like $\overrightarrow{r}$ above. $\overrightarrow{a_i}$ vector is essentially the $\frac{d^2\overrightarrow{r}}{dt^2}$ discretized as well corresponding to the given $\overrightarrow{x_i}$. $\overrightarrow{v_i}$ is our velocity calculation. Finally or step size in time is h. Overall this method would give us a more accurate result than the rather trivial Euler algorithm we also used to test the conditions.

Scaling

So the initial conditions given by NASA are in the Units of AU for distance and AU per a day for velocity. So we want to use natural units of the system. the natural units are AU and years. They are the most natural of units since $GM_{sun} = 4\pi^2 a^3/T^2$ which a is the semimajor axis and T is the period. For Earth, $a = 1AU$ and $T = 1$ year. We listed the planets in Earth masses initially. But normalized for the suns mass in the calculation for the force in natural units. of course we also had to translate the velocity given in AU per a day to AU per year by multiplying the velocity by 365.

Testing

To test the algorithm, we would look at the plots to see if they are elliptical as they should be by Kepler's first law. For more theoretical tests, we are going to measure the center of mass, total energy and total angular momentum. The center of mass should have a constant speed since newton's third law would require total momentum to be conserved. total energy should be conserved since the Lagrangian that represents the system has no time dependent terms. So the symmetry with time conserves the energy. The total angular momentum should also be conserved since all forces are directed from one object to the other. Therefore each fore component should cancel the other's torque.

Object Orientation

The Euler and Verlet methods are simple models that can be used to construct short scripts for dynamical problems. However, for more sophisticated simulations of N > 2 bodies it becomes advantageous to object orient the code. The most important object in our project were the planets themselves; constructing a planet class allowed separate instances to keep track of individual information.

```
class planet():
    def __init__(self,name):
        self.mass = 0
        self.name = name
        self.pos = np.array([0,0,0])
        self.pos_old = np.array([0,0,0])

        self.vel = np.array([0,0,0])
        self.vel_old = np.array([0,0,0])

        self.acc = np.array([0,0,0])
        self.acc_old = np.array([0,0,0])

        self.KE = 0
        self.PE = 0
        self.L = np.array([0,0,0])
        open('planet_' + self.name, 'w+').close
```

FIG. 1. Planet class object with initialization of $\overrightarrow{r}_{i+1}$ and $\overrightarrow{r}_i$, etc and creation of a data file for quantities of motion.

```
def ssverlet(V0 = 2*np.pi, years = 1):
    N = 365 * years
    dt = 1/365
    planets = make_planets()
    acc_update(planets)
    for t in np.arange(N):
        for i in np.arange(len(planets)):
            planets[i].update()

            planets[i].pos = (
                planets[i].pos_old
                + planets[i].vel_old * dt
                + 1/2 * dt**2 * planets[i].acc_old
                - planets[0].pos
                )

            acc_update(
                planets,
                planet=planets[i].name
                )

            planets[i].vel = (
                planets[i].vel_old
                + 1/2*(
                    planets[i].acc_old
                    + planets[i].acc) * dt
                - planets[0].vel
                )

            if (t % N/10) % 1 == 0:
                planets[i].output()
            else:
                pass
```

FIG. 2. Verlet method taking into account the motion of the sun. The calculation of velocity is dependant on $\overrightarrow{a}_{i+1}$ and $\overrightarrow{a}_i$, therefore the calculation of $\overrightarrow{a}$ must occur prior to the calculation of $\overrightarrow{v}_{i+1}$

RESULTS AND DISCUSSIONS

Here is our plot for the solar system around the sun.

As you can see here that the the orbits are very near elliptical, as expected by Kepler's first law. At least our model shows that the orbits are rather stable and we don't go crashing into the sun or something. For This plot we subtracted the suns position from the other positions. This gave us the coordinates of all planets with respect to the sun. Since the center of mass should be close to the suns radius. Then our center of mass should be near stationary relative to the system which means it is conserved.

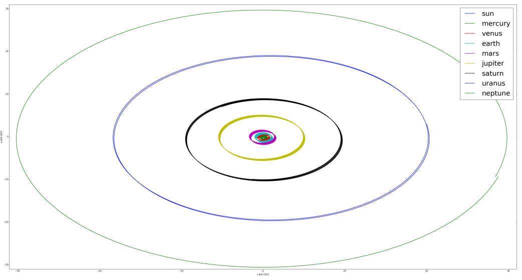

FIG. 3. Solar system simulation over 165 years. When the position of the sun is allowed to be affected by the planets we find a small wobble in the orbit of the solar system.

Center of Mass

As we can see in this graph. that the center of mass precesses around the sun. which is expected. since the center of mass is about $\frac{1}{100}$ AU then our picture above has only an error of .01 AU. which is not much considering that the Earth and most planets are at least an AU away.

Angular Momentum

Figure 5 we have the plots of the angular momentum. Based on the theory the angular momentum calculated

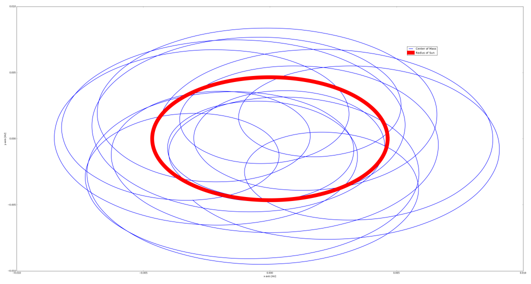

FIG. 4. Motion of the center of mass for the solar system. The simulation was preformed over 165 years, enough time for Neptune to complete a full orbit. red is the sun's radius, blue is the Center of mass position.

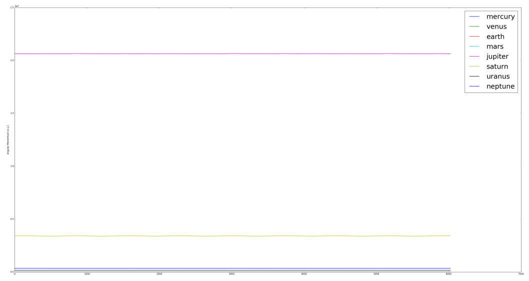

FIG. 5. angular momentum over 165 years. With the simulation performed in the reference frame of the sun the next largest contribution to angular momentum of the system is Jupiter. We find the magnitude of the angular momentum remains bounded and appears to be a constant over a long period.

should be equivalent to the angular momentum around the center of mass of the system, since Center of mass should be a constant of motion. We just calculated given $\vec{x}_i \times M_i \vec{v}_i$ for each planet. The congregate angular momentum is conserved and can be seen on the graph by the fact that the angular momentum of each planet is near conserved. This also confirms the accuracy of Kepler's second law. Kepler's second law is originally written such that the area swept from the planet to the sun in a given time is the conserved. Since the mass of the planets don't change, then Kepler's second law and angular momentum conservation are the same.

Energy

In Figure 6 we have the plots of the energies. As one can see here that for at least the small mass planets that the energies are nearly conserved. For the larger planets there is a bit more variance. We suspect that the variance is due to the fact that the energies are measured in the non inertial frame of the sun. This is not really that big of a deal for what we are testing though since that would imply a periodic energy variance. We can see from the plots that there is definitely a periodic component so we can safely assume that the total energy is conserved.

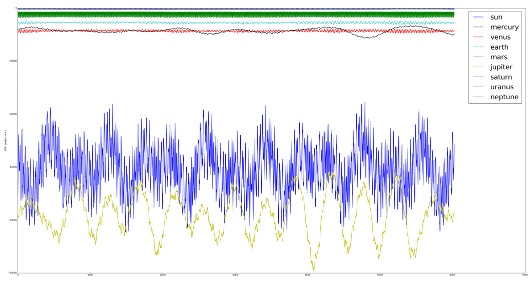

FIG. 6. We see the oscillation in total energy over a 165 year simulation. Periodic oscillations in the energy are expected due to numerical error on the order of dt^2. For bounded orbits the total energy of a planet is expected to be negative.

CONCLUSIONS

Our algorithm does have some error. The error on the position algorithm is proportional to the step size cubed. Which means it has an integration error proportional to h^2. It makes sense that is a little bit of error for the orbits. For the most part our predictions for conservation's seem to be consistent with our results discussed in "Results and Discussion" section. Due to the angular momentum conservation, elliptical orbits and the number of orbits follow Kepler's third law. We can honestly say we have rediscovered Kepler's laws of planetary motion.

REFERENCES

[1] Morten Hjorth-Jensen. *Computational Physics Lecture Notes.* 2015.
[2] Horizons ephemerides.