May 2016

Simulation of n-Bodies Through Newton's Universal Gravitation

R. Davis

Fairfield University

Abstract

Using Euler-Cromer Method, we are trying to simulate n-body motion around a center of mass utilizing Newton's Law of Gravitation as well as Newton's Three Laws of Motion. By changing the the Δt within the function, as well as the mass and the initial velocity, we can predict the behaviors of n-bodies around a center of mass. We can also vary the number of bodies and see how it effects the system as well investigate, at what velocities the planet could escape from its orbit. We will also investigate, what would happen if we introduced a black hole like object, utilizing Newton's Equations.

I. Introduction

Newton's three laws state that:

1. Every object in a state of uniform motion tends to remain in that state of motion unless an external force is applied.
2. The relationship between an objects mass, m, and its acceleration a, and the applied force is $F = m\frac{d^2x}{dt^2}$. Acceleration and the forces are vectors; in this law the direction of the force vector is the same as the direction of the acceleration vector.
3. For every action there is an equal and opposite reaction.

Given the three laws we can then figure that every body that we study is going to exert its own force on the other bodies present. Given an initial set of conditions for every body in question, we can then determine the forces on the bodies as well as the motions relative to each other. Using the Euler-Cromer Method, we use can use Newton's Universal Law of Gravitation the needed initial accelerations, from which we can eventually derive velocity and position.

Newton's Second Law is given by:

$$F = m\frac{d^2x}{dt^2} \ (1)$$

Newton's Law of Universal Gravitation is given by:

$$F_G = -G\frac{m_1 m_2}{|r^2|}\hat{r} \ (2)$$

Where G[1] is Newton's Gravitational Constant, and r is the distance between the

[1] $G = 6.67 * 10^{-11} m^3 kg^{-1} s^{-2}$

two bodies in question. By setting the two forces equal we then get:

$$\frac{d^2x}{dt^2} = -\frac{Gm_1}{|r^2|}\hat{r} \; (3)$$

Since one of the masses does cancel out, because the mass does not effect the force on itself, we are left with an equation that can predict the acceleration in question for the given planet.

The difference between a single body motion and a n-body simulation comes down that, every other body effects the other. Therefore, during every calculation, you have to recalculate the position vector for every individual planet, to determine the proper force being exerted on the other planet. For this we use:

$$r_1 = \sqrt{\Delta x_1^2 + \Delta y_1^2} \; (4.1)$$

$$r_2 = \sqrt{\Delta x_2^2 + \Delta y_2^2} \; (4.2)$$

$$r_{12} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \; (4.3)$$

This allows us to recalculate the vectors for each planet at every point in time given a certain Δt in time.

Using the Euler-Cromer Method we calculate the accelerations of the individual planets as well as their velocity and position using the following equations:

$$a_{x(n+1)} = \sum -G\frac{m_n}{|r_{nn+1}|}(x_{n+1} - x_n) \; (5.1)$$

$$a_{y(n+1)} = \sum -G\frac{m_n}{|r_{nn+1}|}(y_{n+1} - y_n) \; (5.2)$$

$$v_{x(i+1)} = v_{x(i)} + \frac{d^2x_{i+1}}{dt^2}\Delta t \; (5.3)$$

$$v_{y(i+1)} = v_{y(i)} + \frac{d^2y_{i+1}}{dt^2}\Delta t \; (5.4)$$

$$x_{i+1} = x_i + \frac{dx_{i+1}}{dt}\Delta t \; (5.5)$$

$$y_{i+1} = x_i + \frac{dy_{i+1}}{dt}\Delta t \; (5.6)$$

The reason for using the Euler-Cromer method and not the plain Euler method, is because the Euler method does not compensate for the constant gain in acceleration. As a result the body in question keeps gaining energy rather than looking at the energy it has already gained from its motion from the previous step, it keeps spiraling out away from the center of mass.

The initial values of the planets can be created using a random number generator. However, we have to make sure to keep consistent with units. As a results we will be using Astronomical Units, AU[2], and Solar Masses, $M_{\odot}$[3], to make the calculations much easier and scalable to our simulation.

The reason we want to randomize the initial conditions is because it is very difficult to accurately pin point where a certain planet will be. However, based on Newton's Law of Universal Gravitation, all the planets will eventually attain equilibrium in the system based on the input results we got.

[2] $1\,AU = 149.6 * 10^6\,km$

[3] $M_{\odot} = 1.988 * 10^{30} kg$

II. Numerical Method

In order to use, the Euler Method to calculate our function we must first be given a set of initial conditions:

$$y' = f(x, y)$$
$$y(x_0) = y_0$$

From this, we also decide on an interval, and depending on how many iterations we would like to step through, chop it up into even smaller pieces. The rest of the solutions are found, through iterations of those steps, with the equations below.

The origin of Euler Method can be seen from the original Newton's method which is finding the root in the vicinity using approximations of an ϵ. The approximation can be made setting $x = x_0 + \epsilon$ where ϵ is the Δt which is how close do you want to get to the approximation and x_0 is your initial condition which you calculated based on a set of given equations.

Knowing those two facts we can take a look at the Taylor Expansion which is described by:

$$f(x_0 + \epsilon) = f(x_0) + f'(x_0)\epsilon +$$
$$\frac{1}{2}f''(x_0)\epsilon^2 + \cdots + \frac{f^n(x_0)}{n!}\epsilon^n$$

By taking the first two terms we can approximate what the next term will be, and iterating that over and over till, we can get a close enough approximation.

The function would look something like this:

$$f(x_0 + \epsilon) \approx f(x_0) + f'(x_0)\epsilon$$

Our four equations as described by the Euler Method which we used were:

$$a_{x(n+1)} = \sum -G\frac{m_n}{|r_{nn+1}|}(x_{n+1} - x_n)$$

$$a_{y(n+1)} = \sum -G\frac{m_n}{|r_{nn+1}|}(y_{n+1} - y_n)$$

$$v_{x(i+1)} = v_{x(i)} + \frac{d^2x_i}{dt^2}\Delta t$$

$$v_{y(i+1)} = v_{y(i)} + \frac{d^2y_i}{dt^2}\Delta t$$

$$x_{i+1} = x_i + \frac{dx_i}{dt}\Delta t$$

$$y_{i+1} = x_i + \frac{dy_i}{dt}\Delta t$$

As previously discussed, the reason we do not want to just use the Euler Method, is because it tends toward infinite energy for the system, therefore the body in question spirals out from our origin. This is corrected through the Euler-Cromer method which calculates the values ahead by one, making sure the system does not gain that energy, but retains the energy already in it. Therefore, to compensate for that we use the previously mentioned equations for our time step. Where our index i is $i + 1$ for values being calculated for the future velocity and position.

In junction with the Newton-Cromer Method, I also applied some random numbers to the applications as well as calculating based on actual distances given from the table.

To better represent the first three planets of our Solar System, as well as the Sun and the Moon.

III. Calculations | Observations

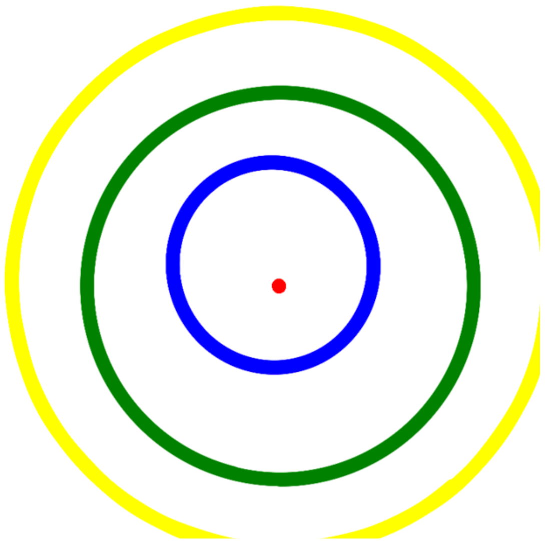
To be more successful with the data initially I did a randomized set of numbers for each of the bodies observed. However, we wanted to investigate also our solar system on top of the originally given results.

Using the sun as our standard here are the values we used. Unless otherwise noted, all the following results will be using a $\Delta t = 0.005$.

Table 1:

Planet	Mass	X	Y	V_X	V_Y
Sun	1	.003	.001	4.9E-4	0.002
Mercury	1.659E-7	-0.34	-.27	4.25	-7.61
Venus	2.447E-6	-0.16	0.699	-7.2	-1.77
Earth	3.00E-6	0.65	-4.85	-4.85	4.09
Moon	3.69E-8	0.65	0.75	-4.7	3.98

When we plotted the Inner Solar System this is what we got:



Where the red denotes the Sun, the blue, Mercury, the green Venus, and the Yellow Earth.

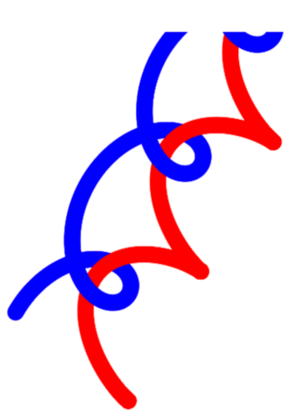
The sun also received an initial velocity. This is because the sun is also plunging through space at the same time as the planets are orbiting. However, because the time step was at about $\Delta t = 0.0005$ it would take a while for us to notice the sun moving

Table 2:

Planet	Mass	X	Y	V_X	V_Y
Body 1	1	.003	.001	4.9E-4	0.002
Body 2	1	-0.34	-.27	4.25	-7.61

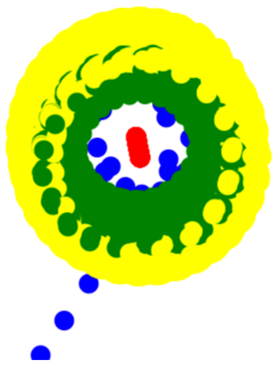
If we changed the values and we made one of the planets the same mass as the sun, and got rid of the rest, we were able to create a binary system.

The following describes a binary star system:



In this image, we put two bodies at equal mass, however, because Body 2 has much more significant velocity in the X and Y direction, it will take over and Body 1 will follow it.

Now, if we changed the time step, $\Delta t = 0.1$, we can observe the Sun moving away from the center and hurtling through space:



However, we do run into a problem by changing the Δt. The problem because the higher the value the less accurate the motion of the planets is. To get the best precision we want to ensure the lowest value possible by the simulation. In this image we can see that the sun is traveling in the given direction, according to Table 1, initial conditions, the planets seem to be following. However, what is curious to note is that because Mercury has such a close orbit, and the inaccuracy of the values miscalculated Mercury's orbit, its trajectory was set off and it was able to escape the sun's gravitational pull. This we call the escape velocity.

The second topic we would want to calculate for such a system is the escape velocity of the object.

The escape velocity can be derived from Newton's Universal Law of Gravitation as applied to potential energy which is equal to:

$$V = G\frac{m}{r}$$

We can set that equal to our kinetic energy equation which requires velocity:

$$T = \frac{1}{2}m\left(\frac{dx}{dt}\right)^2$$

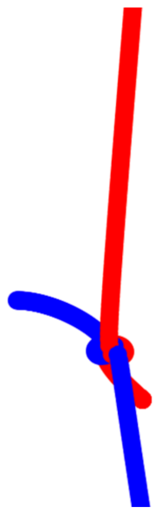
By setting these two equations equal to each other we can figure out what kind of energy the planet would need to escape the orbit of the Sun:

$$\frac{dx}{dt} = \sqrt{\frac{2Gm}{r}}$$

In our case we will repeat the binary body system, and using the equation we will calculate the required velocity for the planet to escape.

Table 3:

Planet	Mass	X	Y	V_X	V_Y
Body 1	1	0	0	0	0
Body 2	1	-0.34	-.27	5.0	0

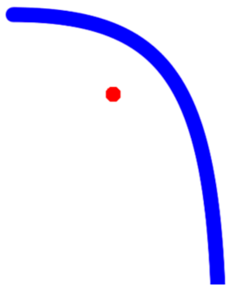


In this scenario, both bodies interact with each other, and because the masses are equal they produce trajectories off each other.

Now if we take the very same two bodies and change the masses, and increase the velocity to 15 AU/yr we can allow one of the bodies to fly off into the distance, that would be what we call the escape velocity.

Table 4:

Planet	Mass	X	Y	V_X	V_Y
Body 1	1	0	0	0	0
Body 2	0.1e-5	-0.34	-.27	15.0	0

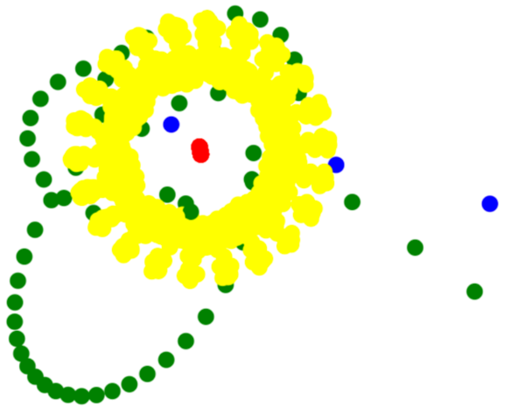


We can conclude if the two masses have equal mass, then they will form a binary star system, if they each have velocities. However, if one is stationary and the other is moving, one will project the other into the opposite direction as can be seen from Table 3 results.

On the other hand, if one mass is much greater than the other, and is stationary, and we give the other the scape velocity, it will attempt to orbit the larger mass and instead fly off into the distance.

We also want to investigate, how changing the Δt would effect the accuracy of the orbits of the bodies.

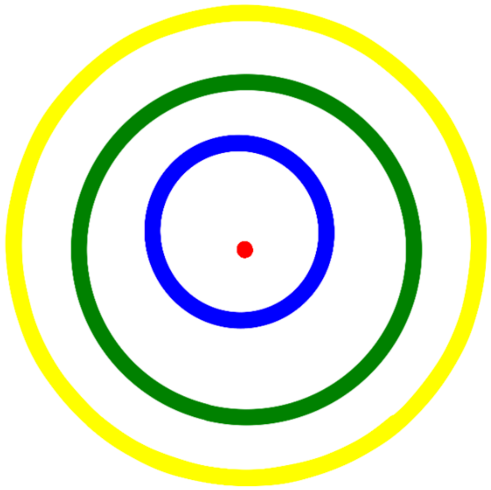
By changing $\Delta t = 0.1$, and maintain the values from Table 1, we were able to determine an interesting observation:



What the above image shows with the time step changed, is the inaccuracy of the planets as they rotate around the center of mass. Where you have both, Mercury and Venus, blue and green respectively, flying off into the distance and escaping the orbit of the large mass, the Sun. While Earth and moon remain in orbit, however, there is no clear orbit.

The orbits that we observe are very spotty since we cannot accurately predict where the planet will be in the next position since the time steps are so large.

Now, if we changed the time step to $\Delta t = 0.001$, we would be able to attain a much better accurate representation of where the planets would be, still using the values of Table 1:

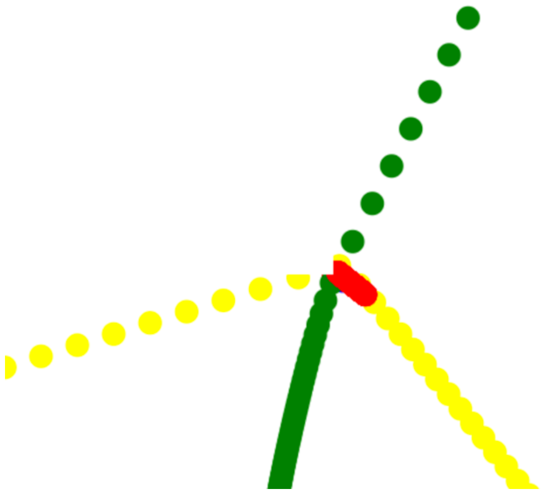


As we observe, this image looks almost the same as the image from Table 1. That is because with the proper precision of the time step we can achieve a decently accurate representation of our inner solar system.

There was one more instance which I wanted to observe, is what if we made the sun, super massive, Black Hole, massive, and made the planets into heavy stars, of three times the mass of the sun:

Table 5:

Body	Mass	X	Y	V_X	V_Y
Black Hole	1e3	.003	.001	4.9E-4	0.002
Star 1	1.659	-0.34	-.27	4.25	-7.61
Star 2	2.447	-0.16	0.699	-7.2	-1.77
Star 3	3.00	0.65	-4.85	-4.85	4.09
Star 4	3.69	0.65	0.75	-4.7	3.98



What makes this interesting is you can tell how the planets converge to the single mass, representing the black hole, which is in red.

The dots trailing after are the point, after the star was consumed by the black hole. However, Newton's equations do not predict for black holes, but it is an interesting look at how it would function if a black hole did actually exist in the space.

What the dots actually end up telling us that the body got propelled at very high speeds away from the black hole. However, from our knowledge of black holes, we know the body would be consumed by the high gravity of the black hole.

The very same method that we used for the five body problem, we can also apply to any body problem. This would require us to change our code, to accommodate for those other bodies, as well as scaling to represent all the bodies present.

IV. Analysis

The problem with simulating this sort of motion is that Newton's Laws do a good job, but not the greatest. Since the discovery of General Relativity, it would be the standard to be able to simulate this using Einstein's equations since they also accommodate for curvature of space-time.

The other issue that we can foresee quite easily here, is the time step issue. You make the time step too large, and the accuracy decrease. You are trading accuracy for speed in that case, so you are not going to get the best results.

On the other hand, if you make the time step too small you are trading speed for accuracy. Therefore, it would take longer computation time to calculate the more bodies you include, and the more precise you want to get with each individual body.

When dealing with these types of functions you are dealing with what are known as $O(n^2)$ funcitons, which describe the complexity of the computation you are doing. Where n represents the number of bodies you are dealing with.

Therefore, if you had a system of one hundred bodies, that number would exponentially increase in accordance with that function.
If we wanted to improve our computations, we would also want to look into better algorithms to solve the n-body problem.

The Newtonian is pretty good in its estimation however, that, like stated above, requires an n^2 computation. We can improve that to an $O(N \log N)$ computation or better, however, like the time step change, we loose accuracy.

The other methods we could use are called the Tree Method and Particle Mesh Method.

In the tree method, as the Barnes-Hut Simulation, we divide the volume into cubic cells, so only planets need to be treated individually, and planets in large distant cells would be clustered together as a single mass.

This in turn reduces the number of planet to planet interactions and allows for faster computation time. However, to do this, we have to make sure that each planet is confined to tiny cells, so when combined it contains many planets per cell.

The other method, would be the particle mesh method. In this case the space is discretized on a mesh and for the purpose of computing gravitational potential particles are assumed to be divided between the nearby vertices of the mesh.

Finding this potential is easy since the Poisson equations:

$$\nabla^2 \phi = 4\pi G \rho$$

Where G is Newton's Constant and ρ is the density of the particles at the mesh points.

This also becomes easy to solve since we can use the fast Fourier transform and using the frequency domain where the Poisson has a simple form:

$$\hat{\phi} = -\frac{4\pi G \hat{\rho}}{k^2}$$

Where k is the wavenumber and the hats denote the transforms. The gravitational field can be found by multiplying k and computing the inverse Fourier transform. The problem with this method is it is limited by the mesh size. It is better to engage this method in small scale forces.