

Modeling Dynamics of Euler's Three Body Problem

Jasper Swallen

Department of Physics and Astronomy,

University of Southern California, Los Angeles, California 90089, USA

(Dated: November 5, 2020)

Abstract

Abstract goes here

CONTENTS

I. Background and Applications	2
II. General Equations of Motion	3
III. Numerical Solution	4
A. Finding Equations	4
B. Plottable Solutions	6
1. Euler's Method	6
2. Runge-Kutta Method	6
3. Forest & Neri Method	7
IV. Interpreting Solutions	8
A. Code	8
References	10

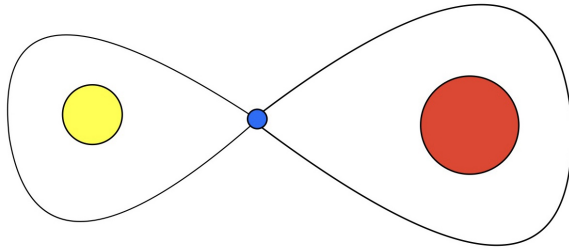


FIG. 1. A model solar system with simplified orbits

I. BACKGROUND AND APPLICATIONS

Euler's Three Body Problem is a simplification of the 3-body problem (which is itself a special case of the N-body problem). In it, two large bodies are fixed at points in space (and are assumed to not act upon each other), while one smaller body is free to move through space. This smaller body's movement is dependent only on its relative position to the two larger masses and their masses. It can be assumed that the smaller mass m is much smaller than the two larger masses m_2 and m_3 . This can therefore be represented by a one-body problem. Given the level of simplification, it would not seem that this system is a close representation of real-world problems. However, this is not the case.

There are two main applications of this problem. In a model solar system with two large stars and an orbiting planet, the movement of the planet can be described by these dynamics if the stars are too far apart to significantly attract each other but the planet is moving quickly enough to oscillate between their fields (for example, Fig. 1). The other application is in an electric field. Although electric fields are not described by exactly the same forces, their potential is still inverse to the distance. Thus, an analogue would be an electron moving through the electric field of two nuclei.

Newton's Law of Gravitation states that, for any two masses m_1 and m_2 separated by a radius $r = |\mathbf{r}_1 - \mathbf{r}_2|$ (where \mathbf{r}_1 and \mathbf{r}_2 are the positions of m_1 and m_2 , respectively), the gravitational force of m_2 on m_1 is

$$\mathbf{F}_{12} = -\frac{Gm_1m_2}{|\mathbf{r}_1 - \mathbf{r}_2|^3} * (\mathbf{r}_1 - \mathbf{r}_2) \quad (1)$$

and the potential energy is

$$U(\mathbf{r}_1, \mathbf{r}_2) = -\frac{Gm_1m_2}{|\mathbf{r}_1 - \mathbf{r}_2|} \quad (2)$$

Since gravitation forces can be superimposed, the force of every body on a given body in an N-body system can be calculated as

$$\mathbf{F}_i = \sum_{j=1, j \neq i}^N -\frac{Gm_i m_j}{|\mathbf{r}_i - \mathbf{r}_j|} * (\mathbf{r}_i - \mathbf{r}_j) \quad (3)$$

Additionally, this is a closed system, so the total energy of the system is conserved.

II. GENERAL EQUATIONS OF MOTION

The given simplification of the problem into a one-body problem results (in the plane) in two degrees of freedom. In 3-dimensional space, there are three degrees of freedom. If the three masses initially lie in a plane together, they will remain in this plane. For simplicity, the initial conditions will be chosen for such a case.

Since the total energy of the system is preserved, and the potential of the system can be described as above, the following equations of energy hold

$$\begin{aligned} T &= \frac{1}{2}mv^2 = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) \\ U_i &= -\frac{Gmm_i}{|\mathbf{r}_i - \mathbf{r}|} \\ E &= \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) - \frac{Gmm_1}{|\mathbf{r}_1 - \mathbf{r}|} - \frac{Gmm_2}{|\mathbf{r}_2 - \mathbf{r}|} \end{aligned} \quad (4)$$

While solving a Lagrangian or Hamiltonian with these equations in cartesian coordinates is possible, simplifications arise when the system is converted to an elliptical coordinate system (ξ, η) (see Fig. 2). If the masses are positioned in this coordinate system at the foci $2f$ apart, which can be done by rotating and shifting, this is translatable to $x = f \cosh \xi \cos \eta$ and $y = f \sinh \xi \sin \eta$.

Therefore, there are two conserved quantities in the system: Whittaker's constant (see [1, p.

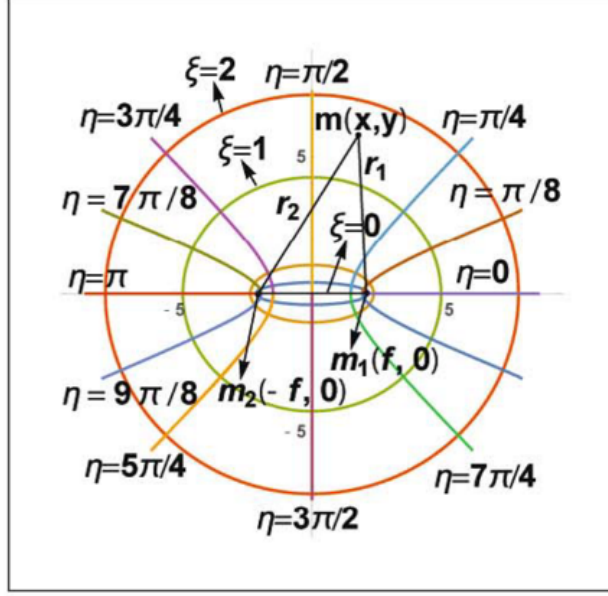


FIG. 2. the elliptical coordinate system (ξ, η) . ξ can be thought of as correspondent to radial distance, and η the angle. Adopted from [1]

100]) and total energy. Whittaker's constant can be represented as

$$\begin{aligned}
 w &= \mathbf{L}_1 \cdot \mathbf{L}_2 + 2mf(-\mu_1 \cos \theta_1 + \mu_2 \cos \theta_2) \\
 &= m^2 r_1^2 r_2^2 \dot{\theta}_1^2 \dot{\theta}_2^2 + 2mf(-\mu_1 \cos \theta_1 + \mu_2 \cos \theta_2) \\
 &= -2mf^2 E - \alpha
 \end{aligned} \tag{5}$$

where r_i is the distance $|\mathbf{r}_i - \mathbf{r}|$ between m_i and m ; $\mathbf{L}_i = m r_i^2 \dot{\theta}_i \hat{z}$ is the angular momentum of a mass; θ_i is the angle between \mathbf{r}_i and the x -axis; $\mu_i = Gmm_i$; and α is the “separation constant” derived from the separation of variables of E in elliptical coordinates [1, p. 101].

By using the Hamilton-Jacobi equation and Liouville's theorem, this system can be solved analytically. See Ó'Mathúna [2, pages 49-105 and 113-142] for a detailed solution.

III. NUMERICAL SOLUTION

A. Finding Equations

While analytical solutions are useful for determining the exact motion of systems, numerical solutions are often much simpler. Such is the case here. To do this, without loss of generality, shift the system so that m_1 is at the origin and m_2 is a distance d along the y -axis from the origin. Now represent m_2 as a factor of $m_1 = m_a$ such that $m_2 = \alpha m_a$, as shown in Fig. 3.

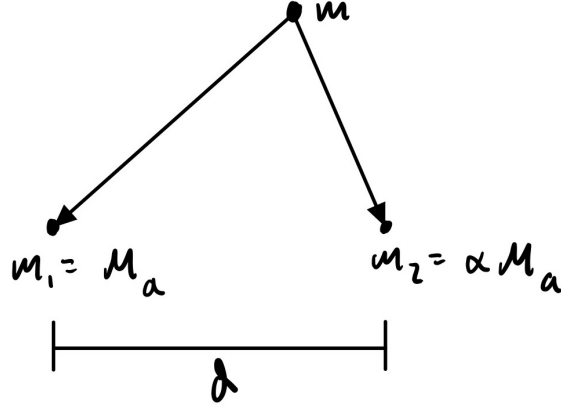


FIG. 3. the shifted system with m_1 at the origin

Up until this point, the masses and distances have been unitless and arbitrary. If these masses are instead measured in terms of solar units, further simplifications arise. If m_a is set to a “solar mass”, we see that $Gm_a = 4\pi^2$. This is because the angular velocity of a mass around a primary (for example, the Earth around the Sun) is $2\pi * AU/yr$, where AU is the average distance between the primary and the mass.

Since the origin is now fixed, the equations of motion now use relative positioning to the origin (and m_1). From this and Newton’s Law of Gravitation,

$$F_x = -\frac{Gm_a m}{r_1^3}x - \frac{G\alpha m_a m}{r_2^3}(x - d) \quad (6)$$

$$F_y = -\frac{Gm_a m}{r_1^3}y - \frac{G\alpha m_a m}{r_2^3}y \quad (7)$$

with $r_i = |\mathbf{r}_i - \mathbf{r}|$ the distance between m and m_i . In cartesian coordinates, this results in

$$\begin{aligned} \mathbf{r}_1 &= x\hat{x} + y\hat{y} \\ \mathbf{r}_2 &= (x - d)\hat{x} + y\hat{y} \end{aligned} \quad (8)$$

so

$$\begin{aligned} r_1 &= \sqrt{x^2 + y^2} \\ r_2 &= \sqrt{(x - d)^2 + y^2} \end{aligned} \quad (9)$$

Since the equations of acceleration are often used, I will define them explicitly here, dependent

on x and y inputs

$$\begin{aligned}\ddot{x}(x, y) &= 4\pi^2 \left(\frac{x}{(x^2 + y^2)^{\frac{3}{2}}} + \frac{\alpha(x - d)}{((x - d)^2 + y^2)^{\frac{3}{2}}} \right) \\ \ddot{y}(x, y) &= 4\pi^2 y \left(\frac{1}{(x^2 + y^2)^{\frac{3}{2}}} + \frac{1}{((x - d)^2 + y^2)^{\frac{3}{2}}} \right)\end{aligned}\tag{10}$$

This is a system of nonlinear differential equations. As demonstrated above, analytical solutions are possible, but messy.

B. Plottable Solutions

1. Euler's Method

Using Euler's approximation

$$x(t + \delta t) = x(t) + \delta t \dot{x}(t) + O(\delta t^2)\tag{11}$$

and neglecting higher order terms $O(\delta t^2)$, the solution is quickly plottable in cartesian coordinates. This has error on the order of δt ([3, p. 298]). Adapting 11 for x , y , \dot{x} , and \dot{y} , we get the following system of equations.

$$\begin{aligned}x_{n+1} &= x_n + \dot{x}_n \delta t \\ y_{n+1} &= y_n + \dot{y}_n \delta t \\ \dot{x}_{n+1} &= \dot{x}_n - \delta t \ddot{x}(x_{n+1}, y_{n+1}) \\ \dot{y}_{n+1} &= \dot{y}_n - \delta t \ddot{y}(x_{n+1}, y_{n+1})\end{aligned}\tag{12}$$

When they are evaluated in order, this is a programmable and plottable solution.

2. Runge-Kutta Method

Another way to incrementally solve integrable solutions is the Runge-Kutta method (specifically, the Runge-Kutta 4th Order Method). With the same logic of iteration as the Euler Method above, Runge-Kutta allows for more precision by using intermediate steps to calculate the $n + 1$ position.

The general Runge-Kutta 4th order method is

$$\begin{aligned}
x_{n+1} &= x_n + \frac{1}{6}\delta t(k_1 + 2k_2 + 2k_3 + k_4) \\
k_1 &= f(x_n) \\
k_2 &= f(x_n + \delta t \frac{k_1}{2}) \\
k_3 &= f(x_n + \delta t \frac{k_2}{2}) \\
k_4 &= f(x_n + \delta t k_3)
\end{aligned} \tag{13}$$

In the context of this 3-Body problem, the equations become (evaluated in this order)

$$\begin{aligned}
x_{n+1} &= x_n + \delta t \dot{x}_n \\
y_{n+1} &= y_n + \delta t \dot{y}_n \\
k_{1x} &= \ddot{x}(x_{n+1}, y_{n+1}) \\
k_{1y} &= \ddot{y}(x_{n+1}, y_{n+1}) \\
k_{2x} &= \ddot{x}(x_{n+1} + \delta t \frac{k_{1x}}{2}, y_{n+1} + \delta t \frac{k_{1y}}{2}) \\
&\dots \\
\dot{x}_{n+1} &= \dot{x}_n + \frac{1}{6}\delta t(k_{1x} + 2k_{2x} + 2k_{3x} + k_{4x})
\end{aligned} \tag{14}$$

3. Forest & Neri Method

$$\begin{aligned}
\beta &= 2^{\frac{1}{3}}, c_4 = c_1 = \frac{1}{2(2-\beta)}, c_3 = c_2 = \frac{1-\beta}{2(2-\beta)} \\
d_3 &= d_1 = \frac{1}{2-\beta}, d_2 = \frac{-\beta}{2-\beta}, d_4 \equiv 0
\end{aligned} \tag{15}$$

$$\begin{aligned}
x_1 &= x_n + \delta t c_1 \dot{x}_n & \dot{x}_1 &= \dot{x}_n + \delta t d_1 \ddot{x}(x_1, y_1) \\
x_2 &= x_1 + \delta t c_2 \dot{x}_1 & \dot{x}_2 &= \dot{x}_1 + \delta t d_2 \ddot{x}(x_2, y_2) \\
&\dots & & \dots \\
x_{n+1} &= x_4 & \dot{x}_{n+1} &= \dot{x}_4
\end{aligned} \quad (\text{for } x \text{ and } y) \tag{16}$$

IV. INTERPRETING SOLUTIONS

There are several cases in the limit where this solution demonstrates its clarity. The first case is when d is very large (and x is not). In this case, the terms dependent on r_2 disappear. This is equivalent to a system with only one star with mass m_a . On the other hand, if d is very small, $r_1 \cong r_2$. This is also equivalent to a system with one star, but this time the star has a mass $(1 + \alpha)m_a$.

I have yet to finish my program, so I do not have plots of results to discuss. Once I do, I will analyze these plots, especially in terms of how varying different parameters results in different solutions, and how chaotic these parameters are.

Finally, I will demonstrate my code in the text and with a link to my public GitHub repository containing a Jupyter Notebook that readers will be able to download and run.

Appendix A: Code

The code for this project is fully available at [my public GitHub repository](#), containing both a Jupyter Notebook and command-line Python script. The following is `main.py::eulerMethod`, which calculates and plots the graph of Euler's Three Body Problem with Euler's numerical approximation method. This code is found in `main.py` and `main.ipynb`. The requirements to run this code are Python 3, `matplotlib`, and `numpy`. Then, run `$ python3 main.py` and input the initial positions and masses of all three objects, as well as the initial velocity of the moving object.

```
1 def eulerMethod(mass_1_x , mass_1_y , mass_2_x , mass_2_y , alpha ,
    moving_mass_x , moving_mass_y , moving_mass_x_prime ,
    moving_mass_y_prime , its , delta):
2     x = [moving_mass_x]
3     y = [moving_mass_y]
4     x_prime = [moving_mass_x_prime]
5     y_prime = [moving_mass_y_prime]
6
7     # We have the following equations of motion
8     # x'_{(n+1)}=x'_n-deltat4pi^2 (x_{(n+1)}/(r_1^3)_{(n+1)} +alpha(x_{(n+1)}-d
        ) / (r_2^3)_{(n+1)})
```



```

9      #  $y'_{(n+1)} = y'_n - \frac{\Delta t}{4} \pi^2 y_{(n+1)} \left( \frac{1}{(r_1)^3} - \frac{1}{(r_2)^3} \right)$ 
10
11     for n in range(its):
12         # calculate next x, y, x', and y'
13         x_n_plus_one = x[n] + delta * x_prime[n]
14         x.append(x_n_plus_one)
15         y_n_plus_one = y[n] + delta * y_prime[n]
16         y.append(y_n_plus_one)
17
18         x_prime_n_plus_one = x_prime[n] + delta * \
19             calculateXAccel(x[n + 1], y[n + 1])
20         x_prime.append(x_prime_n_plus_one)
21
22         y_prime_n_plus_one = y_prime[n] + delta * \
23             calculateYAccel(x[n + 1], y[n + 1])
24         y_prime.append(y_prime_n_plus_one)
25
26     plt.plot(x, y)
27
28     masses = [[mass_1_x, mass_2_x, moving_mass_x],
29               [mass_1_y, mass_2_y, moving_mass_y]]
30     colors = np.array([[255, 0, 0], [0, 255, 0], [0, 0, 255]])
31
32     plt.scatter(masses[0], masses[1], c=colors / 255)
33
34     labels = ["m1", "m2", "m"]
35     for i, txt in enumerate(labels):
36         plt.annotate(txt, (masses[0][i], masses[1][i]))

```

The Runge-Kutta and Forest & Neri methods work in a similar manner, although they require

many more calculations per iteration.

-
- [1] G. S. Krishnaswami and H. Senapati, *An Introduction to the Classical Three-Body Problem: From Periodic Solutions to Instabilities and Chaos*, Resonance **24**, 87 (2019).
 - [2] D. Ó'Mathúna, *Integrable Systems in Celestial Mechanics*, 1st ed. (Birkhäuser Boston, 2008).
 - [3] W. J. Wild, *Euler's Three Body Problem*, American Journal of Physics **48**, 297 (1980).