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: October 11, 2020)

Abstract

Abstract goes here

CONTENTS

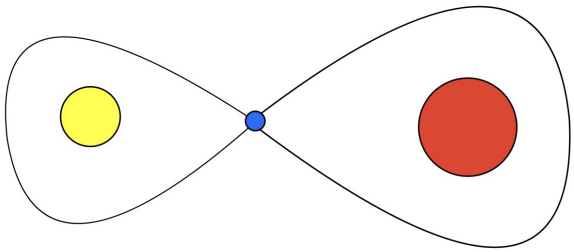

FIG. 1. A model solar system with simplified orbits

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 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 $r = |\boldsymbol{r_1} - \boldsymbol{r_2}|$ (where $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ are the positions of m_1 and m_2, respectively), the gravitational force of m_2 on m_1 is

$$\boldsymbol{F_{12}} = -\frac{Gm_1m_2}{|\boldsymbol{r_1} - \boldsymbol{r_2}|^3} * (\boldsymbol{r_1} - \boldsymbol{r_2}) \tag{1}$$

and the potential energy is

$$U(\boldsymbol{r_1}, \boldsymbol{r_2}) = -\frac{Gm_1m_2}{|\boldsymbol{r_1} - \boldsymbol{r_2}|} \tag{2}$$

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

$$\boldsymbol{F_i} = \sum_{j=1, j\neq i}^{N} -\frac{Gm_i m_j}{|\boldsymbol{r_i} - \boldsymbol{r_j}|} * (\boldsymbol{r_i} - \boldsymbol{r_j}) \tag{3}$$

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

$$T = \frac{1}{2}mv^2 = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) \tag{4}$$

$$U_i = -\frac{Gmm_i}{|\boldsymbol{r_i} - \boldsymbol{r}|} \tag{5}$$

$$E = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) - \frac{Gmm_1}{|\boldsymbol{r_1} - \boldsymbol{r}|} - \frac{Gmm_2}{|\boldsymbol{r_2} - \boldsymbol{r}|} \tag{6}$$

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) 2. If the masses are positioned in this coordinate system at the foci $2f$ apart, this is translatable to

$$x = f \cosh\xi \cos\eta$$

and

$$y = f \sinh\xi \sin\eta$$

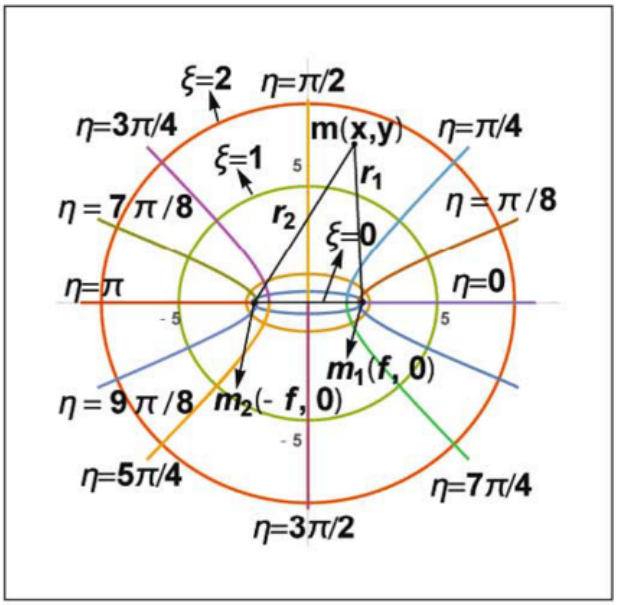

FIG. 2. the elliptical coordinate system (ξ, η). ξ can be thought of as correspondent to radial distance, and η the angle.

Therefore, there are two conserved quantities in the system: Whittaker's constant (see Krishnaswami 100) and total energy. Whittaker's constant can be represented as

$$\begin{aligned} w &= \boldsymbol{L_1} \cdot \boldsymbol{L_2} + 2mf(-\mu_1 \cos\theta_1 + \mu_2 \cos\theta_2) \\ &= m^2 r_1^2 r_2^2 \dot{\theta_1}^2 \dot{\theta_2}^2 + 2mf(-\mu_1 \cos\theta_1 + \mu_2 \cos\theta_2) \\ &= -2mf^2 E - \alpha \end{aligned} \tag{7}$$

where r_i is the distance $|\boldsymbol{r_i} - \boldsymbol{r}|$ between m_i and m; $\boldsymbol{L_i} = m r_i^2 \dot{\theta}_i \hat{z}$ is the angular momentum of a mass; θ_i is the angle between $\boldsymbol{r_i}$ 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 (Krishnaswami 101 [1]).

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

While analytical solutions are useful for determining the exact motion of systems, numerical solutions are often much simpler. Such is the case here. Using Euler's approximation

$$x(t + \delta t) = x(t) + \delta t \dot{x}(t) + O(\delta t^2) \tag{8}$$

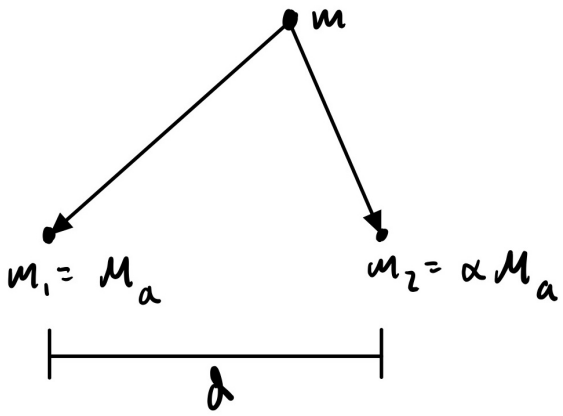

FIG. 3. the shifted system with m_1 at the origin

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 (Wild 298 [3]).

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 $m_2 = \alpha m_a$ 3.

Up until this point, our masses and distances have been unitless and arbitrary. If these masses are instead measured in terms of solar units, further simplifications arise. If we set m_a equal 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 we fixed the origin, our equations of motion now use relative positioning to the origin (and m_1). From this and Newton's Law of Gravitation, we see that

$$F_x = -\frac{Gm_a m}{r_1^3}x - \frac{G\alpha m_a m}{r_2^3}(x-d) \tag{9}$$

$$F_y = -\frac{Gm_a m}{r_1^3}y - \frac{G\alpha m_a m}{r_2^3}y \tag{10}$$

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

$$\begin{aligned} \boldsymbol{r_1} &= x\hat{x} + y\hat{y} \\ \boldsymbol{r_2} &= (x-d)\hat{x} + y\hat{y} \end{aligned} \tag{11}$$

so

$$\begin{aligned} r_1 &= \sqrt{x^2+y^2} \\ r_2 &= \sqrt{(x-d)^2+y^2} \end{aligned} \tag{12}$$

This is a system of nonlinear differential equations. As demonstrated above, analytical solutions are possible, but messy. Instead, using Euler's method 8 and plugging in 9 and 10

$$\begin{aligned} \dot{x}(t+\delta t) &= \dot{x}(t) + \delta t\ddot{x}(t) + O(\frac{\dot{x}}{dt^2}) \\ &\approxeq \dot{x}(t) - \delta t 4\pi^2(\frac{x}{r_1^3} + \frac{\alpha(x-d)}{r_2^3}) \end{aligned} \tag{13}$$

$$\begin{aligned} \dot{y}(t+\delta t) &= \dot{y}(t) + \delta t\ddot{y}(t) + O(\frac{\dot{y}}{dt^2}) \\ &\approxeq \dot{y}(t) - \delta t 4\pi^2 y(\frac{1}{r_1^3} + \frac{1}{r_2^3}) \end{aligned} \tag{14}$$

If we set $f_n(t) = f(n\delta t)$, we get the following system of equations

$$\begin{aligned} x_{n+1} &= x_n + \dot{x}_n\delta t \\ y_{n+1} &= y_n + \dot{y}_n\delta t \\ \dot{x}_{n+1} &= \dot{x}_n - \delta t 4\pi^2(\frac{x_{n+1}}{r_{1n+1}^3} + \frac{\alpha(x_{n+1}-d)}{r_{2n+1}^3}) \\ \dot{y}_{n+1} &= \dot{y}_n - \delta t 4\pi^2 y_{n+1}(\frac{1}{r_{1n+1}^3} + \frac{1}{r_{2n+1}^3}) \end{aligned} \tag{15}$$

This is a programmable and plottable solution.

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, $r_1 \approxeq r_2$. 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 a stripped down version of `main.py`.

```
1  import matplotlib.pyplot as plt
2  import numpy as np
3  import math
4  ### Input code is skipped for brevity
5  # declare arrays that will be appended to in loop
6  x = [moving_mass_x]
7  y = [moving_mass_y]
8  x_prime = [moving_mass_x_prime]
9  y_prime = [moving_mass_y_prime]
10
11 # We have the following equations of motion
12
13 for n in range(its):
14     # calculate next x, y, x', and y'
15     x_n_plus_one = x[n] + delta * x_prime[n]
16     x.append(x_n_plus_one)
17     y_n_plus_one = y[n] + delta * y_prime[n]
18     y.append(y_n_plus_one)
19
20     x_prime_n_plus_one = x_prime[n] - delta * 4 * (np.pi ** 2) * (x[n +
           1] / np.power((x[n + 1] ** 2) + (y[n + 1] ** 2), 3 / 2) + alpha
           * (x[n+1] - mass_2_x) / np.power((x[n + 1] - mass_2_x) ** 2 + y
          [n + 1] ** 2, 3 / 2))
21     x_prime.append(x_prime_n_plus_one)
```

```
22
23      y_prime_n_plus_one = y_prime[n] - delta * 4 * (np.pi ** 2) * y[n +
            1] * (1 / np.power((x[n + 1] ** 2) + (y[n + 1] ** 2), 3 / 2) + 1
             / np.power((x[n + 1] - mass_2_x) ** 2 + y[n + 1] ** 2, 3 / 2))
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]))
37
38  plt.show()
```

---

[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).