The Three Body Problem

Edward Rios[1,*]

[1]*Department of Applied Mathematics and Physics, Valdosta State University*

The three body problem, describes the motion of three masses interacting by the Newtonian gravity. For over three centuries, the absence of constraints on the initial positions and velocities of these masses has captivated the interest of numerous scientists. This paper is an adaptation focusing on the problem found by Burrau (1913), and the general solution made by Szebehely and Peters (1967). The discovered solution does not exhibit quasi-periodic or periodic behavior; instead, it takes the shape of 'elliptic-hyperbolic'. For the adaptation of this problem, I used `Python` to compute a high order numerical integration to get an accurate result to match the answers from Szebehely and Peters and also because is needed for the close approaches between bodies. In the ultimate arrangement, two out of the three involved bodies establish a lasting binary relationship, while the third body is ejected into infinity. New adaptations of the solution are also described. This study not only highlights the complexities of celestial mechanics but also emphasizes the importance of employing precise numerical methods to explore complex gravitational interactions.

I. INTRODUCTION

The three-body problem stands as one of the foundational challenges in physics, rooted in Newton's second law and his law of universal gravitation, governing the motions of celestial bodies like planets and stars. This problem has inspired many astronomers and mathematicians in the last three centuries. While Newton's equations adeptly predict the movements of two gravitating masses, extending these principles to describe the motion of three or more interacting celestial objects remains an intricate puzzle. Throughout history, many astronomers, physicists, and mathematicians have grappled with this enigma, seeking elusive solutions. However, the inherently unpredictable nature of three-body motion has rendered the problem one of the most formidable conundrums in the annals of scientific inquiry Musielak and Quarles (2014). By the fact that initial conditions for a three body can alter the system drastically, it is evident that it can't be solved as the same way as a two body system, because it could be solved by using mathematical formulas. The three body problem is one of the standard examples to explain chaos Boekholt et al. (2021). Chaos is an inherent property of most dynamical systems in the universe, ranging from small bodies small bodies up to galaxies. The main signature of chaos is the exponential sensitivity to small changes in the initial conditions Boekholt et al. (2020). Even then there has been different applications for the problem, a general and specific solution stills not found. In the late 1800s mathematicians Enrst Heinrich Burns and Poincaré (1892)came to the conclusion that no analytical solution exist. In the early years of the 21st century, there has been found that the Three-body problem can be represented in different ways, and various scales such as the solar system (satellites, comets, planets etc.), binary stars, binary black holes, galaxies etc Valtonen and Karttunen (2006).

In 1893, a mathematician named Meissel proposed a particular problem for the three body problem known as the Pythagorean problem. Later on, in 1913 Carl Burrau tried to develop a numerical integration for the problem, on his paper of numerical calculations for a special case of the three body problem Burrau (1913). The Burrau's problem was based on three point masses attract each other according to the Newtonian law of gravitation. The masses of the particles are $m_1 = 3$, $m_2 = 4$, and $m_3 = 5$; they are initially located at the apexes of a right triangle with sides 3, 4, and 5, so that the corresponding masses and sides are opposite. The particles are free to move in the plane of the triangle and are at rest initially. The initial configurations for the Pythagorean problem can be seen in figure 1.

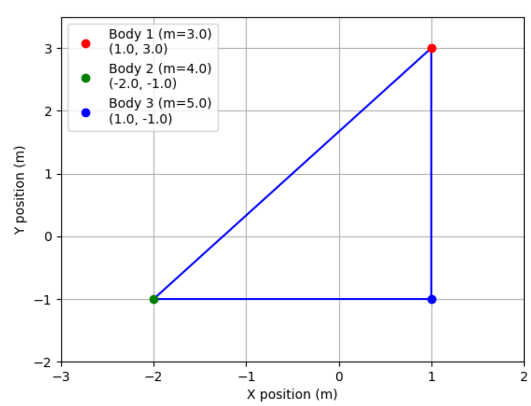

Figure 1: Initial configuration of the Pythagorean problem of three bodies adapted by the figure of Szebehely and Peters (1967).

This paper is based on one of the first solutions applied for the three body problem worked by Victor Szebehely and C. Frederick Peters by using computational

* erios@valdosta.edu

simulation to study the motion of the orbits for a three body system. Their solution was being applied for the Burrau's problem, and they considered its problem to be short to come a conclusion and try to 'complete' a solution for the problem. Then Peter introduced a new method to integrate the equations of motion Szebehely and Peters (1967). Two out of the three bodies follow paths that resemble elliptical orbits as they orbit each other closely, behaving like a binary system. Meanwhile, the third body takes a different trajectory, departing on a hyperbolic path that ultimately leads it to move away to infinity. The initial velocities for this bodies are zero and they remain zero during the entire motion, then it is going to be two-dimensional motions of a system having negative total energy. The regions of chaoticity and regularity are established by keeping the locations of two masses fixed and changing the initial position of the third body Aarseth et al. (1994). Initially, the central point of mass is located at the starting point of the chosen coordinate system, and it remains fixed at that position as the system develops. Their research focused on observing the interactions among the three orbits over various time intervals, examining how the orbits influenced each other and how the dynamics of chaos unfolded as time progressed.

A. Progress of the dynamical system

The development of the bodies orbits can be seen at figure 2. The progression of the orbits through time are according to the same evolution Szebehely and Peters (1967) had on their original paper. The marks being shown in the horizontal x axis and the vertical y axis are at unit distances.The programming language `Python` was used to generate the transformation depicted in Figure 2. The code that produces the figures can be found in Appendix V.

On the next sections are some of the key steps needed to take to solve the problem according to Szebehely and Peters (1967), the method that is used to integrate the equations of motion towards the three bodies action on each other. Unlike Meissel or Burrau, who attempted to solve this problem manually in the past, contemporary approaches leverage machine learning for more efficient and streamlined solutions to complex problems. In this context, the programming language `Python` is employed due to its straightforward and rapid comprehension.

II. METHODS

As I mentioned earlier the three-body problem is different than the two-body problem because by adding one more body to the problem, it creates a complicated non-linear problem. It contains 18 variables—each body has three position and three velocity components—and the equations of motion (derived using Newton's Laws) are a set of nine second-order differential equations Barrow-Green (2010).

To understand the units used in the figures, there has to be a physical meaning, noting that the equations of motions are integrated using unity for the dimensionless parameter of the problem, in this case using an equation that is related to the orbital motion of a celestial body:

$$T^2 \frac{GM}{L^3} = 1, \tag{1}$$

In the system of units utilized, where T, L, and M represent time, length, and mass, respectively, and G=6.67X10^{-8} is the gravitational constant in cgs (centimeter, gram, second) units.

The numerical model employed to address this problem traces its origins to Newton's laws of motion, which can be summarized as follows:

- An object in uniform motion will persist in that state unless acted upon by an external force.
- Force is equivalent to the change in momentum per unit change in time.
- For every action, there is an equal and opposite reaction.

Next, when considering the gravitational force exerted on the bodies, commonly referred to as Newton's gravitational force, the applicable equations of motion are as follows:

$$M\frac{d^2\vec{R_i}}{dt^2} = G\sum_{j\neq i}^{3} \frac{M_i M_j}{r_{ij}^3}\vec{r_{ij}}, \tag{2}$$

where the applied summation accounts for the gravitational force acting on the masses of the three celestial bodies, taking into consideration their respective distances. According to Szebehely and Peters (1967), the mass is being express in grams and the distances in centimeters , then the unit of time becomes 3872 seconds or close to 1.08 h.

A. Method of Regularization

In order to match an accurate answer with respect to the results obtained by Szebehely and Peters (1967), the equation of motions are subjected to regularization, because it deals with collision and close approaches. One problem I had particularly by solving this problem using `Python` is the accuracy of integration due to the close approaches between bodies. It is important to understand that when two small particles get really close, it's crucial to use a very high order to get an accurate answer. If a correct method is not being used, it could cause small mistakes and produce different outcomes, leading to different final results with respect to the original paper. The fundamental idea by doing this method is to eliminate singularities occurring at binary collisions in the the problem of three bodies when it comes to introducing a new variable. This method was proposed by Sundman (1913). Sundman approached the problem by considering triple as well as binary collisions , and fundamental to his solution was the introduction of an auxiliary variable which the coordinates and the time were generalized to complex values Barrow-Green (2010).

Before Sundman came with the method for a solution to regulate the accuracy of the numerical integration, there was an important work on singularities that proved they are collisions by Painlevé (1897). He showed that the problem could be solved using convergent power series, in another words that the Newton's equations of motion for the three bodies could be integrable by using power series. Painlevé thought that the initial conditions must satisfy two different analytic relations, so he was unable to make a further progress, but at least he influenced another physicist in solving the problem.

Going back to Sundman and the clarification made by Szebehely and Peters (1967), Sundman's method does not seem to be satisfactory because of practical computational reasons. One of the main discrepancies between the analytic and computational requirements is the appearance of terms of the type x_i/r in transformed equations. x_i and r must approach zero at collision. Then Szebehely and Peters (1967), showed that by introducing a new independent variable, while regularizing the restricted problem, introduces complications in the equations of motion and brings about the emergence of terms that are not ideal for numerical integration. The remedy found was to increase the complexity of the regularizing transformation in the case of the restricted problem.

After the possible method that Painlevé (1897) found for singularities that proved they are collisions, another step was taken by Levi-Civita (1904) and it was also important for Szebehely and Peters (1967) to solve the problem. It combines the introduction of a new time variable τ and a transformation of coordinates from the system, x, y to the system ξ, η, according to Szebehely and Peters (1967), these aspects are described by the following equations:

$$\tau^* = \int_{t_0}^{t} \frac{dt}{r} + \tau_0^*, \tag{3}$$

$$x + iy = (\xi + i\eta)^2. \tag{4}$$

The equation 3 and 4 requires to transform the equations of motion 2 into the application of Levi-Civita's method. Check the book by Wintner (2014) for celestial mechanics, where the method of regularization is applied to a relatively simple problem. Given that r_{ij} is the distance between the ith and jth particles, a limiting value for r_{ij} is set. When two of the three bodies are closer than the limiting value set, the equation of motion for these two bodies are to be regularized. If the distance between each body is below the limit, that pair is selected for regularization which has the smallest distance. At certain time when the distance between a critical pair decreases below the given limit, the equations are regularized and when the pair increases their distance above the limit, the equations of motion are transferred back to the original variables. All these procedure is not easy to implement because every time there is change in the variables, the integration must be restarted with a new set of initial conditions. Therefore, a Runge-Kutta method stands out as a superior and accurate choice for numerical integration in this particular problem.

B. Method of Numerical Integration

Three years before Szebehely and Peters (1967) came with the complete solution of the general problem of three bodies, a mathematician named Zonneveld, wrote a book where he developed a fifth order Runge-Kutta scheme to numerically integrate the equations of motion. Zonneveld made and expression for the last term of the fifth order, included in the Taylor series, and it was meant to derive the algorithm. According to Szebehely and Peters (1967), and private communication they had with Ollongren back in 1966, he derived a formula to predict the optimum step size to be used at each step of the integration. If h_i is the size of the ith step, E_{max} the maximum allowable truncation error, and E_5, the estimate of the truncation error of the fifth-order , then the optimum size of the $(i+1)$th step is given by

$$h_i + 1 = \left(\frac{E_{max}}{E_{max} + E_5} + 0.45 \right) h_i. \tag{5}$$

When $E_5 \geq E_{max}$ the step is rejected, and the same formula has to be used again until the requirement for the tolerance is required, i.e., $E_5 < E_{max}$. Based on the application that was made by Szebehely and Peters (1967), one out of every 1000 steps was rejected. In the regularized system the total energy is defined in terms of the original variables which makes it an inconvenient quantity to compute. For this case, the transformed Hamiltonian, which should be identical to zero, is used as the control of the energy.

In the presence of optimization, particularly in this context where a Taylor series is employed, determining the optimal number of terms in the series might not be straightforward. This is because TAYLOR utilizes a text input of the ODEs to be solved and offers the user an optimal time-stepper to minimize error.

C. Adaptation to the Solution

The numerical integration I used to adapt the solution made by Szebehely and Peters (1967) was with a method named `DOP853` applied in `Python` that can be seen in the Appendix V. This method is a variant of the Runge-Kutta method that provides high accuracy and efficiency. The numbers 8(5,3) in `DOP853` indicate that the method has an eight-order accurate formula for the solution, with a fifth-order formula for estimating the local error, and a third order formula for estimating the error in the error estimate. It is particularly suitable for problems where the solution undergoes rapid changes or has stiff regions. This method also employs an adaptive step size control, meaning that the algorithm dynamically adjusts the step size during integration based on the characteristics of the solution. All this is done by estimating the local error at each step and comparing it to the specified tolerance levels. If the estimated error is within acceptable bounds, the step size remains unchanged. If the error exceeds the tolerances, the step size is adjusted to ensure that the solution remains accurate. This adaptability allows for accurate representation of the three-body problem while optimizing computational efficiency by using larger steps in smoother regions and smaller steps where precision is crucial.

III. RESULTS

A. Orbital Dynamics Over Time

Looking at the solution from Szebehely and Peters (1967), they made a simulation based on how the orbits differ and interact with each other in a time interval of 70 seconds. In their well-known paper, they demonstrated the dynamic changes in the orbits of three bodies occurring every 10 seconds. I adopted a consistent approach of dividing the time interval into 10-second segments to produce the same figures. Given the current ease of implementation, creating these figures is simpler than in the past due to the availability of tools. Using the programming language `Python` and `Matplotlib`, a command that facilitates the creation of static, animated, and interactive visualizations in `Python`, made the process of generating the figures straightforward.

The adaptation of the orbits from figure 2 were exactly the same as the one from Szebehely and Peters (1967), where the motion of the first body is shown by a dotted line, the dashes represent the orbit of m_2 and the solid line illustrates the motion of the body with the largest mass m_3. Check the original paper by Szebehely and Peters (1967) to see details as the numbers on the curves and what they mean.

Looking at figure 2, at interval $0 \leq t \leq 10$, the body with the greatest mass, representing the third body, repeatedly undergoes close encounters with the first ans second bodies. However, the first and second bodies themselves do not come into close proximity to each other. During the entire motion, the body with the greatest mass consistently acts as an intermediary between the other two masses. This implies that the interaction, communication, and direct gravitational effect between m_1 and m_2 occur solely through m_3. The separation between $m1$ and $m2$ can be seen in figure 2 from **a** to **c**. It also can be seen that in part **f**, there is an instant when the two bodies with the smaller masses are closer to each other than the third body ($r_{12} < r_{23}$) Szebehely and Peters (1967).

According to Szebehely and Peters (1967), by observing the figure 2 part **a** at $t = 3.35$ m_1 moves away from the origin, while m_2 and m_3 are on approach trajectories. Analyzing Burrau (1913) integration that ended in $t = 3.35$, due to a lack of the necessary tools, it is understandable that he could not identify any indications of either periodicity or asymptotic behavior. Saving the code into a file proves to be a useful method for acquiring data during close encounters between two bodies and calculating the approximate distance at specific times. For part **b** of figure 2, m_1 and m_2 are experiencing their most distant separation , and in the middle m_3 is connecting both bodies, making the assumption that the two bodies bodies with smaller masses are being attract by the body with larger mass. The closest approach of the entire motion for the three bodies is happening in part **b**. This close approach is being made from $r_{23} \cong 4 \times 10^{-4}$, which is the smallest distance occurring between any two bodies at any time during the evolution of this dynamic system between $t = 0$ and $t = \infty$. On figure 2 part **f**, shows the final formation of a binary by m_2 and m_3. One way to characterize the interaction is to depict m_1 as traversing the binary system without significantly affecting it, while concurrently achieving a considerable speed. Lastly, in part **g** is where the simulation of the problem ends. The binary and the body with mass m_1 depart with hyperbolic velocities. According to Szebehely and Peters (1967), the period of the binary is approximately 0.9 time unit and therefore this binary performs 15 revolutions during one complete rotation of the galaxy. This calculations mentioning about the hyperbolic departure can be used from Alekseev (1962) conditions. Aleeksev aimed to solve the problem of hyperbolic motion, introducing the additional constraint that the initial distance between the bodies is minimized. The method developed is based on a general theorem in the theory of the perturbation of differential equations. The following table, contains the close approaches between t=0 and t=30.

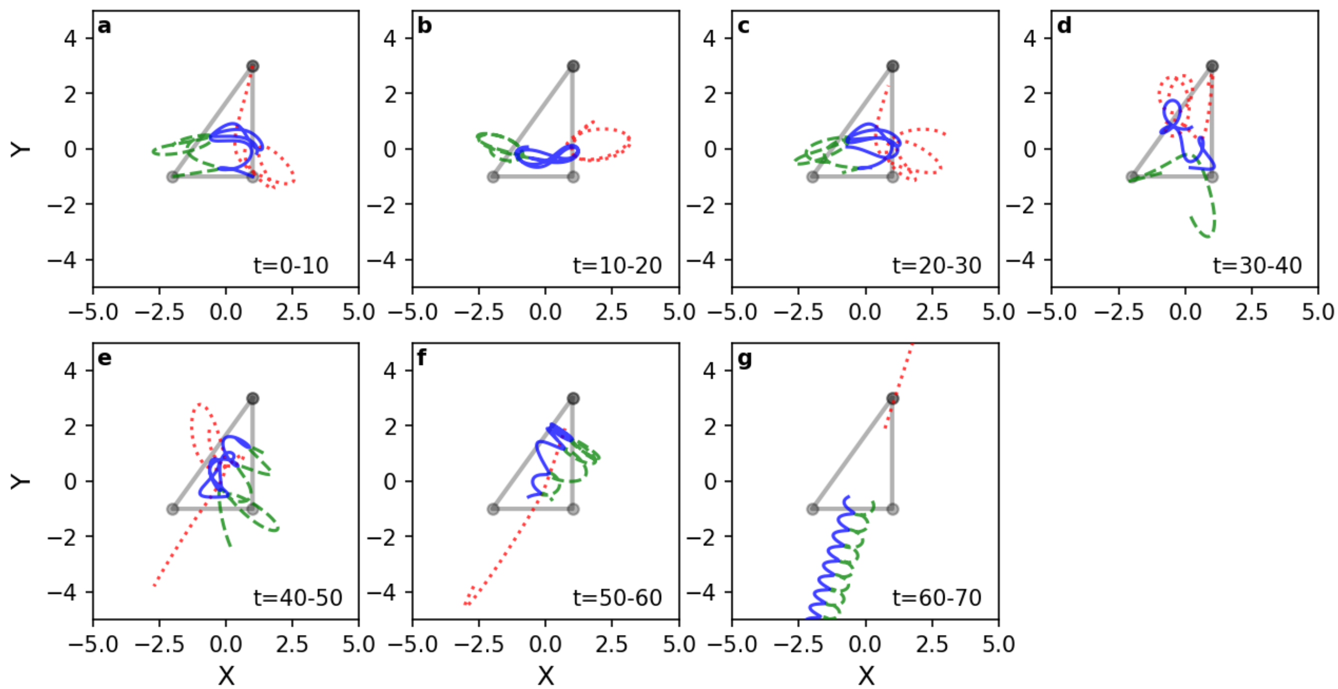

Figure 2: This figure illustrates how the orbits of the three bodies interact with each other, undergoing changes every 10 seconds. Notice how the right triangle is always in the background to represent the initial position of the bodies.

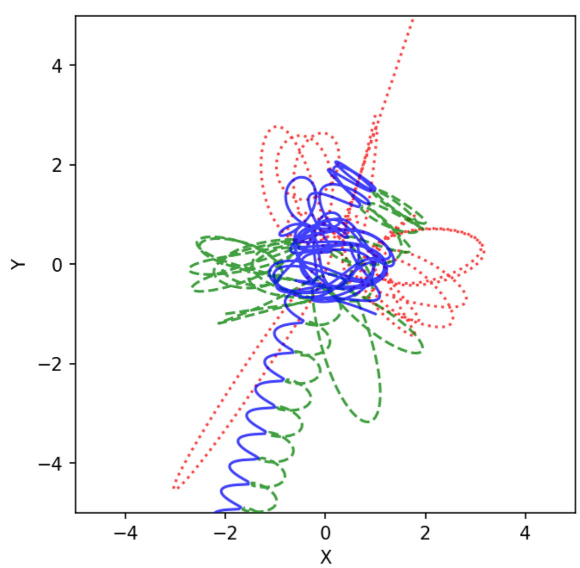

Figure 3: The figure shows the overall motion of the three bodies during the time interval of 70 seconds.

Time	Approximate Distances	Between Particles
$t_1 = 1.879$	10^{-2}	m_2, m_3
$t_4 = 3.026$	0.6	m_1, m_3
$t_2 = 3.801$	$6{\times}10^{-2}$	m_2,m_3
$t_5 = 6.898$	0.1	m_1, m_3
$t_3 = 8.760$	$8{\times}10^{-3}$	m_2, m_3
$t_6 = 9.962$	0.5	m_1, m_3
$t_9 = 11.611$	0.2	m_2, m_3
$t_7 = 14.618$	0.2	m_1, m_3
$t_{10} = 15.830$	$4{\times}10^{-4}$	m_2, m_3
$t_8 = 17.001$	0.3	m_1, m_3
$t_{11} = 19.807$	0.2	m_2, m_3
21.791	0.4	m_1, m_3
22.966	$2{\times}10^{-2}$	m_2, m_3
24.537	0.1	m_1, m_3
27.780	$5{\times}10^{-2}$	m_2, m_3
28.679	0.5	m_1, m_3
29.802	$3{\times}10^{-3}$	m_2, m_3

Table I: Data set of close approaches from $t = 0$ to $t = 30$ based on table I made by Szebehely and Peters (1967)

By looking at table I the closest approach occurs at t_{10} between m_2 and m_3 with an approximate distance of 4×10^{-4}. Based on Szebehely and Peters (1967), this data from $t = 0$ to $t = 30$ was for the first 430 million years.

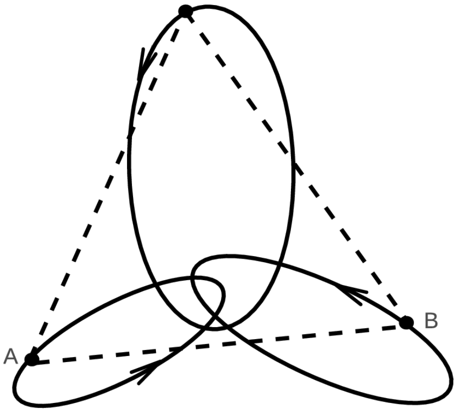

Figure 4: Illustration of the Lagrange solution with an equilateral triangle joining three masses M_1, M_2 and M_3 located at the corresponding points A, B and C. Illustration made by Musielak and Quarles (2014).

B. Recent Alternations to the Problem

In the last decade, Boekholt and Portegies Zwart (2015), Portegies Zwart and Boekholt (2018), obtained a reversible solution to the Pythagorean problem (Burrau (1913), Szebehely and Peters (1967) and Aarseth et al. (1994). They applied the `Brutus` N-body code and the method of convergence in which the accuracy and precision of the integration is systematically increased until convergence of the solution to the first few decimal places (Boekholt and Portegies Zwart (2015). While forward integration experiences exponential divergence, backward integration exhibits exponential convergence towards the initial perturbation size across nine orders of magnitude. Portegies Zwart and Boekholt (2018) referred to this behavior as "definitive reversibility". They adopt the initial conditions for the Pythagorean problem and integrate up to $t = 100$. The first calculations they made were with a high tolerance and a short word-length, so later they could compared the simulation with a higher tolerance and larger word-length.

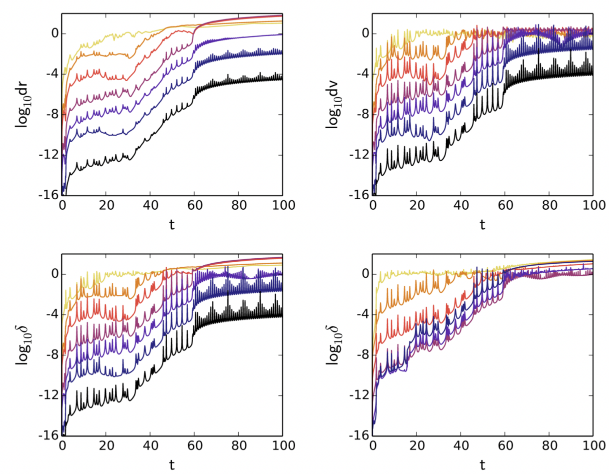

Figure 5: **Exponential Divergence in the Pythagorean problem.** In the top two panels and the lower left panel `Brutus` is compared with `Brutus` with increasing precision. See entire caption from Boekholt and Portegies Zwart (2015).

To confirm the time reversibility of Newton's equations of motion through numerical methods, one can showcase the presence of converging trajectories in phase space. This involves demonstrating the capability to compute any non-colliding initial condition forward to an arbitrary point in time and backward to restore the initial state, with the accumulated integration error significantly smaller than the initial discrepancy.

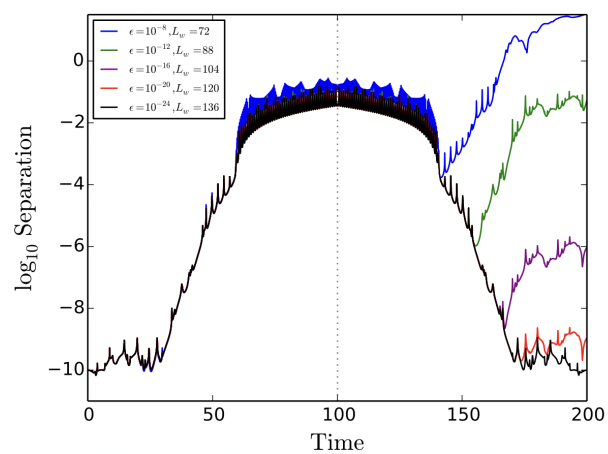

Figure 6: Evolution of the phase-space separation δ between two solutions of the Pythagorean 3-body problem. Check the entire illustration from Portegies Zwart and Boekholt (2018).

Figure 6 demonstrates the numerical time-reversibility by presenting the result for the Pythagorean 3-body problem. This solution was implemented by using `Brutus` which is part of the AMUSE software environment. `Brutus` can be tuned until the solution converges, and this is due by the integration accuracy caused by the tolerance (ϵ) and the length of the mantissa η. This problem-solving approach provides insight into an alternative and distinctive method of addressing the problem. It allows for diverse representations, extending beyond tracking the changing orbits over time to interpreting the evolution of the phase space across time.

IV. CONCLUSION

Exploring the three-body problem has proven to be an insightful way to understand its complexities. It underscores the challenge of finding solutions, revealing multiple approaches and various methods for resolution. Examining the problem reveals another intriguing aspect: its evolution over time and the adaptive strategies employed by astronomers, physicists, and mathematicians in attempting to solve it. In contemporary times, addressing the problem has become more feasible through computational simulations, particularly in the case of the Pythagorean Problem. This was not an option during the attempt made in Burrau (1913) to solve the problem. Burrau's simulation reached until a certain point because he did not have the required tool to compute beyond that. Reading different resources is also a helpful way to see what numerical integration methods are more precise, and then compare them. In this problem based on the solution from Szebehely and Peters (1967), I tried solving the problem by using `Python` and a high order numerical integration, `DOP853` which is using an eight-order accurate for the solution. The results and the motion of the three bodies orbits through the time interval of 70 seconds from figure 2 were accurate with the results obtained from Szebehely and Peters (1967). I found this problem interesting to analyze its chaos, and I wondered about the potential outcomes if the simulation run time exceeds the solution from Portegies Zwart and Boekholt (2018). I am looking forward to seeing how technology evolves and progresses with this problem because I think that will allow us, as scientists, to interpret it better.

V. APPENDIX

```
# Importing necessary libraries
import numpy as np # NumPy provides numerical
    red↪ operations and arrays
from scipy.integrate import solve_ivp #
    red↪ solve_ivp is used for solving initial
    red↪  value problems
import matplotlib.pyplot as plt # Matplotlib
    red↪ is used for plotting

def three_body_problem(t, z):
    # Function defining the general three-body
        red↪ problem for solve_ivp
    # z = [x1, y1, x2, y2, x3, y3, vx1, vy1,
        red↪ vx2, vy2, vx3, vy3]; t = time (
        red↪ unused)

    # Extracting positions and velocities
    dx12, dx13, dx23 = z[2] - z[0], z[4] - z
        red↪ [0], z[4] - z[2]
    dy12, dy13, dy23 = z[3] - z[1], z[5] - z
        red↪ [1], z[5] - z[3]

    # Distances between bodies
    r12 = np.sqrt(dx12**2 + dy12**2) # |r1 - r2
        red↪ |
    r13 = np.sqrt(dx13**2 + dy13**2) # |r1 - r3
        red↪ |
    r23 = np.sqrt(dx23**2 + dy23**2) # |r2 - r3
        red↪ |

    # Initialize return state
    yp = np.zeros(12)
    yp[:6] = z[6:]

    # Equations of motion
    yp[6] = mass[2] * dx12 / r12**3 + mass[3] *
        red↪  dx13 / r13**3
    yp[7] = mass[2] * dy12 / r12**3 + mass[3] *
        red↪  dy13 / r13**3
    yp[8] = -mass[1] * dx12 / r12**3 + mass[3]
        red↪ * dx23 / r23**3
    yp[9] = -mass[1] * dy12 / r12**3 + mass[3]
        red↪ * dy23 / r23**3
    yp[10] = -mass[1] * dx13 / r13**3 - mass[2]
        red↪  * dx23 / r23**3
    yp[11] = -mass[1] * dy13 / r13**3 - mass[2]
        red↪  * dy23 / r23**3

    return yp

def write_to_file(file_name, time, state,
    red↪ header=True):
    # Function to write time and state to a
        red↪ file in CSV format
    if header:
        outfile = open(file_name, 'w')
        out_str = "#t,x1,y1,x2,y2,z2,x3,y3,z3\n
            red↪ "
        outfile.write(out_str)
    else:
        outfile = open(file_name, 'a')

    out_str = "%2.5f,␣" % time
    for j in range(0, 6):
        out_str += "%1.5f,␣" % state[j]
    outfile.write(out_str[:-2] + "\n")
    outfile.close()

# Initial conditions
mass = np.array([0, 3, 4, 5]) # Masses of the
    red↪ three bodies
initial_state = np.array([1, 3, -2, -1, 1, -1,
    red↪  0, 0, 0, 0, 0, 0]) # Initial
    red↪ positions and velocities
time_step = 1e-2 # Time step for integration
time_points = np.arange(0, 80 + time_step,
    red↪ time_step) # Time points for
    red↪ simulation

output_file = "three_body_data.txt" # Output
    red↪ file name
write_to_file(output_file, 0, initial_state) #
    red↪  Write initial state to the file

# Integration and data recording
for t in range(0, len(time_points) - 2): #
    red↪ Exclude the last frame
    time_range = np.array([time_points[t],
        red↪ time_points[t + 1]])
    solution = solve_ivp(three_body_problem,
        red↪ time_range, initial_state, method
        red↪ ='DOP853', t_eval=time_range,
                    rtol=1e-12, atol=1e-12)
                        red↪ # Solve the
                        red↪ three-body
                        red↪ problem
    initial_state = solution.y[:, -1] # Update
        red↪ the initial state for the next
        red↪ iteration
    write_to_file(output_file, time_range[-1],
        red↪ initial_state, header=False) #
        red↪ Write data to the file

# Plotting
data = np.genfromtxt(output_file, delimiter=',
    red↪ ', comments='#') # Load data from the
    red↪  file
time = data[:, 0]
data = data[:, 1:]

colors = ['red', 'green', 'b'] # Colors for
    red↪ the plot
```

```
line_styles = [':', '--', '-'] # Line styles
    red↪ for the plot
font_size = 'large' # Font size for labels

text_labels = ['a', 'b', 'c', 'd', 'e', 'f', '
    red↪ g', 'h'] # Labels for subplots

fig = plt.figure(figsize=(10, 5), dpi=150) #
    red↪ Create a figure for plotting
ax_list = []
for i in range(1, 8): # Reduced the number of
    red↪ subplots
    ax_idx = 240 + i
    ax = fig.add_subplot(ax_idx)
    ax_list.append(ax)

for f in range(0, 7, 1): # Reduced the number
    red↪ of iterations
    ax = ax_list[f]
    cut_idx = np.where(np.logical_and(time >= f
        red↪  * 10, time <= (f + 1) * 10))[0]
    temp_data = data[cut_idx, :]

    if f == 0:
        init_x = [temp_data[0, j % 6] for j in
            red↪ range(0, 8, 2)]
        init_y = [temp_data[0, j % 6] for j in
            red↪ range(1, 9, 2)]
    ax.plot(init_x, init_y, 'k-', lw=2, alpha
        red↪ =0.3, zorder=2)
    ax.plot(init_x, init_y, 'k.', ms=10, alpha
        red↪ =0.3, zorder=2)

    alphas = np.linspace(0.1, 1, len(cut_idx))
        red↪ [::-1]
    for j in range(0, 3):
        ax.plot(temp_data[:, 2 * j], temp_data
            red↪ [:, 2 * j + 1], ls=
            red↪ line_styles[j], color=colors[
            red↪ j], lw=1.5, zorder=5,
                alpha=0.75) # Plotting the
                    red↪ orbits
    ax.text(0.015, 0.92, text_labels[f],
        red↪ fontweight='bold',
        red↪ horizontalalignment='left',
        red↪ transform=ax.transAxes)
    ax.text(0.6, 0.05, 't=%i-%i' % (10 * f, 10
        red↪ * (f + 1)), horizontalalignment='
        red↪ left', transform=ax.transAxes)

    if f > 3:
        ax.set_xlabel("X", fontsize=font_size)
    if f % 4 == 0:
        ax.set_ylabel("Y", fontsize=font_size)
    ax.set_xlim(-5, 5)
    ax.set_ylim(-5, 5)

fig.subplots_adjust(hspace=0.2) # Adjust
    red↪ subplot spacing
fig.savefig("3_body_orbits.png", bbox_inches='
    red↪ tight', dpi=150) # Save the figure
plt.close() # Close the figure

# Plotting the overall motion
fig, ax = plt.subplots(figsize=(5, 5), dpi
    red↪ =150)
ax.set_xlabel("X")
ax.set_ylabel("Y")

# Plotting the orbits
for j in range(0, 3):
    ax.plot(data[:, 2 * j], data[:, 2 * j + 1],
        red↪  ls=line_styles[j], color=colors[
        red↪ j], lw=1.5, zorder=5, alpha=0.75)

# Setting limits
ax.set_xlim(-5, 5)
ax.set_ylim(-5, 5)

# Save the figure
fig.savefig("Overall_3_body_motion.png",
    red↪ bbox_inches='tight', dpi=150)
plt.show()
```


C. Burrau. Numerische Berechnung eines Spezialfalles des Dreikörperproblems. *Astronomische Nachrichten*, 195(6): 113, June 1913. doi:10.1002/asna.19131950602.

Victor Szebehely and C. Frederick Peters. Complete solution of a general problem of three bodies. , 72:876, September 1967. doi:10.1086/110355.

Z. E. Musielak and B. Quarles. The three-body problem. *Reports on Progress in Physics*, 77(6):065901, June 2014. doi:10.1088/0034-4885/77/6/065901.

Tjarda C. N. Boekholt, Arend Moerman, and Simon F. Portegies Zwart. Relativistic Pythagorean three-body problem. *Phys. Rev. D*, 104(8):083020, October 2021. doi: 10.1103/PhysRevD.104.083020.

T. C. N. Boekholt, S. F. Portegies Zwart, and M. Valtonen. Gargantuan chaotic gravitational three-body systems and their irreversibility to the Planck length. , 493(3):3932–3937, April 2020. doi:10.1093/mnras/staa452.

Henri Poincaré. *Les méthodes nouvelles de la mécanique céleste*. 1892. doi:10.3931/e-rara-421.

Mauri Valtonen and Hannu Karttunen. *The Three-Body Problem*. 2006.

S. J. Aarseth, J. P. Anosova, V. V. Orlov, and V. G. Szebehely. Global Chaoticity in the Pythagorean Three-Body Problem. *Celestial Mechanics and Dynamical Astronomy*, 58(1):1–16, January 1994. doi:10.1007/BF00692114.

June Barrow-Green. The dramatic episode of sundman. *Historia Mathematica*, 37(2):164–203, 2010.

Karl F. Sundman. Mémoire sur le problème des trois corps.

Acta Mathematica, 36:105–179, 1913. URL `https://api.semanticscholar.org/CorpusID:124765639`.

Paul Painlevé. Leçons sur la théorie analytique des équations différentielles. *(No Title)*, 1897.

Tullio Levi-Civita. Traiettorie singolari ed urti nel problema ristretto dei tre corpi. *Annali di Matematica Pura ed Applicata (1898-1922)*, 9(1):1–32, 1904.

Aurel Wintner. *The analytical foundations of celestial mechanics*. Courier Corporation, 2014.

V. M. Alekseev. An Estimate of the Perturbations of Hyperbolic Motion in the Three-Body Problem. , 5:841, June 1962.

Tjarda Boekholt and Simon Portegies Zwart. On the reliability of N-body simulations. *Computational Astrophysics and Cosmology*, 2:2, March 2015. doi:10.1186/s40668-014-0005-3.

Simon F. Portegies Zwart and Tjarda C. N. Boekholt. Numerical verification of the microscopic time reversibility of Newton's equations of motion: Fighting exponential divergence. *Communications in Nonlinear Science and Numerical Simulations*, 61:160–166, August 2018. doi:10.1016/j.cnsns.2018.02.002.