

The N-Body Problem and the Stability of the Figure of 8

Antonio León Villares

Supervisor: Dr. Maximilian Ruffert

GitHub: <https://github.com/alv31415/n-body-problem>

Vacation Scholarship, 2021

College of Science and Engineering
School of Mathematics
University of Edinburgh
Summer 2021

Contents

1	Background	2
2	Formulating the N-Body Problem	2
2.1	Equations of Motion	2
2.2	Conservation Equations	3
2.2.1	Law of Conservation of Linear Momentum	3
2.2.2	Law of Conservation of Angular Momentum	3
2.2.3	Law of Conservation of Energy	3
2.2.4	Conservation of Centre of Mass Position and Velocity	3
2.3	Numerical Integration	4
2.3.1	Formulating the N-Body Problem for Numerical Integration	4
2.3.2	Euler Method	5
2.3.3	Euler-Cromer Method	5
2.3.4	2-Step Leapfrog	5
2.3.5	Synchronised Leapfrog	5
2.3.6	3-Step Leapfrog	6
2.3.7	Adaptive Timestep	6
2.3.8	Time Reversibility	6
2.3.9	Integrator Used	7
3	Periodic Solutions to the 3-Body Problem	7
3.1	Euler Orbit	7
3.2	Lagrange Orbit	8
3.3	Figure 8 Orbit	9
4	Stability of the Figure 8 Orbit	10
4.1	Constraints To Perturb The Figure 8	10
4.2	Perturbing The Figure 8	11
4.3	Visualising Stability	12
4.3.1	Methodology	12
4.3.2	Changeable Parameters	12
4.3.3	Speeding Up	13
4.3.4	The Baseline Grid	13
4.3.5	Changing <code>perturb</code>	14
4.3.6	Changing <code>n_trials</code>	16
4.3.7	Changing <code>steps</code> and <code>delta</code>	17
4.3.8	Changing <code>adaptive_constant</code>	19
4.3.9	Changing <code>delta_lim</code>	21
4.3.10	Visualising Stability of Stable Regions	22
5	Conclusion	24
6	Acknowledgements	25

1 Background

The N-Body Problem considers n bodies in \mathbb{R}^3 which interact solely via gravitational forces. A solution to the N-Body Problem will be the derivation of a set of n vector functions which describe the motion of each of the bodies. The investigations regarding the N-Body Problem began with Newton, who managed to solve for the motion of the Earth and the Moon. Later on, Euler (1765) and Lagrange (1772)[1], discovered periodic solutions to the 3-Body Problem. More recently, a third periodic solution, known as the Figure of 8 was discovered, which unlike the Euler and Lagrange orbits, is stable (that is, doesn't become chaotic under slight perturbations). Thus far, only the 2-Body Problem has been solved analytically: for instances with $N \geq 3$, no analytic solution has been found, and numerical methods are used instead.

The focus of this investigation will be to use the Python programming language to investigate how perturbations to the initial conditions of the Figure of 8 orbit affect its stability. In order to do this, we develop numerical tools to calculate, plot and analyse the effect of perturbation. We also use the 2-Body problem to validate our code.

2 Formulating the N-Body Problem

2.1 Equations of Motion

In order to be able to solve the N-Body Problem, we first need to formulate the underlying equations of motion. Consider n bodies which interact solely via gravitational forces. By Newton's Law of Universal Gravitation, given 2 bodies i and j , the gravitational force exerted on i by j is given by:

$$\underline{F}_{ij} = G \frac{m_i m_j}{\|\underline{r}_j - \underline{r}_i\|^3} (\underline{r}_j - \underline{r}_i) \quad (1)$$

where G is the gravitational constant:

$$G = 6.67408 \times 10^{-11} m^3 kg^{-1} s^{-2}$$

Furthermore, Newton's 2nd Law of Motion expresses how the acceleration felt by an object is dependent on its mass and the net amount of force exerted on said body:

$$\sum \underline{F} = m \underline{a}$$

Thus, if we consider some body i , and it interacts solely gravitationally with $n - 1$ other bodies, it follows that:

$$\sum_{j=1, j \neq i}^n \underline{F}_{ij} = m_i \underline{a}_i \quad (2)$$

Now, let $\underline{r}_i(t)$ denote the position of body i at some time t . Then, the velocity $\underline{v}_i(t)$ and the acceleration $\underline{a}_i(t)$ of the body can be expressed as:

$$\underline{v}_i(t) = \dot{\underline{r}}_i(t)$$

$$\underline{a}_i(t) = \ddot{\underline{r}}_i(t)$$

Using the above alongside, (1) and (2):

$$m_i \underline{a}_i = \sum_{j=1, j \neq i}^n \underline{F}_{ij} \quad (3)$$

$$\Rightarrow m_i \ddot{\underline{r}}_i = \sum_{j=1, j \neq i}^n G \frac{m_i m_j}{\|\underline{r}_j - \underline{r}_i\|^3} (\underline{r}_j - \underline{r}_i) \quad (4)$$

$$\Rightarrow \ddot{\underline{r}}_i = \sum_{j=1, j \neq i}^n G \frac{m_j}{\|\underline{r}_j - \underline{r}_i\|^3} (\underline{r}_j - \underline{r}_i) \quad (5)$$

This differential equation is thus the one that governs the motion of the n bodies.

2.2 Conservation Equations

Since we are attempting to model a physical system (that is, n bodies orbiting gravitationally), it is important that any solution to the set of differential equations satisfies the following conservation laws.

2.2.1 Law of Conservation of Linear Momentum

The total linear momentum of the N-Body system is given by:

$$\underline{p} = \sum_{i=1}^1 m_i \underline{v}_i$$

2.2.2 Law of Conservation of Angular Momentum

The total angular momentum of the N-Body system is given by:

$$\underline{L} = \sum_{i=1}^1 \underline{r}_i \times \underline{p}_i = \sum_{i=1}^1 m_i (\underline{r}_i \times \underline{v}_i)$$

2.2.3 Law of Conservation of Energy

Since we consider moving bodies in a gravitational field, the total energy (E) must be the sum of kinetic (EK) and gravitational potential energies (GPE).

$$\begin{aligned} EK &= \frac{1}{2} \sum_{i=1}^n m_i \|\underline{v}_i\|^2 = \frac{1}{2} \sum_{i=1}^n \frac{\|\underline{p}_i\|^2}{m_i} \\ GPE &= -G \sum_{1 \leq i < j \leq n} \frac{m_i m_j}{\|\underline{r}_j - \underline{r}_i\|} \\ E &= EK + GPE \end{aligned}$$

2.2.4 Conservation of Centre of Mass Position and Velocity

The centre of mass (COM) of the system is:

$$\underline{R} = \frac{\sum_{i=1}^n m_i \underline{r}_i}{\sum_{i=1}^n m_i}$$

which moves with velocity:

$$\underline{V} = \frac{\underline{p}}{\sum_{i=1}^n m_i}$$

It is particularly useful to consider a system in which the COM is the origin of the coordinate system, which can be achieved via the following change of coordinates to each of the bodies:

$$\underline{r}_i^* = \underline{r}_i - \underline{R}$$

$$\underline{v}_i^* = \underline{v}_i - \underline{V}$$

Under such a transformation we expect that in the calculated solution:

$$\underline{R} = \langle 0, 0, 0 \rangle$$

$$\underline{p} = \langle 0, 0, 0 \rangle$$

2.3 Numerical Integration

As discussed above, in order to solve the ODE in (5), we rely on numerical integration. Overall, we considered 5 numerical integrators.

When discussing numerical integrators, it is important to distinguish between symplectic and non-symplectic integrators, alongside time-reversible and non-time-reversible integrators.

In simple terms, a *symplectic* integrator is one which will, in particular, preserve the energy of the physical system on which it is applied. A *time-reversible* integrator is one that not only works in the forward direction (from time t you can calculate quantities at time $t + 1$), but also in the backward direction (from time $t + 1$ you can calculate quantities at time t).

Lastly, the *order* of an integrator is a measure of the accuracy of an integrator, in terms of the time step used by the integrator. For example, if an integrator uses a time step of Δt , then the error in the numerical approximation can be expressed, using asymptotic notation, in the form $\mathcal{O}(\Delta t^n)$. A numerical integrator with error $\mathcal{O}(\Delta t^n)$ is said to be of *nth* order. Thus, the higher the order of a numerical integrator, the more accurate its approximation.

2.3.1 Formulating the N-Body Problem for Numerical Integration

In general, numerical integration seeks to solve first order ODEs, of the form:

$$\frac{dy}{dt} = f(t, y)$$

The issue here is that the ODEs derived in (5) are second order:

$$\ddot{\underline{r}}_i = \sum_{j=1, j \neq i}^n G \frac{m_j}{\|\underline{r}_j - \underline{r}_i\|^3} (\underline{r}_j - \underline{r}_i)$$

However, we can transform these 3 second order differential equations, into a set of 6 first order differential equations. We can calculate the acceleration of a body by just using its position. Since acceleration is the derivative of velocity, in particular we have:

$$\ddot{\underline{r}} = f(t, \dot{\underline{r}})$$

which can be solved numerically. Similarly, velocity is the derivative of position, so:

$$\dot{\underline{r}} = f(t, \underline{r})$$

Thus, we can find $\underline{r}(t)$ by applying numerical integration twice: firstly to solve for velocity, and secondly to solve for position.

2.3.2 Euler Method

Euler's Method is a first order, non-symplectic, non-time-reversible integrator, given by the recursion:

$$\begin{aligned}\ddot{\underline{r}}_{i,t} &= \underline{a}_{i,t} = G \sum_{j=1, j \neq i}^n \frac{m_i m_j}{\|\underline{r}_{i,t} - \underline{r}_{j,t}\|^3} (\underline{r}_{j,t} - \underline{r}_{i,t}) \\ \dot{\underline{r}}_{i,t+1} &= \underline{v}_{i,t+1} = \underline{v}_{i,t} + \underline{a}_{i,t} \Delta t \\ \underline{r}_{i,t+1} &= \underline{r}_{i,t} + \underline{v}_{i,t} \Delta t\end{aligned}$$

2.3.3 Euler-Cromer Method

The Euler-Cromer Method is a first order, symplectic, non-time-reversible integrator, given by the recursion:

$$\begin{aligned}\ddot{\underline{r}}_{i,t} &= \underline{a}_{i,t} = G \sum_{j=1, j \neq i}^n \frac{m_i m_j}{\|\underline{r}_{i,t} - \underline{r}_{j,t}\|^3} (\underline{r}_{j,t} - \underline{r}_{i,t}) \\ \dot{\underline{r}}_{i,t+1} &= \underline{v}_{i,t+1} = \underline{v}_{i,t} + \underline{a}_{i,t} \Delta t \\ \underline{r}_{i,t+1} &= \underline{r}_{i,t} + \underline{v}_{i,t+1} \Delta t\end{aligned}$$

2.3.4 2-Step Leapfrog

The 2-Step Leapfrog Method is a second order, symplectic, time-reversible integrator, given by the recursion:

$$\begin{aligned}\ddot{\underline{r}}_{i,t} &= \underline{a}_{i,t} = G \sum_{j=1, j \neq i}^n \frac{m_i m_j}{\|\underline{r}_{i,t} - \underline{r}_{j,t}\|^3} (\underline{r}_{j,t} - \underline{r}_{i,t}) \\ \dot{\underline{r}}_{i,t+\frac{1}{2}} &= \underline{v}_{i,t+\frac{1}{2}} = \underline{v}_{i,t-\frac{1}{2}} + \underline{a}_{i,t} \Delta t \\ \underline{r}_{i,t+1} &= \underline{r}_{i,t} + \underline{v}_{i,t+\frac{1}{2}} \Delta t\end{aligned}$$

where to initialise $\underline{v}_{t, \frac{1}{2}}$, we can use the Euler or Euler-Cromer method for 10 iterations.

2.3.5 Synchronised Leapfrog

The Synchronised Leapfrog Method is a second order, symplectic, time-reversible integrator, given by the recursion:

$$\begin{aligned}\ddot{\underline{r}}_{i,t} &= \underline{a}_{i,t} = G \sum_{j=1, j \neq i}^n \frac{m_i m_j}{\|\underline{r}_{i,t} - \underline{r}_{j,t}\|^3} (\underline{r}_{j,t} - \underline{r}_{i,t}) \\ \underline{r}_{i,t+1} &= \underline{r}_{i,t} + \underline{v}_{i,t} \Delta t + \frac{1}{2} \underline{a}_{i,t} \Delta t \\ \dot{\underline{r}}_{i,t+1} &= \underline{v}_{i,t+1} = \underline{v}_{i,t} + \frac{1}{2} (\underline{a}_{i,t} + \underline{a}_{i,t+1}) \Delta t\end{aligned}$$

2.3.6 3-Step Leapfrog

The 3-Step Leapfrog Method is a second order, symplectic, time-reversible integrator (derived from the Synchronised Leapfrog):

$$\begin{aligned}\ddot{\underline{r}}_{i,t} &= \underline{a}_{i,t} = G \sum_{j=1, j \neq i}^n \frac{m_i m_j}{\|\underline{r}_{i,t} - \underline{r}_{j,t}\|^3} (\underline{r}_{j,t} - \underline{r}_{i,t}) \\ \dot{\underline{r}}_{i,t+\frac{1}{2}} &= \underline{v}_{i,t+\frac{1}{2}} = \underline{v}_{i,t} + \frac{\Delta t}{2} \underline{a}_{i,t} \\ \underline{r}_{i,t+1} &= \underline{r}_{i,t} + \Delta t \underline{v}_{i,t+\frac{1}{2}} \\ \dot{\underline{r}}_{i,t+1} &= \underline{v}_{i,t+1} = \underline{v}_{i,t+\frac{1}{2}} + \frac{\Delta t}{2} \underline{a}_{i,t+1}\end{aligned}$$

2.3.7 Adaptive Timestep

Whilst the above formulations typically used a fixed timestep, upon running some simulations, it soon became apparent that, as soon as the bodies sped up/got too close to each other, the constant time step led to errors. In order to mitigate this, an adaptive time step can be used. That is, at each integration step, use a time step which varies as a function of the position and velocity of the bodies in the simulation. We defined the adaptive time step at any particular step as:

$$\Delta t = c \times \min_{1 \leq j < i \leq n} \left\{ \frac{\Delta x_{ij}}{\Delta v_{ij}} \right\}$$

where, c is a constant, and for any 2 bodies i and j , Δx_{ij} corresponds to the distance between the bodies, and Δv_{ij} corresponds to the difference in speed between the bodies. Then, if the bodies go too close together, or they move at great speeds, the adaptive time step will be smaller, allowing for more accurate calculations. Similarly, for bodies far apart, or moving with small speeds, larger time steps won't heavily impact accuracy.

2.3.8 Time Reversibility

Whilst the adaptive time step can solve the issue of energy loss, it introduces a new issue: loss of symplecticity. This is caused by the fact that in using an adaptive time step, time-reversibility is lost. It is important to note that the Synchronised Leapfrog can't be used with an adaptive timestep, as it loses its symplecticity[2].

In order to preserve symplecticity and time-reversibility, the following modification can be used when using an adaptive timestep. At time t , calculate the timestep:

$$\Delta t_t = c \times \min_{1 \leq j < i \leq n} \left\{ \frac{\Delta x_{ij,t}}{\Delta v_{ij,t}} \right\}$$

Using Δt_t , apply an integration step to obtain new positions and velocities at time $t + 1$. Using these newly calculated values, compute the adaptive timestep at time $t + 1$:

$$\Delta t_{t+1} = c \times \min_{1 \leq j < i \leq n} \left\{ \frac{\Delta x_{ij,t+1}}{\Delta v_{ij,t+1}} \right\}$$

Then, for time t , use the timestep given by:

$$\Delta t^* = \frac{\Delta t_t + \Delta t_{t+1}}{2}$$

This Δt^* is now symmetric in the forward and backward time directions.

2.3.9 Integrator Used

Throughout the rest of the investigation, we decided on using the 3-Step Leapfrog. Firstly, being second-order we could use larger time steps to obtain fairly accurate orbits. Moreover, it is much simpler to initialise than 2-Step Leapfrog (requiring integration steps to initialise $\underline{v}_{i,t+\frac{1}{2}}$), and it can use adaptive time step (unlike Synchronised Leapfrog).

3 Periodic Solutions to the 3-Body Problem

As discussed above, there are 3 well-known periodic solutions to the 3-Body Problem: the Euler Orbit, the Lagrange Orbit and the Figure of 8 Orbit. In order to investigate solution stability, it was important to be able to derive the initial conditions under which the Euler and Lagrange solutions arose (Figure 8 initial conditions were found online).

At the start, we aimed to use the ODEs that defined the general motion of the 3-Body problem to, by using certain properties of the configurations, find the initial conditions/solutions. Clearly, this wasn't going to work, especially since to find analytic solutions we would need the initial conditions that I was seeking.

We then chose to use the fact that the solutions perform circular orbits, to see that in such orbits, we must have that the centripetal force ($\underline{F}_{C,i}$) of any particular body must be precisely equal to the gravitational force ($\underline{F}_{G,i}$) exerted by the other 2 bodies. Thus, for any particular body:

$$\begin{aligned}\underline{F}_{C,i} &= \underline{F}_{G,i} \\ \implies -\frac{m_i \|\underline{v}_i\|^2}{\|\underline{r}_i\|^2} \underline{r}_i &= G \frac{m_i m_x}{\|\underline{r}_x - \underline{r}_i\|^3} (\underline{r}_x - \underline{r}_i) + G \frac{m_i m_y}{\|\underline{r}_y - \underline{r}_i\|^3} (\underline{r}_y - \underline{r}_i) \\ \implies -\frac{\|\underline{v}_i\|^2}{\|\underline{r}_i\|^2} \underline{r}_i &= G \frac{m_x}{\|\underline{r}_x - \underline{r}_i\|^3} (\underline{r}_x - \underline{r}_i) + G \frac{m_y}{\|\underline{r}_y - \underline{r}_i\|^3} (\underline{r}_y - \underline{r}_i)\end{aligned}$$

Lastly, for simplicity, we can assume that all bodies are unit masses. We thus get:

$$-\frac{\|\underline{v}_i\|^2}{\|\underline{r}_i\|^2} \underline{r}_i = \frac{G}{\|\underline{r}_x - \underline{r}_i\|^3} (\underline{r}_x - \underline{r}_i) + \frac{G}{\|\underline{r}_y - \underline{r}_i\|^3} (\underline{r}_y - \underline{r}_i) \quad (6)$$

This means that (after some rearranging) we can obtain an expression for $\|\underline{v}_i\|$ purely in terms of the positions of the 3 bodies ($\underline{r}_i, \underline{r}_x, \underline{r}_y$). Moreover, since we know that \underline{v}_i must be perpendicular to \underline{r}_i in circular motion, then we can easily find \underline{v}_i :

$$\underline{v}_i = \|\underline{v}_i\| \frac{\underline{r}_i \perp}{\|\underline{r}_i \perp\|}$$

Since we are considering orbits on an xy plane, then:

$$\underline{r}_i = \langle r_x, r_y, 0 \rangle \implies \underline{r}_i \perp = \langle r_y, -r_x, 0 \rangle$$

where we have chosen $\underline{r}_i \perp$ so that it induces clockwise motion.

3.1 Euler Orbit

Consists of 3 co-linear bodies, with 2 external bodies orbiting a stationary, central one. Let \underline{r}_1 be the position vector of the central body, with \underline{r}_2 and \underline{r}_3 being the position vectors of the other 2 bodies. Then, we can assume:

$$\begin{aligned}\underline{r}_1 &= \underline{0} \\ \underline{r}_2 &= -\underline{r}_3\end{aligned}$$

$$\underline{v}_2 = -\underline{v}_3$$

Using (6), alongside the assumptions:

$$\begin{aligned} -\frac{\|\underline{v}_2\|^2}{\|\underline{r}_2\|^2}\underline{r}_2 &= \frac{G}{\|\underline{r}_1 - \underline{r}_2\|^3}(\underline{r}_1 - \underline{r}_2) + \frac{G}{\|\underline{r}_3 - \underline{r}_2\|^3}(\underline{r}_3 - \underline{r}_2) \\ \Rightarrow -\frac{\|\underline{v}_2\|^2}{\|\underline{r}_2\|^2}\underline{r}_2 &= -\frac{G}{\|\underline{r}_2\|^3}\underline{r}_2 - \frac{G}{4\|\underline{r}_2\|^3}\underline{r}_2 \\ \Rightarrow \|\underline{v}_2\|^2 &= \frac{G}{\|\underline{r}_2\|} + \frac{G}{4\|\underline{r}_2\|} \\ \Rightarrow \|\underline{v}_2\| &= \sqrt{\frac{5G}{4\|\underline{r}_2\|}} \end{aligned}$$

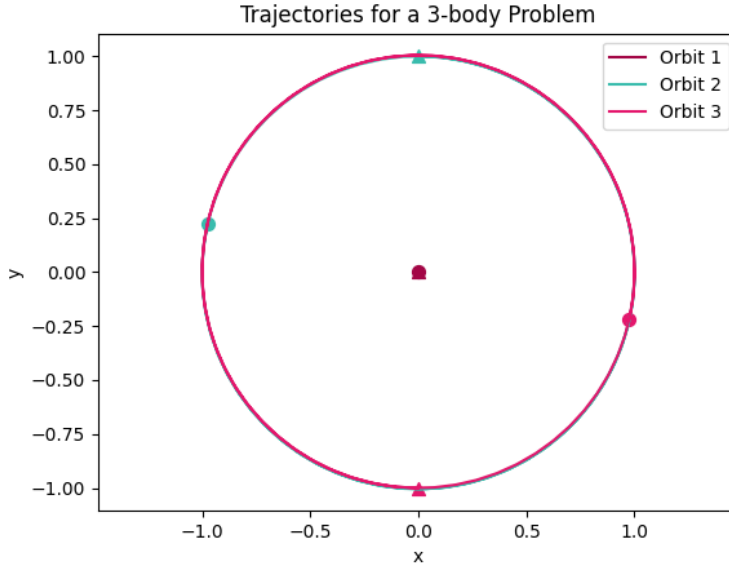


Figure 1: The Euler Orbit. Triangles represent start position. Circles represent end position.

3.2 Lagrange Orbit

Consists of 3 evenly spaced bodies around a circular orbit, forming a triangular configuration. Under the assumptions of equal mass, if we can find the velocity of one body, we can simply rotate the position and velocity vectors of this body to determine the initial conditions for the remaining 2 bodies. To do

this, we consider the clockwise rotation matrix in \mathbb{R}^3 :

$$R_\theta = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

In particular, since the bodies will be placed at the vertices of an equilateral triangle, we require $\theta = \frac{2\pi}{3}$. Let $R = R_{\frac{2\pi}{3}}$. If we are given a body at \underline{r}_1 , with corresponding velocity \underline{v}_1 , then:

$$\underline{r}_2 = R\underline{r}_1 \quad \underline{r}_3 = R^2\underline{r}_1 \quad \underline{v}_2 = R\underline{v}_1 \quad \underline{v}_3 = R^2\underline{v}_1$$

From (6), we have:

$$-\frac{\|v_1\|^2}{\|r_1\|^2}r_1 = \frac{G}{\|r_2 - r_1\|^3}(r_2 - r_1) + \frac{G}{\|r_3 - r_1\|^3}(r_3 - r_1)$$

Using our expressions for r_2 and r_3 , we can directly calculate the RHS of the equation. In particular, for the equality to hold, it must be the case that for some $k \in \mathbb{R}$:

$$\frac{G}{\|r_2 - r_1\|^3}(r_2 - r_1) + \frac{G}{\|r_3 - r_1\|^3}(r_3 - r_1) = kr_1$$

with k being easy to calculate manually.

Thus:

$$\begin{aligned} -\frac{\|v_1\|^2}{\|r_1\|^2}r_1 &= kr_1 \\ \Rightarrow \|v_1\| &= \sqrt{-k\|r_1\|^2} \\ \Rightarrow \|v_1\| &= \sqrt{-k}\|r_1\| \end{aligned}$$

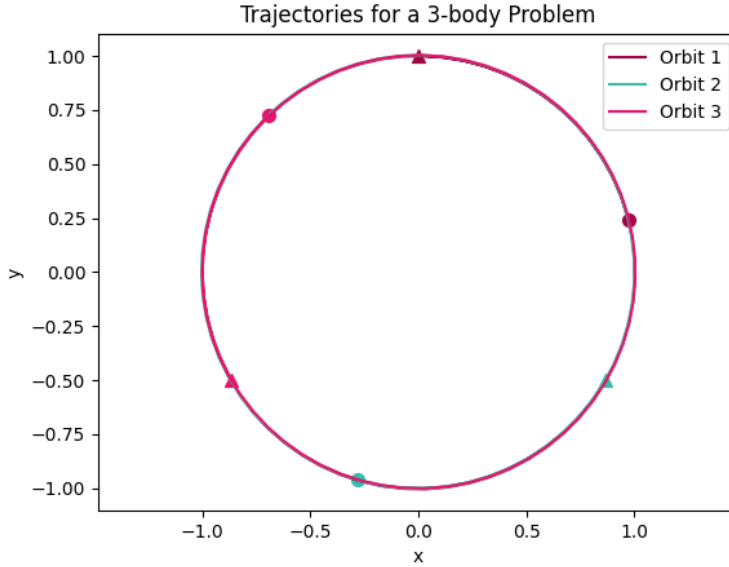


Figure 2: The Lagrange Orbit. Triangles represent start position. Circles represent end position.

3.3 Figure 8 Orbit

Consists of 3 bodies in an initially collinear configuration, with orbits “drawing” a horizontal 8. Unlike Euler or Lagrange, the Figure of 8 orbit is stable, meaning that small perturbation in the initial conditions don’t lead to chaotic orbits. For the Figure of 8, the initial conditions were taken from an article online [3], with:

$$\begin{aligned} m_1 &= m_2 = m_3 = 1 \\ r_1 &= \langle 0.97000436, -0.24308753, 0 \rangle & r_2 &= -r_1 & r_3 &= \langle 0, 0, 0 \rangle \\ v_1 &= -\frac{v_3}{2} & v_2 &= -\frac{v_3}{2} & v_3 &= \langle -0.93240737, -0.86473146, 0 \rangle \end{aligned}$$

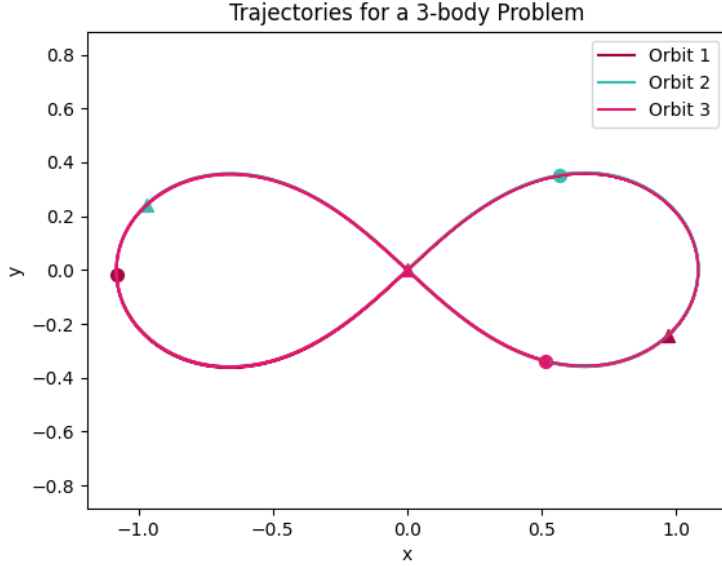


Figure 3: The Figure of 8 Orbit. Triangles represent start position. Circles represent end position.

4 Stability of the Figure 8 Orbit

In order to investigate the stability of the Figure of 8, we need to be able to perturb the initial conditions for the “standard” Figure of 8, such that all the properties of the system are maintained. In other words, we need to determine a way of obtaining new initial conditions, such that this new system has exactly the same values for energy, angular momentum, linear momentum, centre of mass position and centre of mass velocity as the original system. By doing this, we can find by how much the original system can be perturbed, before the orbits become chaotic.

4.1 Constraints To Perturb The Figure 8

We are considering a system in which the z component of every vector is simply 0. Since we have 3 bodies, this means that we will have 12 unknowns: each body has 2 position and 2 velocity components. However, the number of equations that we can make is 8: one equation for conservation of energy, one equation for conservation of angular momentum (orbiting in the xy -plane means total angular momentum only has a non-zero z component), 2 equations (one for the x and one for the y components) or each of linear momentum, centre of mass position and centre of mass velocity. If the system were linear, then we would have an underdetermined system with no solutions. However, since not all the equations are linear (for example, energy relies on kinetic energy, which is dependent on the square of the magnitude of velocity), it is possible that solutions exist.

The initial approach we followed was to attempt to algebraically calculate perturbations. However, this soon lead to problems, as perturbing a single component (for example, the x component of position of a body) meant we had to perturb 3 other components, just to satisfy one of the conservation laws. This made the resulting equations extremely hard to solve algebraically.

We then decided to focus more on symmetries. In particular, how total energy is calculated:

$$E_0 = EK + GPE$$

The key is to notice that kinetic energy (EK) is solely dependent on velocity, whilst gravitational potential energy (GPE) is solely dependent on position. This means that, since E_0 is a constant, if we

arbitrarily choose a value of velocity (which via symmetry and conservation of angular momentum, we can exploit to obtain the other 2 velocities), we would have an expression for gravitational potential energy, and thus, we'd be able to derive the position of the bodies, such that energy is conserved.

4.2 Perturbing The Figure 8

The Figure 8 consists of 3 bodies, one central, and 2 to the side. Let \underline{v}^* be the velocity of one of the side bodies. Then, if we base ourselves on the original coordinates of the Figure of 8, we can assign:

$$\underline{v}_1 = v^* \quad \underline{v}_2 = v^* \quad \underline{v}_3 = -2v^*$$

These ratios were chosen in order to satisfy linear momentum and COM velocity conservation. Then the total kinetic energy of the system will be:

$$\begin{aligned} EK &= \frac{m_1 \|\underline{v}_1\|^2}{2} + \frac{m_2 \|\underline{v}_2\|^2}{2} + \frac{m_3 \|\underline{v}_3\|^2}{2} \\ &= \frac{\|\underline{v}^*\|^2}{2} + \frac{\|\underline{v}^*\|^2}{2} + \frac{4\|\underline{v}^*\|^2}{2} \\ &= 3\|\underline{v}^*\|^2 \end{aligned}$$

Next, lets consider the gravitational potential energy. The position vectors of the 3 bodies are:

$$\underline{r}_1 = \langle x_1, y_1, 0 \rangle \quad \underline{r}_2 = -\underline{r}_1 = \langle -x_1, -y_1, 0 \rangle \quad \underline{r}_3 = \langle 0, 0, 0 \rangle$$

where \underline{r}_2 and \underline{r}_3 are derived due to the colinearity of the initial positions of the bodies, and satisfy COM conservation. Using these values, we calculate the gravitational potential energy as:

$$\begin{aligned} GPE &= -G \left(\frac{m_1 m_2}{\|\underline{r}_2 - \underline{r}_1\|} + \frac{m_1 m_3}{\|\underline{r}_3 - \underline{r}_1\|} + \frac{m_2 m_3}{\|\underline{r}_3 - \underline{r}_2\|} \right) \\ &= -G \left(\frac{2}{\|2\underline{r}_1\|} + \frac{1}{\|\underline{r}_1\|} + \frac{1}{\|\underline{r}_1\|} \right) \\ &= \frac{-5G}{2\|\underline{r}_1\|} \\ &= \frac{-5G}{2\sqrt{x_1^2 + y_1^2}} \end{aligned}$$

Using the above derivations:

$$\begin{aligned} E_0 &= EK + GPE \\ \therefore E_0 &= 3\|\underline{v}^*\|^2 - \frac{5G}{2\sqrt{x_1^2 + y_1^2}} \\ \Rightarrow x_1 &= \sqrt{\left(\frac{5G}{2(3\|\underline{v}^*\|^2 - E_0)} \right)^2 - y_1^2} \end{aligned}$$

Thus, given an arbitrary velocity, and an arbitrary y component of position, we can calculate the necessary x component of position, so as to conserve total energy of the system.

Lastly, we can show that angular momentum is also conserved. Indeed, notice that since we are using unit masses, the linear momentum of a particle is just its velocity. Thus:

$$\begin{aligned}\underline{L} &= \underline{r}_1 \times \underline{v}_1 + \underline{r}_2 \times \underline{v}_2 + \underline{r}_3 \times \underline{v}_3 \\ &= \underline{r}_1 \times \underline{v}^* + (-\underline{r}_1) \times \underline{v}^* + \underline{0} \times (-2\underline{v}^*) \\ &= \underline{0}\end{aligned}$$

which is the angular momentum of the original figure of 8 configuration.

4.3 Visualising Stability

4.3.1 Methodology

The methodology to test for stability was the following. Consider an $n \times n$ grid. Each element of the grid corresponds to a `nbody` simulation of the Figure of 8, albeit with a small perturbation in the x and y components of velocity (v_x and v_y respectively). Then, each simulation is ran for a set time, using 3-Step Leapfrog with an adaptive time step, until one of the following happened:

- **0:** Simulation Successful
- **1:** Simulation Successful, but took 10^5 steps
- **2:** Adaptive Time Step Too Small
- **3:** COM Not Conserved
- **4:** Body Escaped From COM
- **5:** Body Collision
- **6:** Figure 8 Initialisation Error
- **7:** Linear Momentum Not Conserved
- **8:** Angular Momentum Not Conserved
- **9:** Energy Not Conserved

Notice that we have assigned a numerical value to each outcome. These can then be turned into colours, and plotted in a grid to form an image, allowing us to establish for which values of the velocity components the Figure of 8 is stable.

In addition to this, we can encode extra information in the image, by considering how long a simulation lasted before an error occurred. Thus, for any error, the assigned number was given by:

$$number\ assigned\ to\ error + \left(1 - \frac{time\ until\ error}{limit\ time\ assigned\ for\ simulation}\right)$$

If $time\ until\ error \geq limit\ time\ assigned\ for\ simulation$, then a small amount was subtracted from their ratio, until it was below 1.

4.3.2 Changeable Parameters

The following parameters can be altered, as to investigate how they affect the stability of the orbit:

- **perturb:** any perturbation to v_x and v_y will be a multiple of `perturb`
- **n_trials:** the number of times that the initial conditions are perturbed to one side of 0 (so if `perturb = 0.1`, `n_trials = 10`, then v_x and v_y can be perturbed by any multiple of 0.1, from -1 to 1 (inclusive))

- `collision_tolerance`: maximum distance between 2 bodies allowed
- `escape_tolerance`: maximum distance away from the COM allowed
- `time`: desired amount of simulation time
- `tolerance`: the maximum error allowed for any calculation of a conserved quantity
- `adaptive_constant`: the constant by which we multiply when calculating adaptive time step
- `delta_lim`: the minimum value allowed for the adaptive time step

After testing, the only parameters that didn't seem to produce interesting results were `collision_tolerance`, `escape_tolerance` and `tolerance`, so the results produced keep these constant.

4.3.3 Speeding Up

As can be expected, a lot of computing power is required if we want to, for example, produce a sizeable picture (large `n_trials`), a picture with high definition (small `perturb`) or a picture with low energy errors (small `adaptive_constant`).

In order to mitigate the excessive run time, we made use of the `multiprocessing` module available in Python, as to parallelise the computations, allowing us to use the 4 CPU cores in our machine. This meant we could perform more ambitious calculations. For example, for 2 identical experiments, except for the number of `n_trials`, the one with `n_trials = 50` ran in 8 hours, whilst the one with `n_trials = 25` ran in 10 hours-

4.3.4 The Baseline Grid

In order to gauge the effect of the different parameters on the stability, we decided on having a baseline grid, from which to compare any parameter change. The grid in question was defined by the following parameters:

```
perturb = 0.05, n_trials = 10, collision_tolerance = 10-3,
escape_tolerance = 10, steps = 104, delta = 10-2,
tolerance = 10-2, adaptive_constant = 0.1, delta_lim = 10-5
```

which produces a 21×21 image, in which initial conditions are perturbed by any multiple of 0.1 on the range -1 to 1:

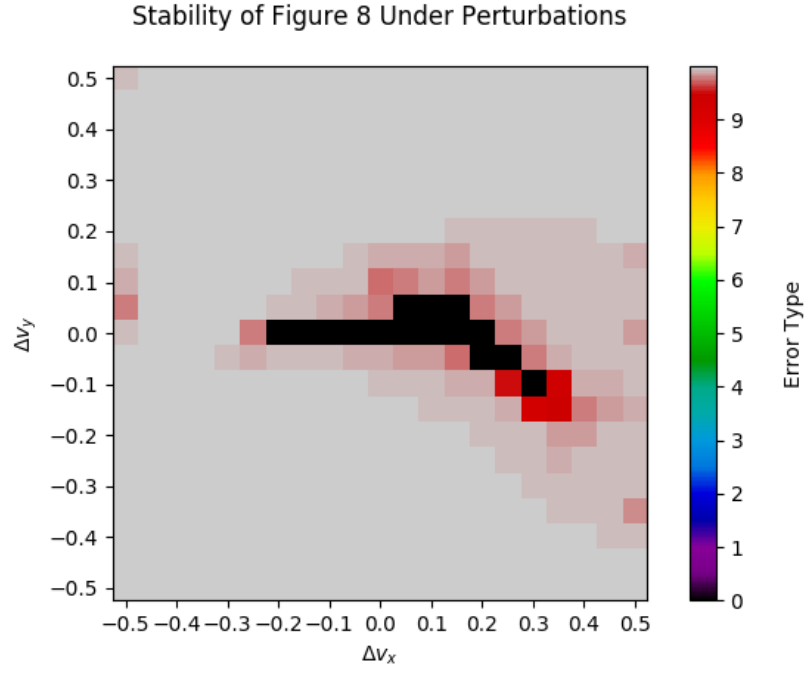


Figure 4: The standard grid. The black area in the centre represents conditions of stability, whilst the surrounding red/grey indicates an energy error.

we found that this was one of the regions on which perturbations lead to some stable orbits.

4.3.5 Changing perturb

Since we wanted to investigate the same area as the standard grid, in modifying `perturb` we also changed `n_trials`:

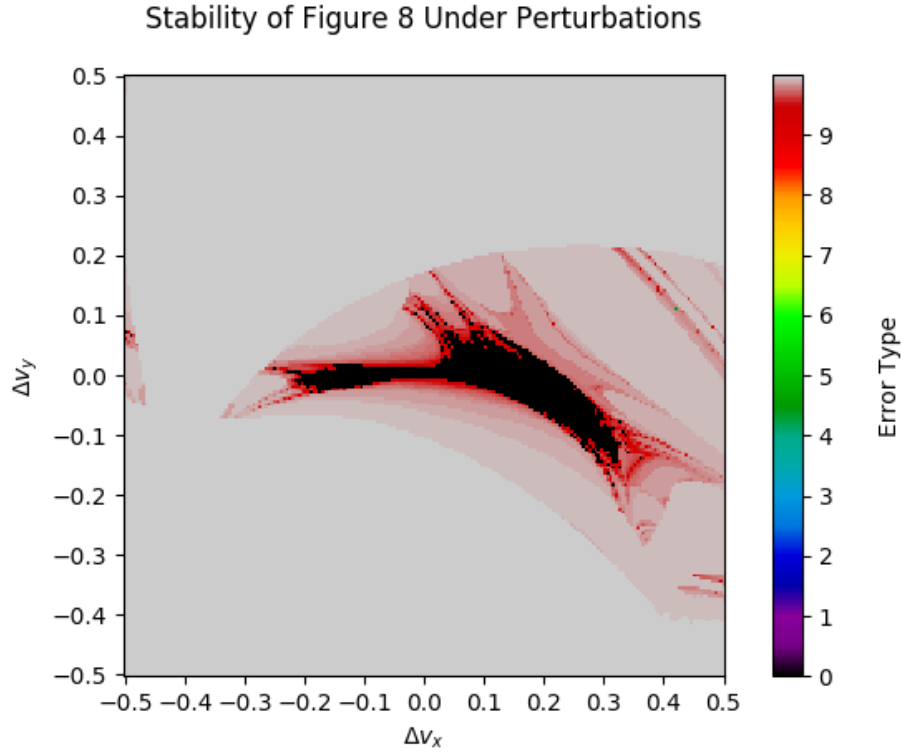


Figure 5: 201×201 image with `perturb` = 0.005, and `n.trials` = 100.

As above, black regions represent stability, whilst red/grey regions represent energy error. By decreasing `perturb` we are obtaining a higher definition image, which reveals a few interesting things. Firstly, the stability region in the centre of the plot. As in the standard grid, stability seems to be better conserved upon changes to v_x . In particular, it is interesting to note that for negative Δv_x , the region of stability seems to be thinner, allowing only very small perturbation in v_y . However, as Δv_x becomes positive, the range by which v_y can change increases drastically.

Secondly, we can now better appreciate the patterns of red surrounding the stable region. It seems to be enveloped in a curved, hook-like shape of red, containing a lot of very interesting patterns and gradients of red. It is particularly noticeable that the brightest red squares are the ones directly surrounding the stable region. These squares represent simulations in which the energy error only occurred after 90 of the 100 desired simulation time units. This is to be expected, as these regions correspond to initial conditions that differ from stable initial conditions by as little as 0.005.

Lastly, there are 2 other areas of the plot which were hinted to exist in the standard grid, but are now clearly revealed. The first is the existence of black pixels in the region of around $\Delta v_x = -0.5, \Delta v_y = 0.1$, indicating stability for these perturbations. It could be interesting to investigate whether increasing the limit time makes this region disappear, or whether increasing the window size will reveal a small cluster of stability. The second region ($\Delta v_x \in \{0.3, 0.5\}$ and $\Delta v_y \in \{0, 0.2\}$) corresponds to a set of bright diagonal lines, which, in particular, contain a pixel of green (body escaping from COM). This might mean that, should the time of execution be extended, all these long, bright red regions might lead to bodies escaping.

4.3.6 Changing `n_trials`

By changing `n_trials`, we wanted to see what happens as the perturbations grow larger, obtaining the following results:

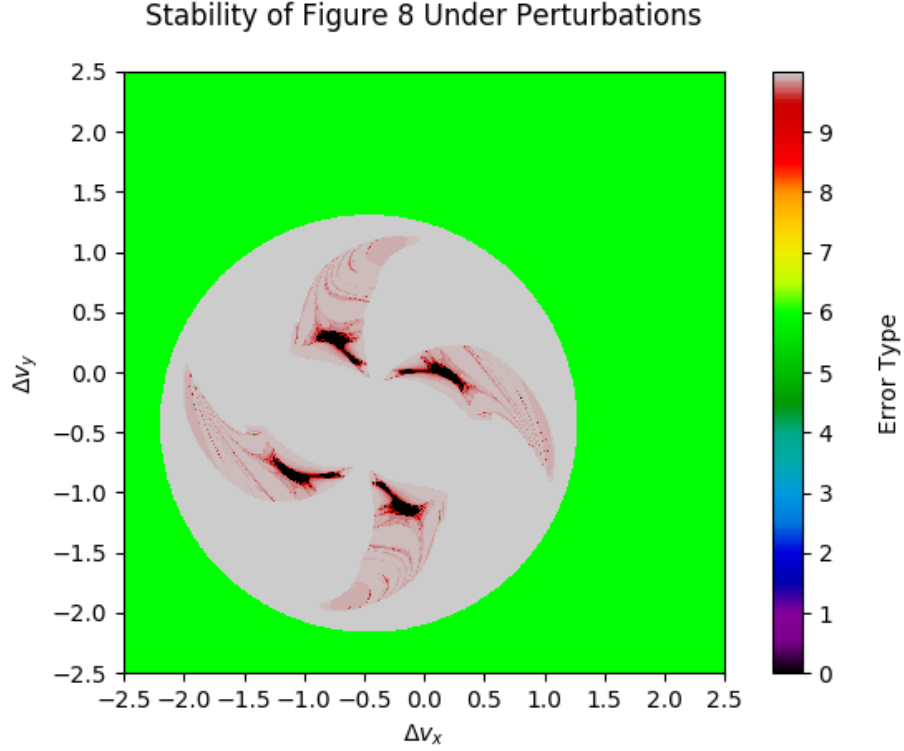


Figure 6: 1001×1001 image using `n_trials` = 500 and `perturb` = 0.005

In the above plots, we can appreciate 2 main points of interest. Firstly, and most strikingly, is the appearance of a circle of grey (energy not conserved), enveloped by a sea of green (failure to initialise the Figure of 8). We believe that this can be explained by considering how the Figure of 8 is initialised in the code. Going back to the formula:

$$x_1 = \sqrt{\left(\frac{5G}{2(3\|\underline{v}^*\|^2 - E_0)}\right)^2 - y_1^2}$$

Thus, the Figure of 8 will fail to initialise, should the expression within the square root be negative. Thus, in order to initialise a Figure of 8 according to the code, we require:

$$\left(\frac{5G}{2(3\|\underline{v}^*\|^2 - E_0)}\right)^2 - y_1^2 \geq 0$$

Since $E_0 = -1.2871419917663258 < 0$, $\frac{5G}{2(3\|\underline{v}^*\|^2 - E_0)} > 0, \forall \underline{v}^* \in \mathbb{R}^3$. Moreover, $y_1 = -0.24308753 < 0$.

Rearranging the above:

$$\begin{aligned}
& \left(\frac{5G}{2(3\|\underline{v}^*\|^2 - E_0)} \right)^2 - y_1^2 \geq 0 \\
\Rightarrow & \frac{5G}{2(3\|\underline{v}^*\|^2 - E_0)} \geq |y_1| \\
\Rightarrow & \frac{5G}{2|y_1|} \geq 3\|\underline{v}^*\|^2 - E_0 \\
\Rightarrow & \frac{5G}{6|y_1|} + \frac{E_0}{3} \geq v_x^2 + v_y^2
\end{aligned}$$

In other words, we get that v_x and v_y must be within a circle of radius $\frac{5G}{6|y_1|} + \frac{E_0}{3}$, centred at $(v_x = 0, v_y = 0)$. Now, $(v_x^0 = 0.466203685, v_y^0 = 0.43236573)$ are the velocity components of the original Figure of 8. If we want $(v_x = 0, v_y = 0)$, we then require that $(\Delta v_x = -v_x^0, \Delta v_y = -v_y^0) \therefore (\Delta v_x = -0.466203685, \Delta v_y = -0.43236573)$. Indeed, we see that this is the centre of the circles shown above, and this was further confirmed via the plotting software.

Recall that in Figure 5 we noted the existence of some black, stable pixels at $\Delta v_x = -0.5, \Delta v_y = 0.1$, and hypothesised that they could correspond to another stability region. Indeed, we see this is the case, as they correspond to the stability region in the top left of Figure 6.

The second point of interest in Figure 6 is the existence of 4 stability regions, which seem to have 180° rotational symmetry about the centre of the grey circle. In fact, it is not only the regions of stability: the surround red, hook-like areas also seem to exhibit this behaviour. This, alongside the fact the the rotation is about the centre of the grey circle, indicates that the symmetry of the 4 regions is brought by the Figure of 8 initialisation, as opposed to being a feature of regions of stability. When initialising a Figure of 8, the position of the 3 bodies are independent of the sign of v_x and v_y : y_1 never changes, and we taken to choose the positive square root for x_1 , which relies on $\|\underline{v}^*\|^2$. In other words, for any initial position, the velocity \underline{v}^* can have the following components:

1. $v_x > 0$ and $v_y > 0$
2. $v_x < 0$ and $v_y < 0$
3. $v_x < 0$ and $v_y > 0$
4. $v_x > 0$ and $v_y < 0$

That is, there are 4 different initial conditions of the Figure of 8 which start in the exact same initial position.

4.3.7 Changing steps and delta

When using non-adaptive timestep, **steps** and **delta** give the nuber of integration steps and the time step to use for integration. For adaptive time step, the time for which the simulation should run is given by **steps** \times **delta**. Thus, altering these values will alter for how long we are calculating orbits.

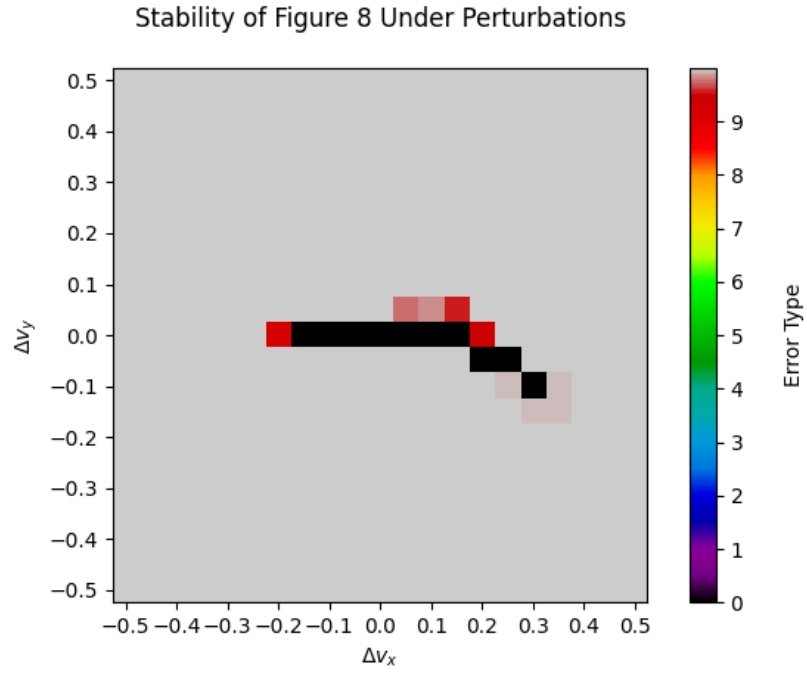


Figure 7: 21×21 image using `time` = 1,000

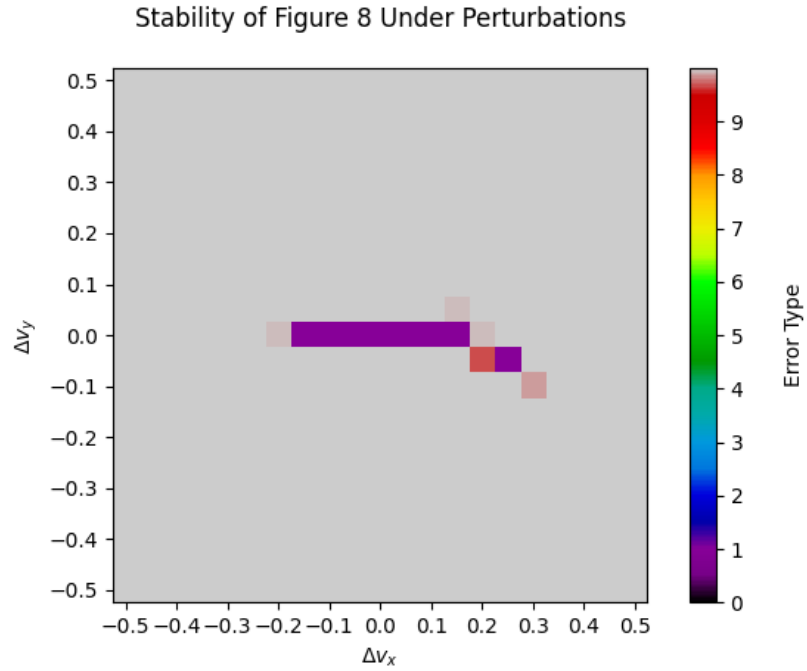


Figure 8: 21×21 image using `time` = 10,000

Comparing Figure 4, Figure 7 and Figure 8 it is easy to see that, by increasing time, the regions of stability decrease, becoming barely a line by Figure 8. In fact, in Figure 8 none of the simulations

were able to reach the target time of 10,000 (which is expected, since at most 10^5 steps were allowed). Lastly, it is important to note that the red areas surrounding the stability region progressively decrease as the time increases, furthering indicating loss of stability as simulation time progresses.

4.3.8 Changing `adaptive_constant`

In changing the `adaptive_constant` we were aiming at removing the energy errors that seem to be so omnipresent in all the above plots.

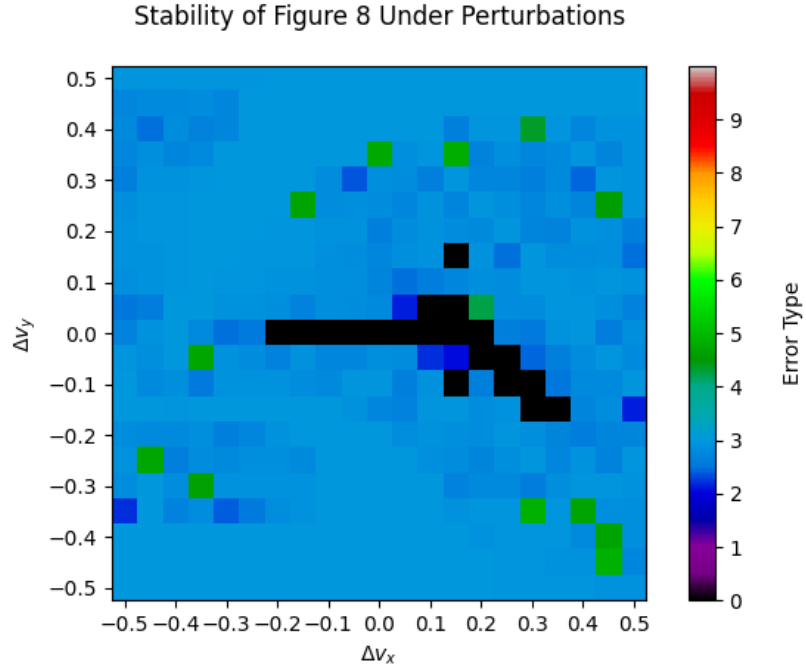


Figure 9: 21×21 image using `adaptive_constant` = 0.01

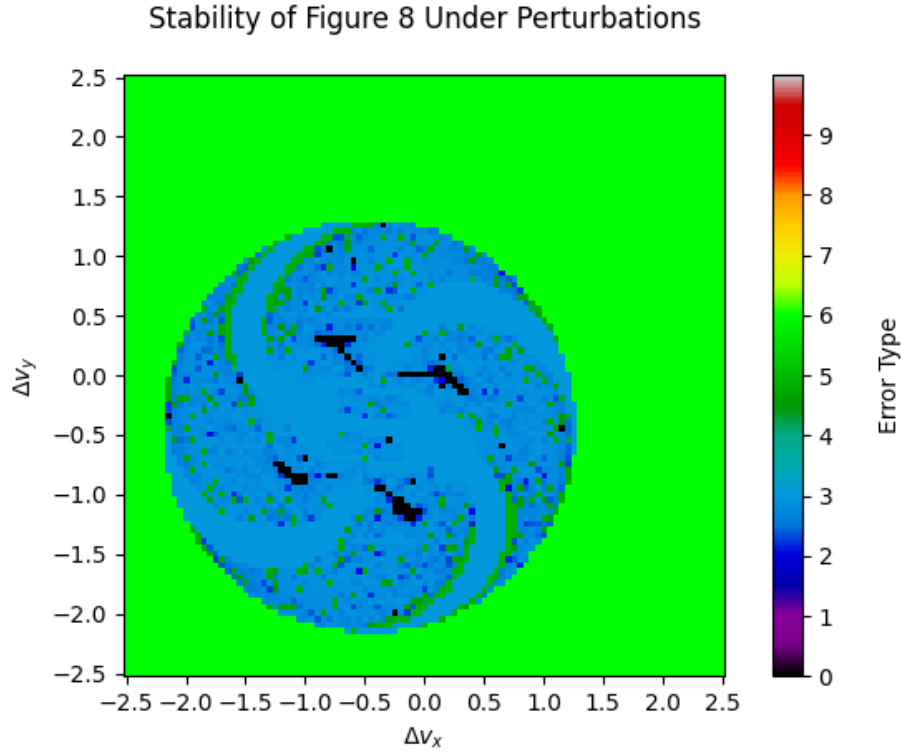


Figure 10: 101×101 image using `adaptive_constant` = 0.01

Whilst experiments were run with `adaptive_constant` = 0.001, the plots produced didn't differ much from those with `adaptive_constant` = 0.01 (this would only happen if we altered other parameters, but apart from taking longer, it wouldn't produce any new or interesting results). Comparing Figure 4 with Figure 9, we can see that all the red pixels have been substituted by blue (adaptive time step smaller than `delta_lim`) and green (body escape from COM). This is to be expected, as reducing the `adaptive_constant` we ensure that the adaptive time step is smaller, thus allowing more accurate orbit calculations. As a result, cases in which the bodies got too close or moved too fast, which previously resulted in energy non-conservation, now become blue. One particular instance of this occurs when the three bodies escape from their COM, which typically happens as a result of the 3 bodies orbiting each other very closely at high speeds, which quickly becomes unstable and leads to the bodies shooting off into space (typically one body remains alone, and the other two form a coupled orbit). The smaller adaptive time step is capable of integrating these situations, which results in an exception being raised only as the bodies escape.

Another interesting comparison between 4 and Figure 9 is the fact that the stable region has become larger in the latter, again caused by the fact that a smaller adaptive constant is capable of more faithfully integrating the orbits, thus reducing orbit instability. This can be further seen by the fact that there are a lot of dark blue patches, which indicate that errors arose later in the simulation.

Lastly, it is important to mention the structure of the blue circle in Figure 10. We can see the hook-like structures from Figure 6 surrounding the stability regions, but this time coloured in dark blue of varying tones, with some tints of green. The hook-like structures then seem to be surrounded in green lines. Lastly, we observe a smooth, light blue region, which lies between the green lines, and some of the hook-like structures. This reveals the "steps" in which the adaptive time step error appears, by considering the transition between regions. In the darker regions, the error takes longer to appear: in other words, adaptive time step is good enough at integrating these regions. This corresponds with

what we saw in Figure 6, where these regions were red, indicating more stability than in surrounding areas. It is important to note that the hook-like structures are much larger in Figure 10 than in Figure 6, which as discussed above could be caused by the fact that the smaller adaptive constant is more accurate, and can handle more chaotic situations. If we perturb sufficiently, we reach the green lines. This means that we reach fairly chaotic initial conditions, such that in a single update of the simulation, move too fast, leading to escape from the COM before the next adaptive time step is calculated. Lastly, in the light blue areas, the simulation ended very close to the start, indicating that the initial conditions lead to highly chaotic systems very soon, such that the adaptive time step quickly became too small.

4.3.9 Changing `delta_lim`

In changing `delta_lim`, we wanted to see what would happen if we allow the adaptive time step to become as small as it needed in order to integrate. Since an `adaptive_constant` of 0.1 only had energy errors, decreasing the delta limit would have no effect. However, when the `adaptive_constant` was 0.01 the main error that arose was that the adaptive time step wanted to become smaller than the `delta_lim` of 10^{-5} . Thus, to test the effect of changing `delta_lim`, we chose to set the `adaptive_constant` to 0.01.

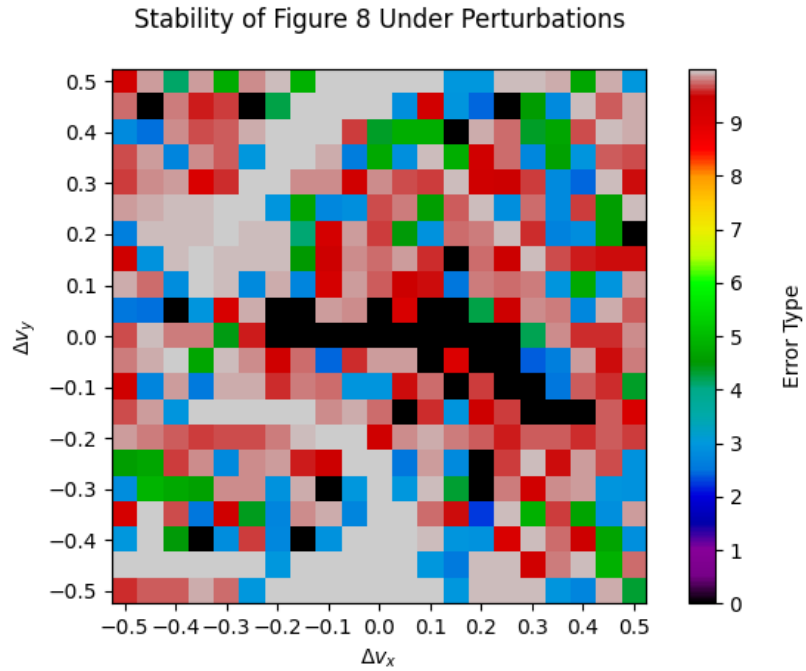


Figure 11: 21×21 image using `delta_lim` = 10^{-6}

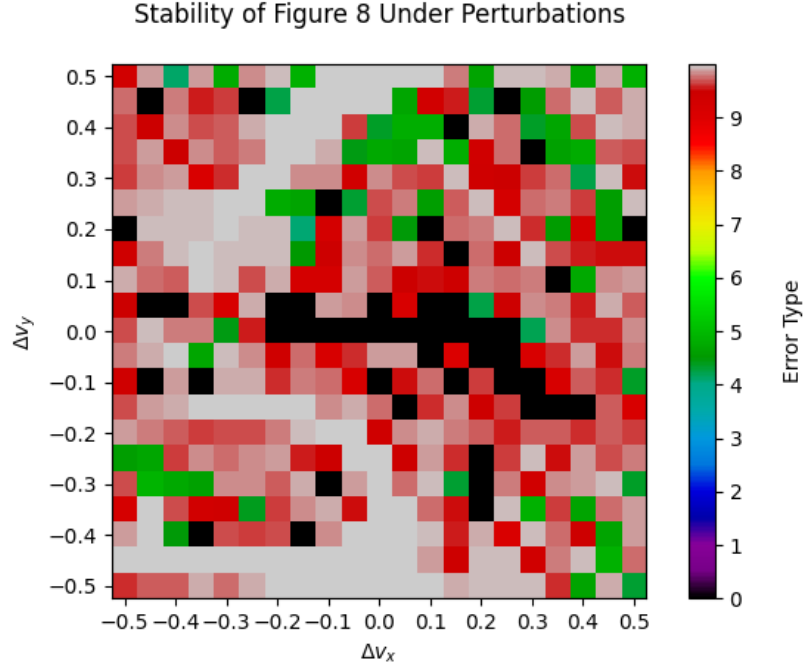


Figure 12: 21×21 image using `delta_lim` = 10^{-7}

In these plots we observe, apart from the blacks and red/greys, the appearance of green (escape from COM) and blues (adaptive time step too small). Since we are using an `adaptive_constant` of 0.01, we shall compare Figures 11 and 12 to Figure 9. It is interesting to note that across these figures, most of the blue pixels are substituted by reds/greys. This is to be expected, as a lower `delta_lim` means that there is more time for other errors to occur, in particular energy error. This is further shown by the fact that by Figure 12 there is no blue left. Secondly, if we roughly compare these figures with Figure 4, where energy error was mostly shown in grey, we observe a much higher proportion of reds, especially as `delta_lim` becomes smaller. Since the adaptive time step can become much smaller, it can better handle situations in which energy was not conserved, so it will take longer until energy error occurs. This is further supported by the fact that as `delta_lim` becomes smaller, the number of black pixels tends to increase, likely caused by the fact that we can now use a smaller adaptive time step, thus we can more accurately integrate orbits, and so, regions in which integration errors lead to instability can now be handled and calculated.

4.3.10 Visualising Stability of Stable Regions

Whilst the above indicates regions of stability, alongside regions of errors, and how unstable the error causing initial conditions are, it doesn't delve into the stability of the stable regions. In particular, for the stable initial conditions, how similar are the plots that they produce to the original Figure of 8?

In order to investigate this, we first need to have a way to compare the orbits produced, and then, we need to have a way to quantify the similarity between any orbit and the Figure of 8.

Consider a point during integration, with the 3 bodies having positions $\underline{r}_1, \underline{r}_2, \underline{r}_3$. To account for rotations and translations, we consider relative coordinates:

$$\underline{R}_1 = \underline{r}_1 - \underline{r}_2 \quad \underline{R}_2 = \underline{r}_2 - \underline{r}_3 \quad \underline{R}_3 = \underline{r}_1 - \underline{r}_3$$

To account for stretches, let $N = \|\underline{R}_1\| + \|\underline{R}_2\| + \|\underline{R}_3\|$. Then, define:

$$X_1 = \frac{\|\underline{R}_1\|}{N} \quad X_2 = \frac{\|\underline{R}_2\|}{N}$$

Notice, there is no need to consider a third coordinate, $X_3 = \frac{\|\underline{R}_3\|}{N}$, as $X_1 + X_2 + X_3 = 1$, so X_1 and X_2 already contain all the information of X_3 . Overall, X_1 and X_2 encode information about rotation, translation and stretches of orbits. What we can then do is, for any orbit, compute the set of X_1 and X_2 of each of the positions of the orbit. These X_1 and X_2 can then be plotted in a grid of squares, drawing a curve. In order to gauge the similarity of an orbit to the original Figure of 8, we can count the number of grid squares through which the resulting curve passes. Call this the *stability score*. If an orbit has a similar stability score as the Figure of 8, then we can assume that they are similar. For example:

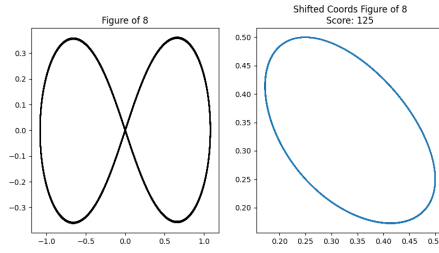


Figure 13: The original Figure of 8, with stability score of 125

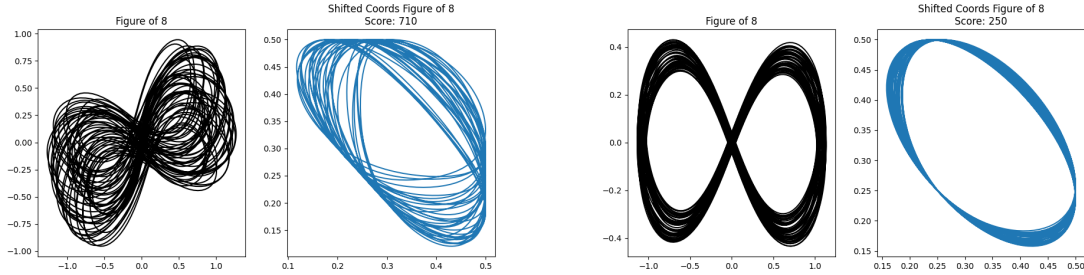


Figure 14: To the left, orbit with high stability score (710); to the right, orbit with low stability score (250). We can see the latter is more similar to the original Figure of 8.

We can incorporate the stability score into the grid, by normalising it between 0 and 1. Then, Figure 5 becomes:

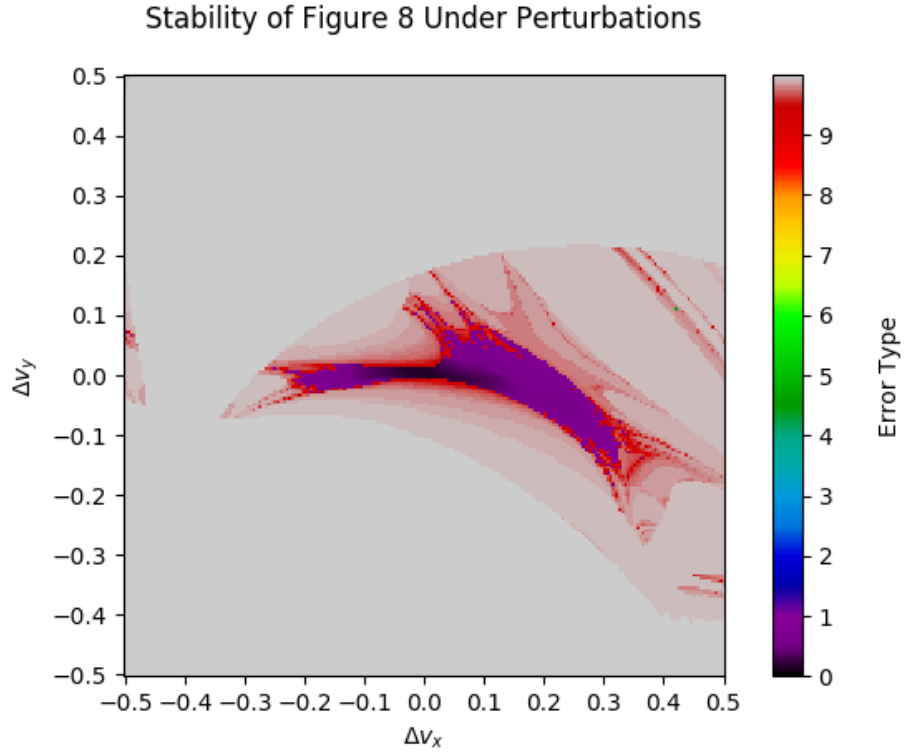


Figure 15: 201×201 image with `perturb = 0.005` and `n_trials = 100`, and displaying stability score

In the above, the darker the region the more similar to the original Figure of 8 it will be. As discussed above, we observed that the stable region seems to be more stable when perturbing \underline{v}_x . Figure 15 further confirms this: changing Δv_y by very little is sufficient to disfigure a Figure of 8, whilst changes to Δv_x tend to more easily preserve the original shape.

5 Conclusion

Overall, this project aimed to develop the necessary tools to investigate and analyse the effect of perturbation on the stability of the Figure of 8. We derived the ODEs that govern the motions of bodies which interact via gravitation. In order to numerically solve these ODEs, we developed Python code, and implemented 5 numerical integrators. For the investigation, the 3-Step Leapfrog was utilised, as it was second order, symplectic, and more versatile than 2-Step Leapfrog and Synchronised Leapfrog. To validate the code developed, the 2-Body Problem was used, as it has known solutions which can be easily checked against. Once we developed the tools necessary to work with the N-Body Problem, we focused on deriving initial conditions for 3 periodic orbit instances of the 3-Body Problem: Euler, Lagrange and the Figure of 8. We decided to focus on investigating the stability of the Figure of 8, the only stable orbit out of the 3 periodic orbits. In order to do this, we considered how to develop initial conditions that would lead to systems with the same physical properties as the Figure of 8. Then, we used this to develop perturbed versions of the Figure of 8, and investigated how different parameters affect the stability of the resulting systems.

We found that the Figure of 8 seems to be particularly stable when \underline{v}_x is perturbed. Perturbing \underline{v}_y typically leads to errors, particularly when $\Delta \underline{v}_x < 0$. We also observed that there seem to be 4 stability region, which have 180° rotational symmetry, which we conjecture is caused by how we calculate the initial conditions of the perturbed Figure of 8. Surrounding the stable regions, we can see that initial

conditions tend to be quite stable. Strangely enough, this manifests in the form of hook-like structures that tend to be more stable than their surroundings, and also showcase the 180° rotational symmetry. Due to how the initial conditions are initialised, the perturbed velocities must satisfy an inequality, meaning that any “allowed” set of perturbed velocities must lie within a circle, as otherwise the Figure of 8 won’t be initialised. Stability seems to be very affected by how long the simulation runs for, with stability regions decreasing in size, and surrounding areas becoming more chaotic. Furthermore, by decreasing the adaptive time step (either by decreasing `adaptive_constant` or `delta_lim`), we observe that stability regions tend to increase in size, with errors appearing much later on in the simulation, caused by the fact that smaller adaptive time steps lead to more accurate calculations, less prone to small integration errors, and allowing simulations to run for longer.

In general, we attempted to derive conclusions from the results observed. Due to time constraints, some of these results couldn’t be verified experimentally. Nonetheless, this provides opportunities for further investigation. Perhaps the most interesting can be what causes the 180° rotational symmetry which is apparent in images such as Figure 6, what causes the hook-like structures and the red “tendrils” surrounding the stable regions in Figure 6, or what could be causing the patterns observed in Figure 10. Further images could be developed focusing on certain regions, such as the “vertices” of Figure 5 or the lines/tendrils present within the hook-like structures of Figure 6, which could reveal fractal behaviour. Lastly, Fourier transforms, alongside action minimisation could be employed to discover new stable, periodic orbits to the 3-Body Problem (or even for the 4 or 5-Body Problem).

6 Acknowledgements

I would like to express my gratefulness to my supervisor, Dr. Maximilian Ruffert, for his support, guidance and assistance throughout the development of this project. I would also like to thank the School of Mathematics and the College of Science and Engineering, for awarding me a College Vacation Scholarship.

References

- [1] Richard Montgomery. “A New Solution to the Three-Body Problem”. In: *AMS* (May 2001). URL: <https://www.ams.org/notices/200105/fea-montgomery.pdf>.
- [2] Robert D. Skeel. “Variable step size destabilizes the Störmer/leapfrog/verlet method”. In: *Bit* 33.1 (1993), pp. 172–175. DOI: 10.1007/bf01990352. URL: <https://bionum.cs.purdue.edu/Skee93.pdf>.
- [3] Alain Chenciner and Richard Montgomery. “A remarkable periodic solution of the three-body problem in the case of equal masses”. In: *Annals of Mathematics* (2000). URL: http://emis.matem.unam.mx/journals/Annals/152_3/chencine.pdf.