Project 1: N-Body Systems

January 13th 2026

1 Code

The purpose of the code is to be able to simulate an n-body gravitationally bound system; this is achieved using a Verlet integrator. The code is made up of three main sections: simulating, plotting and running.

The first step of the code is to import a file of the initial conditions of the simulations. At this point, you can decide to run simulations or plot data you have already created, this is done by assigning a pair of variables called simulating and plotting as True if you want it to happen or False of you don't. An array contains the systems you wish to plot or simulate; this is based on their position in the initial condition file. When set to simulate the specified system is pulled from the initial conditions file and a file for the produced data is created with the initial conditions saved to it as the first step. A step of the Verlet integrator is called to, the code uses a velocity half step. Which makes use of a time step h, position x, acceleration a and velocity v.

$$v_{n+\frac{1}{2}} = v_n + \frac{1}{2}ha(x_n) \tag{1}$$

$$x_{n+1} = x_n + hv_{n+\frac{1}{2}} \tag{2}$$

$$v_{n+1} = v_{n+\frac{1}{2}} + \frac{1}{2}ha(x_{n+1}) \tag{3}$$

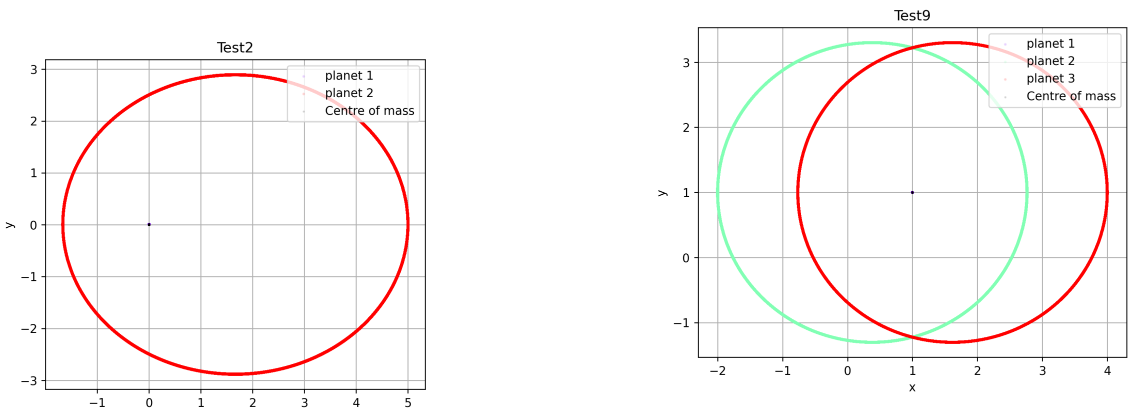
To calculate the acceleration used by the integrator, the force between the planets is calculated as a matrix; the diagonals are explicitly set to zero to prevent any attempts to divide by zero due to no separation. The rows of the matrix are summed for the total force on the bodies. When the force is used, it gets divided by the mass of the body in question. The code loops over the bodies based on the number of masses in the system, allowing it to work for any number of bodies. Full step positions are calculated then the same process is used to calculate the full step velocities. Kinetic and gravitational energies are calculated using their usual equations, $E_k = \frac{1}{2}\sum m_i v_i^2$ and $E_g = -G\sum \frac{Mm_i}{|r-r_i|}$, then combined for the total energy.

Full step velocities, positions and energy are returned to the simulating function. Angular momentum of the system is calculated using the produced positions and velocities with $L = mr \times v$, relative to position (0,0). The velocity and position are assigned to be the starting point for the next Verlet step. All the data created by the step gets saved to the data file, and the process reruns for the next step, until the system is done. Then it moves on to the next one.

Plotting is done in a separate function so that it can be run without simulating every time. The desired simulation data is pulled from the created file, and the x and y position for each body is plotted. A colour map scaled to the number of bodies is used to ensure the planets are plotted in different colours and thus easily viewable. This is then automatically titled based on the system. Plots of the system's energy and angular momentum are produced. The system's energy, the difference in energy between steps and the difference between the stated energy are plotted. Equivalent plots are also made for angular momentum.

2 Validation

As a check to see if the simulation is capable of running in different scenarios, the initial condition file contains 10 systems with different characteristics. Tests 1-3 are two body systems chosen for different-shaped orbits. 1 has $m_1 > m_2$, this was to see if the system can run multiple moving bodies. 2 and 3 are the same except for velocity, 2 doesn't have the velocity for a circular orbit, whereas 3 does. Test simulations 4 and 8 are 2 or 3 bodies in free fall towards each other, to see if the bodies trying to pass through the same point (or at least very close together) are manageable. System 9 consists of 3 bodies of equal mass, one is positioned with the intention of remaining stationary, with the intention of seeing whether matching symmetric orbits are achievable.

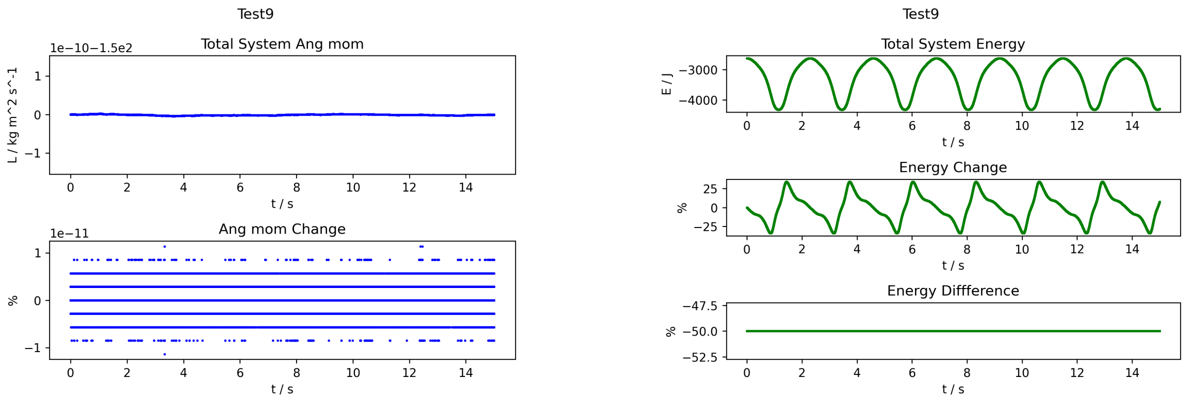


(a) Test 2, $m_1 >> m_2$

(b) Test 9, 3 bodies of equal mass, $v_1 = 0$, $v_2 = -v_3$

Figure 1: Two examples of the test simulations used in validation of the code

Since the code calculates the angular momentum and energy for every time-step, this can be used to check the reliability of the code. An orbital system should maintain its angular momentum and its total energy throughout its simulation. Energy will have a degree of oscillation about this constant energy due to the nature of the Verlet integrator. This can be seen in the plots (Figure 2b), this is a feature of these figures. The oscillation of the energy is very near constant, as can be seen by the plotted energy change between successive steps.



(a) Angular momentum of test 9, demonstrating the series of changes

(b) Shows a clear periodic structure, in keeping with a Verlet integrator

Figure 2: The energy and angular momentum of test 9

Angular momentum should also be constant in a simulation, otherwise energy must be being gained or lost. It is shown to be constant in some tests, as seen in the two-body test 4 (Figure 9c) and the three-body test 9 (Figure 2a). The change in angular momentum from step to step produces some strange patterns, such as in Test 9 (Figure 2a), where the changes between steps are sectioned into five main differences and four less common differences. Initially, I suspected that the number 9 was in some way connected to the number of masses, 3 for this test. But this is refuted by test 2 (Figure 7c), which has 13 different changes. Which leaves me unsure as to why there isn't a more continuous range of differences.

The code's ability to calculate the semi-major axis and the period allows for the testing of Kepler's third law.

$$p^2 = \frac{4\pi^2}{G(M+m)}\alpha^3 \tag{4}$$

The calculation of the period has some faults in that it requires the planet to return to the same point without the system drifting. This prevents the code from understanding some complete orbits as a complete orbit, since the drift means the separation cutoff of 10^{-5} is sometimes too small. The number of bodies which have a calculable period with this method isn't many, due to the flaws. This results in realistically too few points to get a reasonable quality line of best fit. Despite this, one has been plotted anyway, to give a rough idea of where it might be.

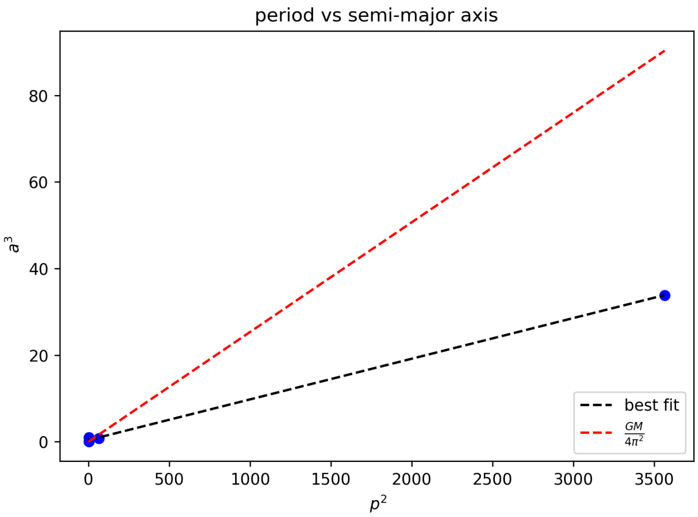


Figure 3: A plot that is meant to show the validation of Kepler's third law.

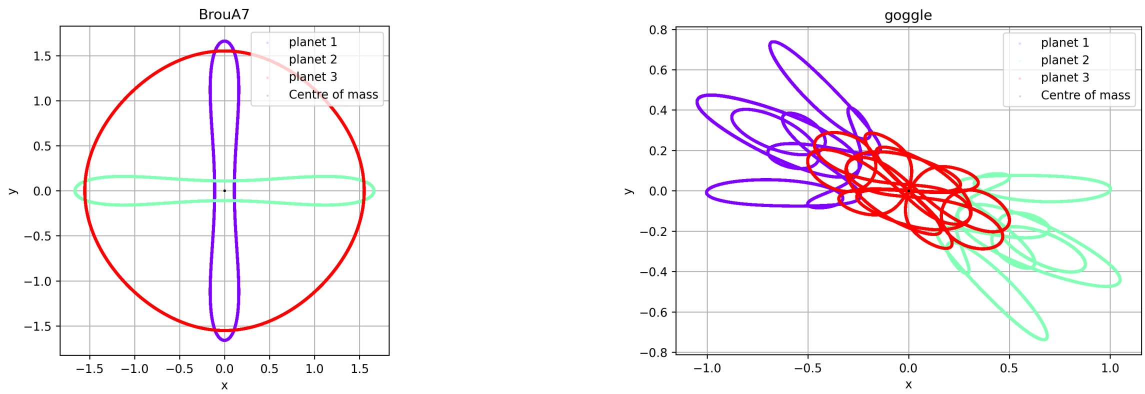
3 Three bodies

The three-body problem is a famously solvable problem for planetary orbits. Its fame originates in its lack of a general solution; this is true for all orbital systems with more than 2 bodies. Whilst there is no general solution, there are solutions. Until the invention of the computer, calculating a solution would have been incredibly difficult due to the sheer number of possible systems.

In 1973, a range of 3 body systems were found [1], These possess shapes that appear unnatural due to their unusual shapes Broucke A7 (Figure 4a) are two examples. They both possess one body in a conventional orbit, while the other two are seemingly bouncing towards and away from the centre of mass. The energy of the produced by the simulation does not match the expected value, the calculated value and the expected oscillates with its centre at roughly 25.25%.

The energy difference is high on most of the stable systems; most are less than 50% difference. Except for the system called "goggles" (googles plot and angular momentum plot), which has roughly 12 steps that are roughly $10^5 - 10^6\%$ away from the expected energy, "googles" does have a regularly oscillating total energy. The angular momentum of "googles" is nearly constant, with variations in the scale of 10^{-13}. Comparatively small angular momenta and a systematic error in the energy are common in all of the stable systems, for reasons I don't know. Given that the energy still has the periodic form expected of a Verlet integrator, it is likely a sign error. I have been unable to find an issue in the code which might explain the difference.

At a visual level, five of the systems appear to match their example counterparts[1][2]. The



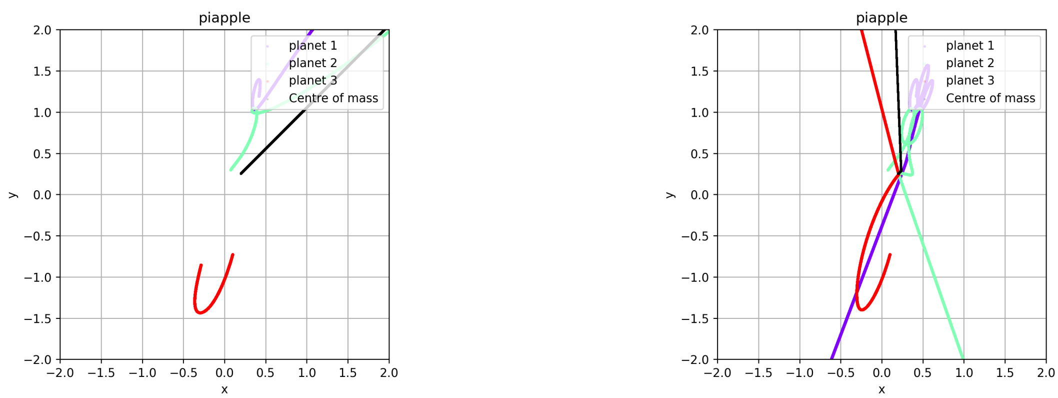
(a) Broucke A7

(b) Goggle

Figure 4: Two of the more unusually shaped orbital systems.

matching systems were “Oval with Flourishes”, “Broucke A7”, “Broucke A13”, “Figure of 8” and “goggles” (see appendix A.2). “Ovals with flourishes is an example of a simpler system (thus easier to visually compare) that appears to fit the same shape.

For several of the systems, the time steps and damping coefficients could not be found. For the system “pineapple” [3] a number of timesteps were tested, ranging from 0.1 to 0.00001. But a full orbit could not be achieved; as a result, damping was added, and the time steps were retested. A range of damping from $10^{-2} - 10^{-6}$ was attempted. But it still couldn’t produce a stable system.



(a) Pineapple with $timestep = 10^{-5}$ and $damping = 10^{-3}$

(b) Pineapple with $timestep = 10^{-5}$ and $damping = 10^{-5}$

Figure 5: A comparison of pineapple with different levels of damping

References

[1] R Broucke and D Boggs. "Periodic orbits in the planar general three-body problem". In: *Celestial mechanics* 11.1 (1975), pp. 13–38.

[2] Milovan Šuvakov and Veljko Dmitrašinović. "Three classes of Newtonian three-body planar periodic orbits". In: *Physical review letters* 110.11 (2013), p. 114301.

[3] Matthew Sheen. *MAE5730*. Sept. 2016. URL: `https://github.com/mws262/MAE5730_examples/tree/cbf76ef568366403affc44790dbfc81a5af32d1d/3BodySolutions`.

A Appendix

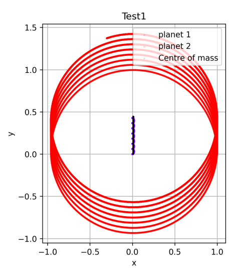
A.1 Initial Conditions

Table 1: Three-body problem initial conditions and parameters

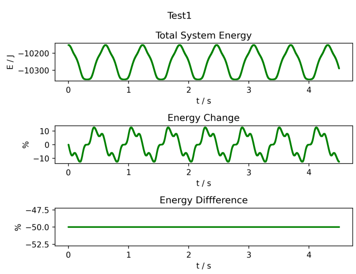
Name	x_1	y_1	x_2	y_2	x_3	y_3	v_{x1}	v_{y1}	v_{x2}	v_{y2}	v_{x3}	v_{y3}	m_1	m_2	m_3	Δt	ϵ	Period	Energy	Ang Mom
Test1	0	0	1	0	0	0	0	0	0	10	0	0	100	1	0	0.0001	0	3	0	0
Test2	0	0	5	0	0	0	0	0	0	100	0	0	100000	1	0	0.00001	0	3	0	0
Test3	0	0	2.5	0	0	0	0	0	0	200	0	0	100000	1	0	0.00001	0	3	0	0
Test4	1	1	-1	-1	0	0	0	0	0	0	0	0	1000	1000	0	0.0001	0	1	0	0
Test5	1	0	-1	0	0	0	0	1	0	-1	0	0	10	10	0	0.0001	0	3	0	0
Test6	1	3	2	2	0	0	3	2	1	2	0	0	2	2	0	0.0001	0	10	0	0
Test7	1	1	-1	-2	3	-2	1	1	2	2	-1	-2	1	1	1	0.0001	0	10	0	0
Test8	0	0	2	0	1	1.732	0	0	0	0	0	0	5	5	5	0.0001	0	2	0	0
Test9	1	1	-2	1	4	1	0	0	0	5	0	-5	100	5	5	0.0001	0	10	0	0
Test10	1	1	-2	1	4	1	1	-0.5	0	5	0	-5	100	5	5	0.0001	0	10	0	0
BrouA7	-0.110	0	1.661	0	-1.552	0	0	0.991	0	-0.157	0	-0.834	1	1	1	0.0001	0	12.056	-0.718	0
BrouA13	-0.897	0	3.235	0	-2.339	0	0	0.829	0	-0.006	0	-0.823	1	1	1	0.0001	0	59.716	-0.433	0
FigOf8	-1	0	1	0	0	0	0.347	0.533	0.347	0.533	-0.694	-1.065	1	1	1	0.001	0	6.324	-1.287	0
Bfly1	-1	0	1	0	0	0	0.307	0.126	0.307	0.126	-0.614	0.251	1	1	1	0.00001	0.0001	6.236	-2.170	0
Moth1	-1	0	1	0	0	0	0.464	0.396	0.464	0.396	0.929	0.792	1	1	1	0.0001	0	14.894	-1.382	0
OvCaSt	0.536	0.054	0.252	0.695	-0.276	-0.336	-0.569	1.255	0.080	-0.459	0.490	-0.797	1	1	1	0.00001	0.0001	5.026	0	0
piapple	0.420	1.190	0.076	0.296	0.100	-0.729	0.102	0.687	0.149	0.240	-0.251	-0.927	1	1	1	0.00001	0.001	5.095	0	0
Oflour	0.716	0.384	0.086	1.343	0.539	0.481	1.245	2.444	-0.675	-0.963	-0.570	-1.481	1	1	1	0.00001	0	8.095	0	0
goggle	-1	0	1	0	0	0	0.083	0.128	0.083	0.128	-0.167	-0.256	1	1	1	0.00001	0	10.467	-2.430	0
pt1	0.708	0.473	0.168	-0.058	-0.507	-0.307	0.824	0.522	-0.077	-0.167	-0.747	-0.355	1	1	1	0.00001	10^{-6}	8	0	0
pt2	0.865	0.629	0.085	0.013	-0.091	-0.892	0.289	0.171	-0.220	0.090	-0.068	-0.262	1	1	1	0.00001	10^{-6}	5.428	0	0
Drgfly	-1	0	1	0	0	0	0.081	0.589	0.081	0.589	0.161	1.178	1	1	1	0.00001	0.001	21.271	-1.440	0

A.2 Figures

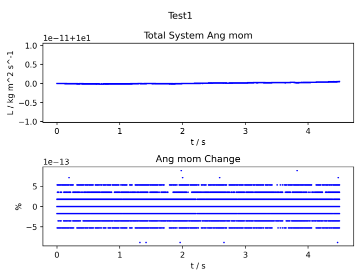
Within this appendix are plots produced by the code.



(a)

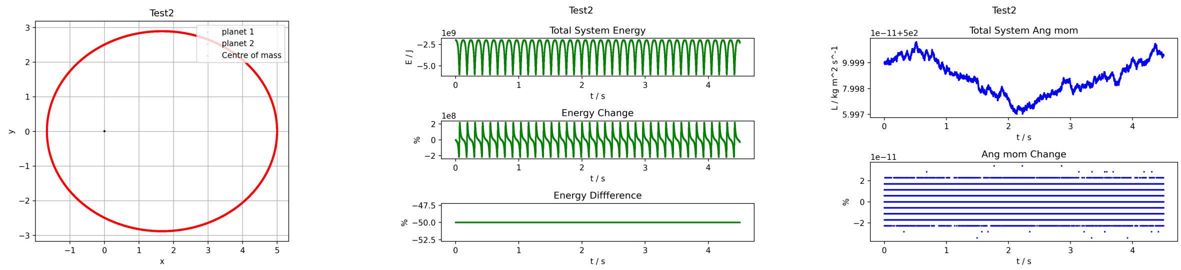


(b)



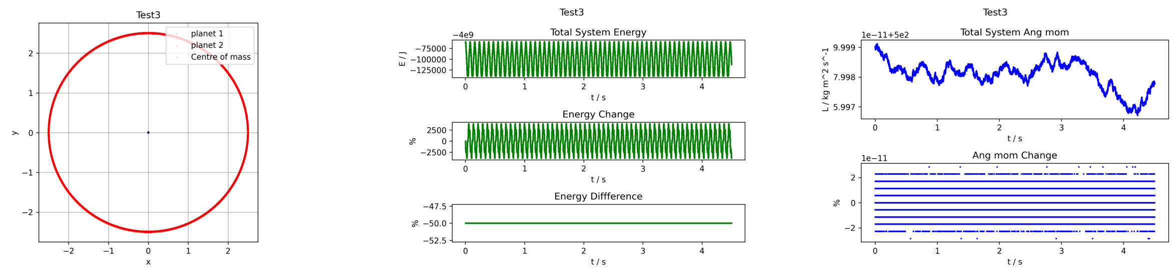
(c)

Figure 6: Test 1



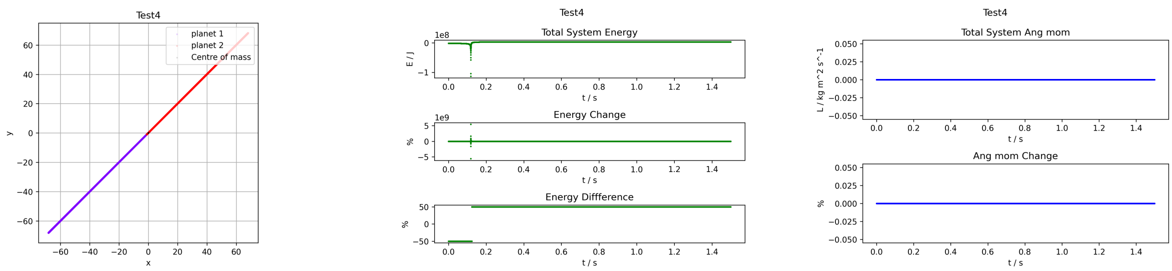
(a) **(b)** **(c)**

Figure 7: Test 2



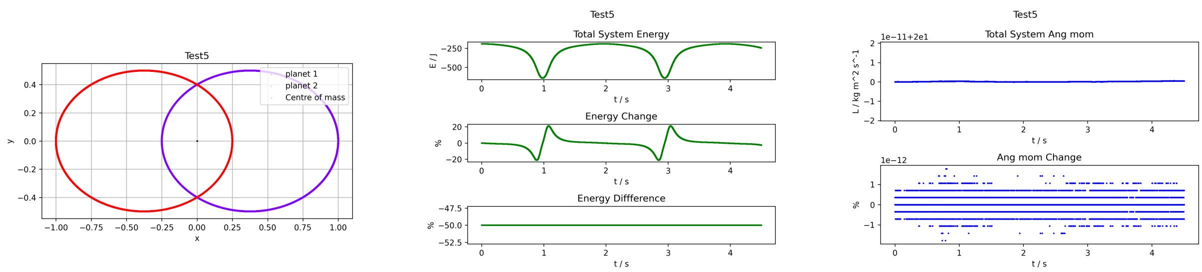
(a) **(b)** **(c)**

Figure 8: Test 3



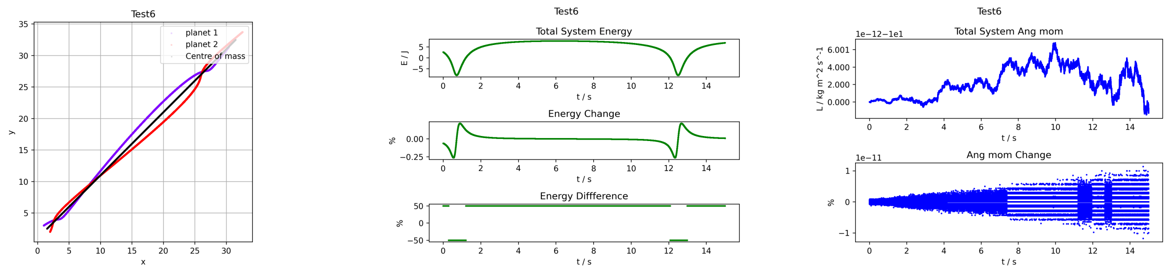
(a) **(b)** **(c)**

Figure 9: Test 4



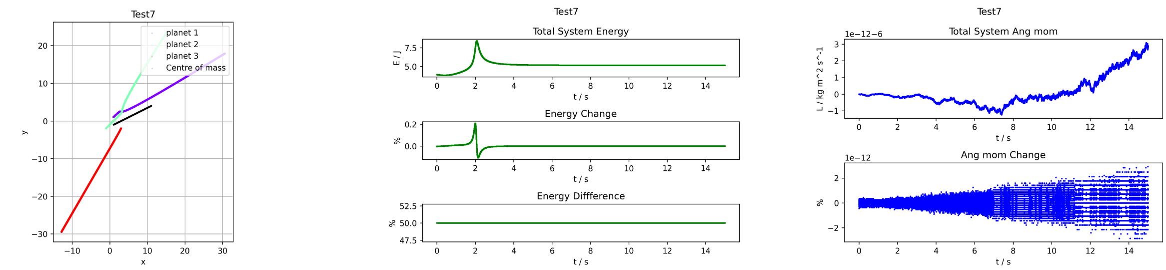
(a) **(b)** **(c)**

Figure 10: Test 5



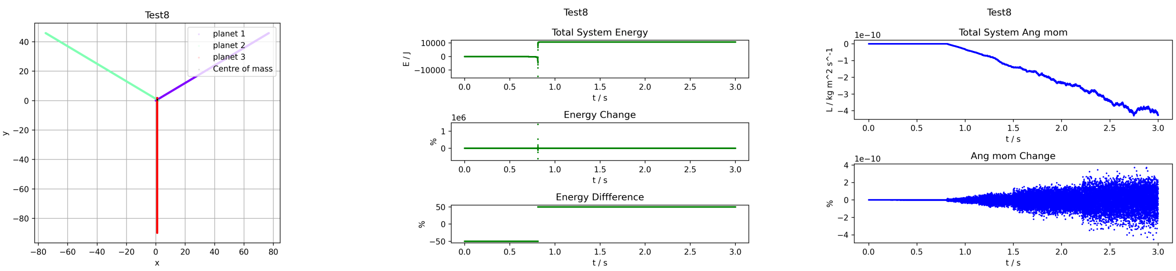
(a) (b) (c)

Figure 11: Test 6



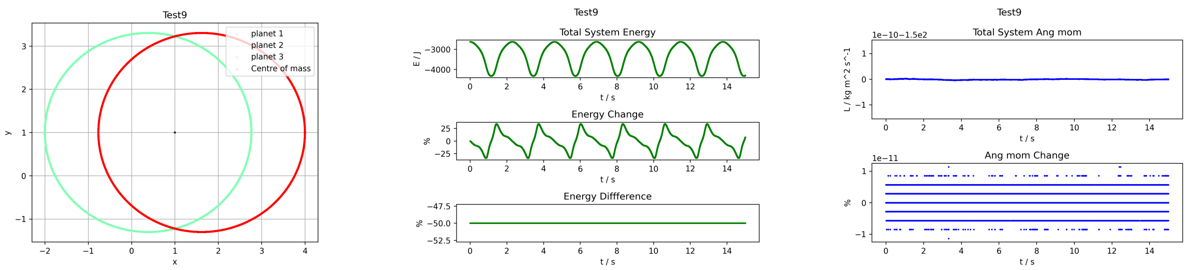
(a) (b) (c)

Figure 12: Test 7



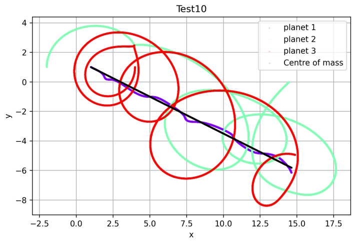
(a) (b) (c)

Figure 13: Test 8

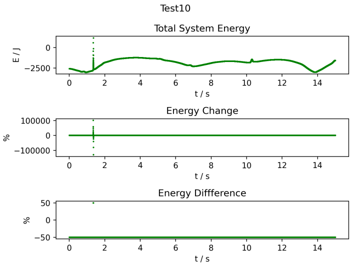


(a) (b) (c)

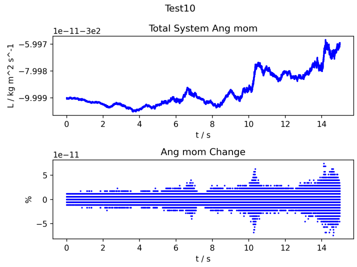
Figure 14: Test 9



(a)

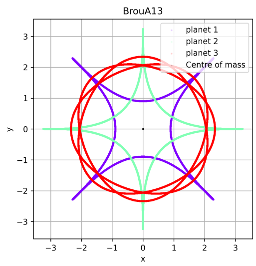


(b)

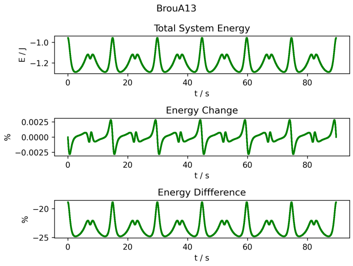


(c)

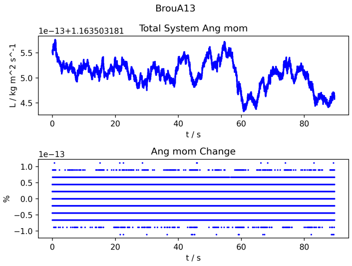
Figure 15: Test 10



(a)

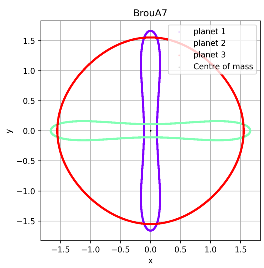


(b)

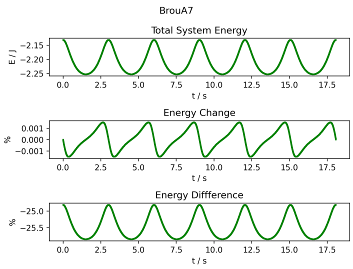


(c)

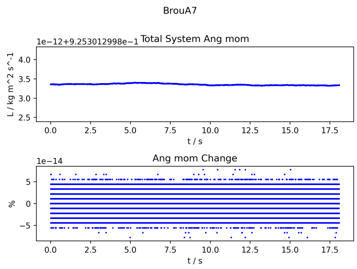
Figure 16: Broucke A13



(a)

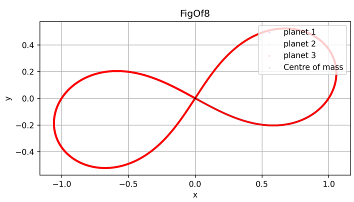


(b)

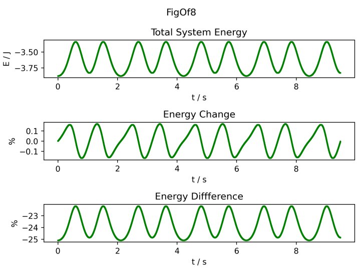


(c)

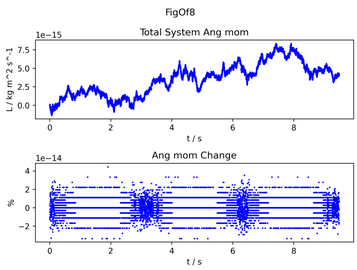
Figure 17: Broucke A7



(a)

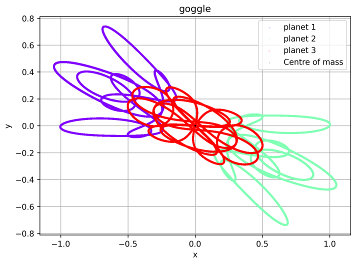


(b)

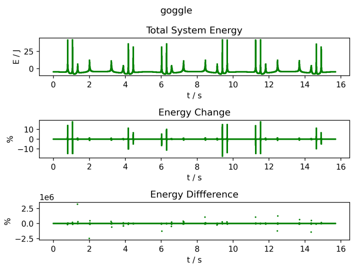


(c)

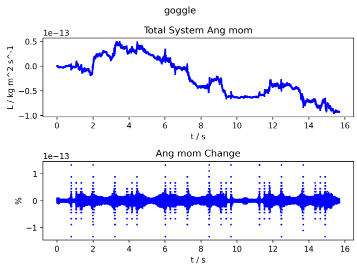
Figure 18: Figure of 8



(a)

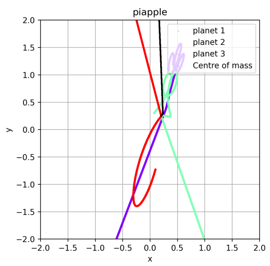


(b)

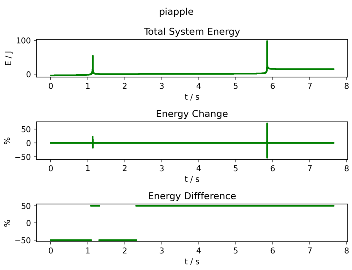


(c)

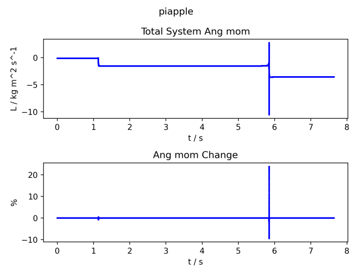
Figure 19: googles



(a)

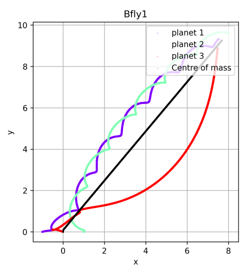


(b)

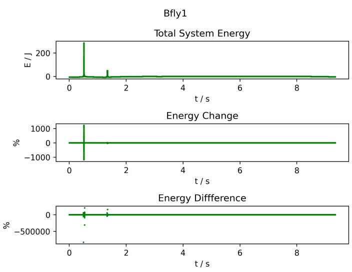


(c)

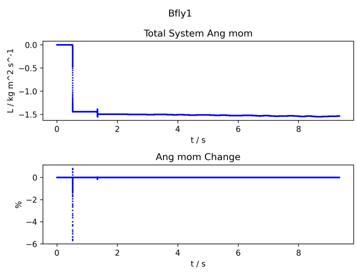
Figure 20: pineapple



(a)



(b)



(c)

Figure 21: moth 1 is another system where I was unable to find condtions for a stable system