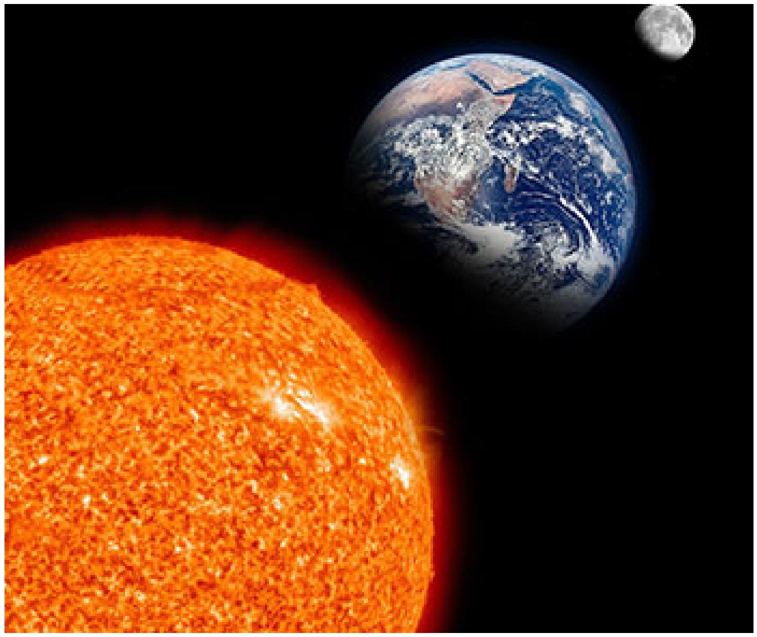
COSC2500 Semester 2, 2019
Project - 3 body problem
William Jago - 44827014



Contents

1 Introduction to the 3 body problem

In classical mechanics, the manner in which large objects (particularly celestial bodies) interact with one another has been one of the most significant problems in historical physics. The most basic form of the problem is known as the central-force or one body problem, where the motion of a smaller object is influenced by a large, immovable object. By considering both the affect of the large object on the small object and vice versa, but with no other external forces, the problem becomes the two body problem. This can then become the three body problem (or more generally the n body problem), which analyses the effect of three bodies on each other, The Earth interacting with the sun and moon is an example of such a problem.

This report will investigate the three body problem with respect to the motion of three masses by constructing a system of equations to model the motion, then solving it numerically and simulating in MATLAB. There is no general solution which can be found analytically, so numerical methods must be used for investigating the motion of the three bodies.

2 Derivation of equations

Newton's law of gravity states that the gravitational force on a given body with mass m_1 by another body with mass m_2 is given by

$$F_1 = \frac{gm_1m_2}{r^2}$$

where g is the acceleration due to gravity (9.8ms^{-2}) and r is the distance separating the bodies. A third body with mass m_3 can also be introduced

$$F_1 = \frac{gm_1m_2}{r_1^2} + \frac{gm_1m_3}{r_2^2}$$

The equations for the other two bodies follow similarly

$$F_2 = \frac{gm_1m_2}{r_1^2} + \frac{gm_2m_3}{r_3^2}$$
$$F_3 = \frac{gm_1m_3}{r_2^2} + \frac{gm_2m_3}{r_3^2}$$

For now, the first body will be focused on. The force can be broken up into component vectors F_x and F_y. The values of these vectors are

$$F_{1x} = \frac{gm_1m_2(x_2 - x_1)}{r_1 \cdot r_1^2} + \frac{gm_1m_3(x_3 - x_1)}{r_1 \cdot r_2^2}$$
$$F_{1y} = \frac{gm_1m_2(y_2 - y_1)}{r_1 \cdot r_1^2} + \frac{gm_1m_3(y_3 - y_1)}{r_1 \cdot r_2^2}$$

This is because the direction vector from mass 1 to mass 2 is given by $\left(\frac{x_2-x_1}{r_1}, \frac{y_2-y_11}{r_1}\right)$, and similarly for 1 to 3. Substitute in the formula for the distance $r = \sqrt{x^2 + y^2}$ and we obtain

$$F_{1x} = \frac{gm_1m_2(x_2 - x_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}^3} + \frac{gm_1m_3(x_3 - x_1)}{\sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2}^3}$$
$$F_{1y} = \frac{gm_1m_2(y_2 - y_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}^3} + \frac{gm_1m_3(y_3 - y_1)}{\sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2}^3}$$

Newton's second law states that $F = ma$, where a is the objects acceleration. Splitting a into its component vectors, $a_x = x''$ and $a_y = y''$. The equations become

$$m_1x_1'' = \frac{gm_1m_2(x_2 - x_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}^3} + \frac{gm_1m_3(x_3 - x_1)}{\sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2}^3}$$
$$m_1y_1'' = \frac{gm_1m_2(y_2 - y_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}^3} + \frac{gm_1m_3(y_3 - y_1)}{\sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2}^3}$$
$$\therefore x_1'' = \frac{gm_2(x_2 - x_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}^3} + \frac{gm_3(x_3 - x_1)}{\sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2}^3}$$

$$y_1'' = \frac{gm_2(y_2 - y_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2)}^3} + \frac{gm_3(y_3 - y_1)}{\sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2)}^3}$$

We now have two second order ODE's describing the motion of the first body. These can be rewritten as two coupled first order ODE's by substituting $y' = v_y$, $x' = v_x$.

$$\begin{aligned}
x_1' &= v_{1x} \\
v_{1x}' &= \frac{gm_2(x_2 - x_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2)}^3} + \frac{gm_3(x_3 - x_1)}{\sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2)}^3} \\
y_1' &= v_{1y} \\
v_{1y}' &= \frac{gm_2(y_2 - y_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2)}^3} + \frac{gm_3(y_3 - y_1)}{\sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2)}^3}
\end{aligned}$$

The ODE's for the second and third bodies are found similarly. For the second body:

$$\begin{aligned}
x_2' &= v_{2x} \\
v_{2x}' &= \frac{gm_1(x_1 - x_2)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2)}^3} + \frac{gm_3(x_3 - x_2)}{\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2)}^3} \\
y_2' &= v_{2y} \\
v_{2y}' &= \frac{gm_1(y_1 - y_2)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2)}^3} + \frac{gm_3(y_3 - y_2)}{\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2)}^3}
\end{aligned}$$

For the third body:

$$\begin{aligned}
x_3' &= v_{3x} \\
v_{3x}' &= \frac{gm_1(x_1 - x_3)}{\sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2)}^3} + \frac{gm_3(x_2 - x_3)}{\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2)}^3} \\
y_3' &= v_{3y} \\
v_{3y}' &= \frac{gm_2(y_1 - y_3)}{\sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2)}^3} + \frac{gm_3(y_2 - y_3)}{\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2)}^3}
\end{aligned}$$

3 Application to real world scenario

An important physical example of the three body problem involves the Earth, Moon and a smaller projectile, such as a satellite, space shuttle or moon lander. Solving the three body problem is necessary to find the forces exerted by the Earth and the Moon on the vessel, and would have been vital for landing spacecraft on the moon for example.

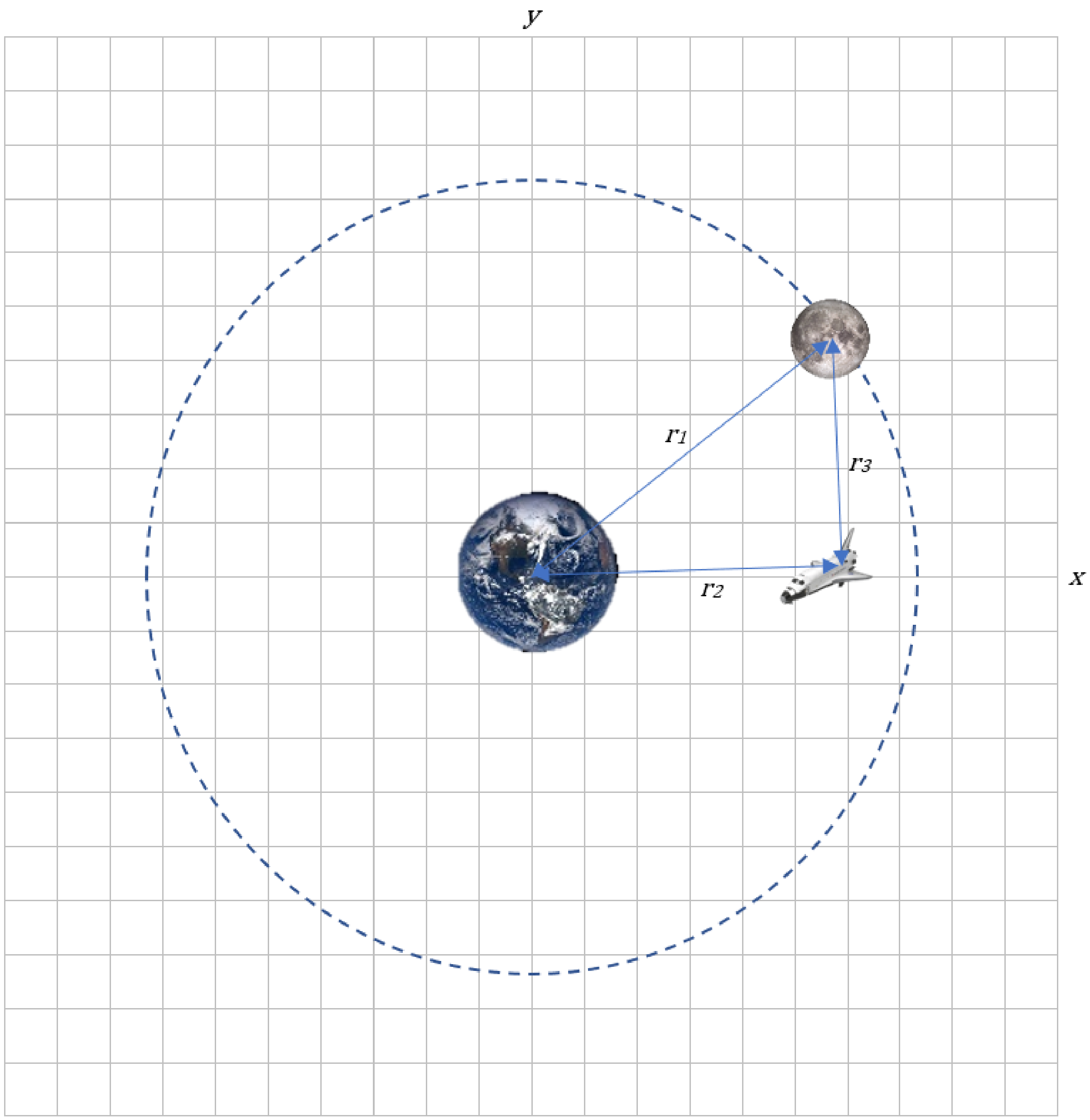


Figure 1: Diagram of the Earth, Moon and spacecraft in the restricted three body problem

This project will assume the three bodies lie on the same plane with only x and y coordinates so that the equations found previously are applicable. For a projectile launched in the direction of the moon, this is not too unreasonable assumption to make.

4 Discussion of script used

The following MATLAB script was written, loosely based on the script from Sauer 6.3. In that chapter, the ydot function was only for a single body of 0 mass rotating around a fixed large body. The ydot function had to be rewritten to accommodate the three bodies moving with each other. The rest of the code to solve and animate the system is entirely original.

```
%change these lines
inter = [0 30];
%x y initial position for mass 1 2 3
ic1 = [0 0 4 4 6 -1];
%x y initial velocity for mass 1 2 3
ic2 = [0 0 0 0 0 0];

%solve and plot
ic = [ic1 ic2];
[t, u] = ode45(@(T,U) ydot(T,U),inter, ic);

figure(1);
vid = VideoWriter('threebod1.avi');
vid.FrameRate = 120;
open(vid);
nextt = 0;
for i = 1:length(u(:,1))
    plot(u(i,1),u(i,2),'ob',...
        u(i,3),u(i,4),'og',...
        u(i,5),u(i,6),'or');
    axis([min([u(:,1);u(:,3);u(:,5)])...
        max([u(:,1);u(:,3);u(:,5)])...
        min([u(:,2);u(:,4);u(:,6)])...
        max([u(:,2);u(:,4);u(:,6)])]);
    title(sprintf('Bodies at t=%f',t(i)));
    xlabel('x');ylabel('y');
    if t(i) >= nextt
        frame = getframe;
        writeVideo(vid,frame);
        nextt = nextt + 1/60;
    end
end
close(vid);

figure(2)
plot(u(:,1),u(:,2),'.b',u(:,3),u(:,4),'.g',...
    u(:,5),u(:,6),'.r');
title("Position of three bodies over time");
```

```
xlabel("x");ylabel("y");

figure(3)
plot(t, (u(:,7).^2+u(:,8).^2).^(1/2),t, ...
    (u(:,9).^2+u(:,10).^2).^(1/2),t, ...
    (u(:,11).^2+u(:,12).^2).^(1/2))
title("Velocity of three bodies over time");
xlabel("t");ylabel("v");

function z=ydot(t,x)
%also change these lines
g=1;
m1=1; mg1=m1*g;
m2=1; mg2=m2*g;
m3=1; mg3=m3*g;

px1=x(1);py1=x(2);vx1=x(7);vy1=x(8);
px2=x(3);py2=x(4);vx2=x(9);vy2=x(10);
px3=x(5);py3=x(6);vx3=x(11);vy3=x(12);

dist1=sqrt((px2-px1)^2+(py2-py1)^2);
dist2=sqrt((px3-px1)^2+(py3-py1)^2);
dist3=sqrt((px2-px3)^2+(py2-py3)^2);
z=zeros(12,1);

z(1,1)=vx1;
z(7,1)=(mg2*(px2-px1))/(dist1^3)+(mg3*(px3-px1))/(dist2^3);
z(2,1)=vy1;
z(8,1)=(mg2*(py2-py1))/(dist1^3)+(mg3*(py3-py1))/(dist2^3);

z(3,1)=vx2;
z(9,1)=(mg1*(px1-px2))/(dist1^3)+(mg3*(px3-px2))/(dist3^3);
z(4,1)=vy2;
z(10,1)=(mg1*(py1-py2))/(dist1^3)+(mg3*(py3-py2))/(dist3^3);

z(5,1)=vx3;
z(11,1)=(mg1*(px1-px3))/(dist2^3)+(mg2*(px2-px3))/(dist3^3);
z(6,1)=vy3;
z(12,1)=(mg1*(py1-py3))/(dist2^3)+(mg2*(py2-py3))/(dist3^3);
end
```

The first few lines represent the initial conditions and the time interval to be solved over, which can be changed for different systems. The other lines which need to be changed are the first lines of the function `ydot`, where the masses and gravity are defined.

The `ydot` function represents the ODE system. It takes an array `x` of 12 elements defined as follows:

```
x = [x1 y1 x2 y2 x3 y3 v1x v1y v2x v2y v3x v3y]
```

Where the first six elements represent the positions of the three masses and the next six elements represent their component velocities. As defined by the coupled ODE above, it sets the first six elements of the result to be the last six elements of the input, and sets the last six elements of the outputs as the derivatives of v according to the formulas derived above.

The program uses `ode45` with this function, the interval and the initial conditions to create a matrix containing the positions and velocities of each mass over the time interval. In one figure, these are plotted once at each t value, and every 1/5th of a second a frame is added to the videowriter, creating an animated view of the masses moving. (a low framerate had to be used so that the animation didn't appear faster when the gaps between `t(i)` and `t(i+1)` were large). In figure 2, it plots the motion of the masses over the entire interval on one graph, showing the path each mass took from start to end. In figure 3, it plots the velocities of each mass over time

5 Numerical solution of the equations and simulation

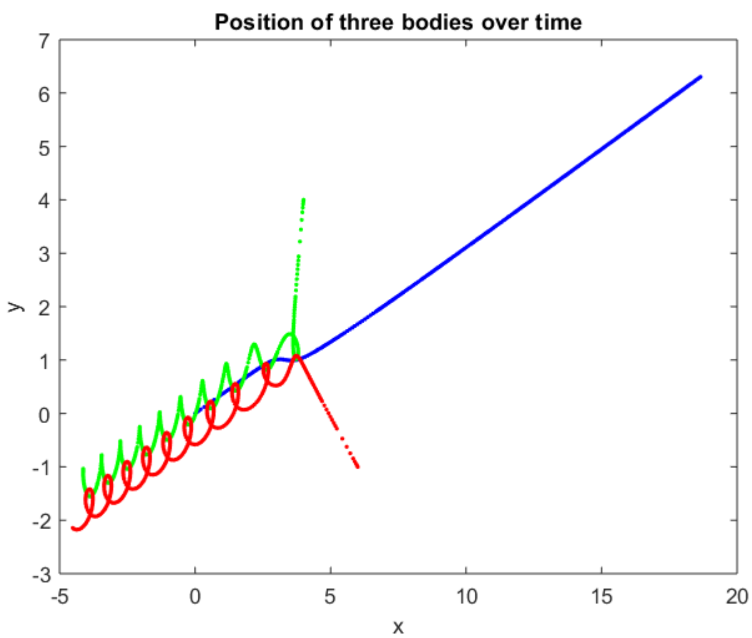
The video results for all cases are located online at:

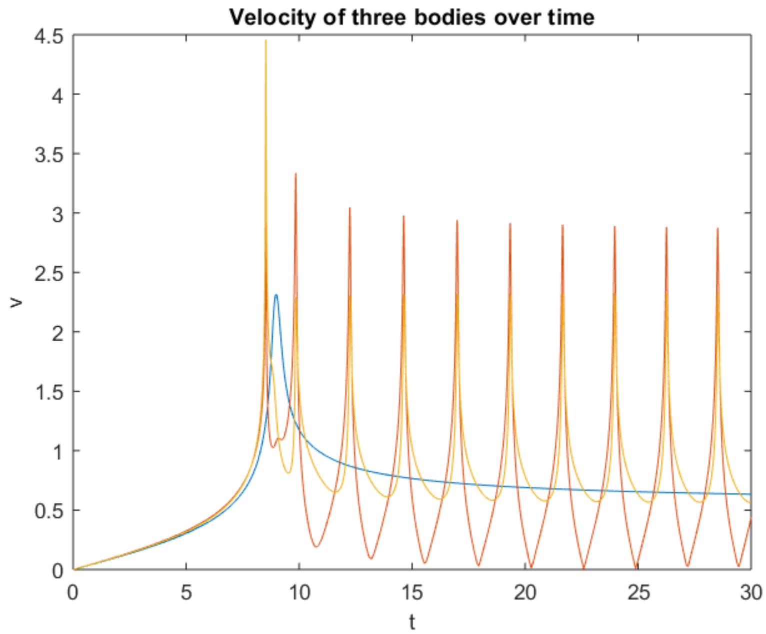
`https://1drv.ms/u/s!ArMTT58FUIq8tmosyrZYVC-VQvkW?e=iOVMLi` (The videos are small and may need to be downloaded to be viewed properly)

Several cases were tested. An initial test simulation was created, using masses set to 1 for each object and $g = 1$. The initial conditions passed were.

```
ic = [0 0 4 4 6 -1 0 0 0 0 0 0]
```

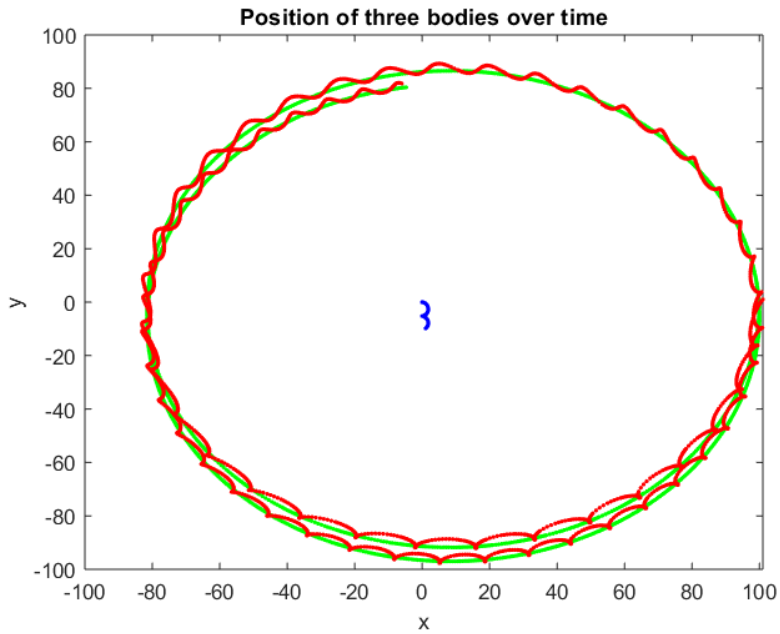
The first mass appeared to move away to the top right while the other two orbited each other down to the bottom left. The positions and velocity are shown below, and the animation is in the file threebod1.avi.



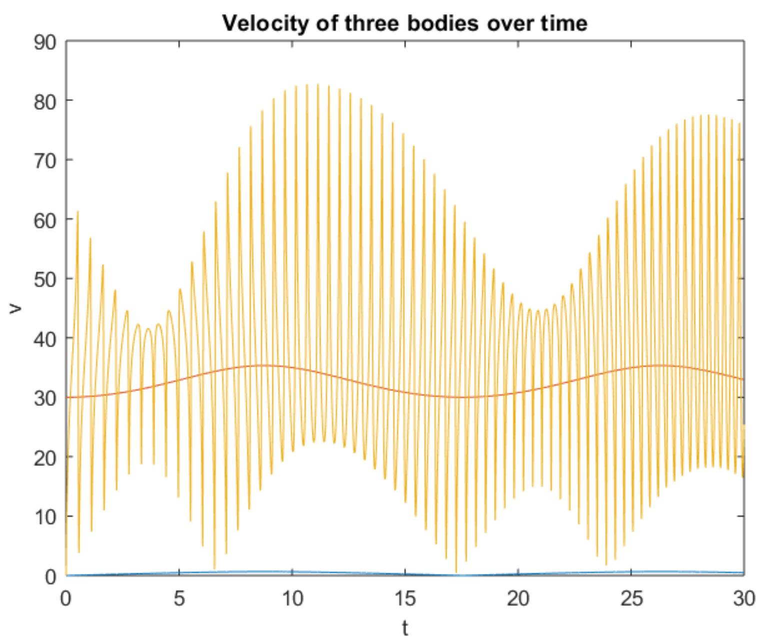


Here, the blue line represents the first mass, orange for the second and yellow for the third.

The second case tested was the real life example involving the earth, moon and a satellite. Initially, values used were from NASA's moon factsheet [1], however when running the simulation with these values the ode45 solver ran into difficulty with the step size becoming too small. Therefore, a smaller model was used using an Earth located at the origin of mass 10000, a moon of mass 100 starting at position (100,0) and starting velocity (0, -30) and a smaller satellite of mass 0 orbiting the moon at position (101,1) with no velocity. The gravity was 9.8 The animation is in the file threebod2.avi, and the motion is shown below.



This diagram shows a relatively stationary Earth (not influenced by the sun) with the moon (green) orbiting in an ellipse. The red satellite is orbiting the moon as it goes around the Earth.

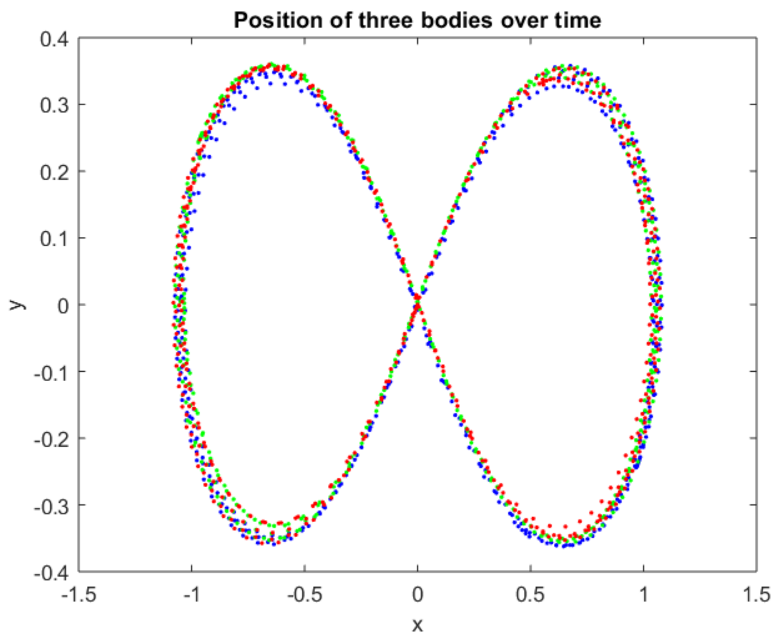


Here, the blue line is the velocity of the Earth, the orange is the moon, and the yellow is the satellite. The Earth stays mostly still and the moon orbits at a reasonably consistent speed. As the satellite is orbiting the moon which in turn is moving, its velocity is constantly fluctuating.

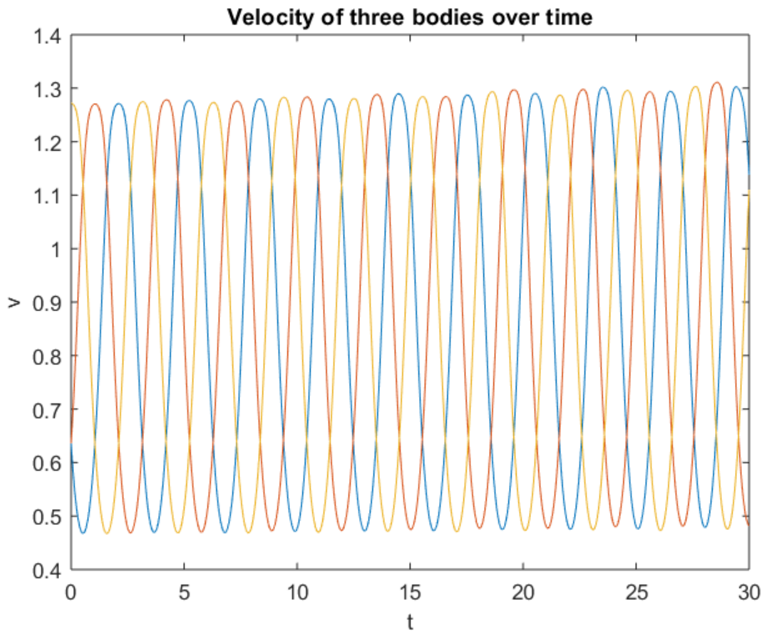
The final case tested is a special case described by Chenciner and Montgomery [2], in which all three bodies follow a periodic figure 8 path. The starting conditions as described in the paper are as follows:

```
ic1 = [0.97000436 -0.24308753 -0.97000436 0.24308753 0 0]
ic2 = [0.93240737/2 0.86473146/2 0.93240737/2 0.86473146/2 ...
    -0.93240737 -0.86473146]
```

The masses and gravity were all set to 1. The animation is in the file threebod3.avi, and the full motion is shown in the following graph



It can be seen that all three particles move in a fairly stable figure 8 pattern. The velocities fluctuate periodically between approx 0.5 and 1.3 for each particle. The particles move fastest as they pass through the centre, and are slower towards the outer edges of the path. This is also noticeable in the spacing of the points in the first graph.


Velocity of three bodies over time
v
t
1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0
5
10
15
20
25
30

6 Conclusion and self assessment

This project focused on a historically significant major problem in physics, and used numerical methods to simulate and solve this problem. The derivation of the equations showed understanding of the underlying mathematical concepts behind the problem. A practical example was put forward to link these concepts to real world scenarios.

A large amount of original code had to be written to simulate and solve the three body problem, which took a large amount of time and effort. The results found were interesting and showed three different cases of the three body problem, and suitable graphics were presented to convey this information.

While this project does not present much previously unseen information, it shows a high level of competency in using numerical methods to solve complex real world problems. For these reasons, I hope this project is worth at least 8-10 marks

References

[1] Grayzeck, E. NASA Official Moon Fact Sheet. 2017. `https://nssdc.gsfc.nasa.gov/planetary/factsheet/moonfact.html`

[2] Chenciner, A; Montgomery, R. A remarkable periodic solution of the three-body problem in the case of equal masses. Annals of Mathematics, 152 (2000), 881–901.

[3] Sauer, T. Numerical Analysis, 2nd Edition. Pearson, 2012.