2.1 The Restricted Three-Body Problem

QUESTION 1

$\dot{x}(1) + \dot{y}(2)$:

$$-\left(\dot{x}\frac{\partial\Omega}{\partial x} + \dot{y}\frac{\partial\Omega}{\partial y}\right) = \dot{x}\ddot{x} - 2\dot{x}\dot{y} + \dot{y}\ddot{y} + 2\dot{x}\dot{y} = \dot{x}\ddot{x} + \dot{y}\ddot{y}$$

Integrate both sides with respect to t:

$$-\int\left(\dot{x}\frac{\partial\Omega}{\partial x} + \dot{y}\frac{\partial\Omega}{\partial y}\right)\,dt = \int(\dot{x}\ddot{x} + \dot{y}\ddot{y})\,dt$$

$$-\Omega(x,y) = \frac{1}{2}\dot{x}^2 + \frac{1}{2}\dot{y}^2 + constant$$

Therefore, after rearranging, we get:

$$\frac{1}{2}\dot{x}^2 + \frac{1}{2}\dot{y}^2 + \Omega(x,y) = constant$$

If x_0, y_0, u_0, v_0 are the initial values of $x, y, \dot{x}, \dot{y}$, respectively, then we can say that:

$$\frac{1}{2}\dot{x}^2 + \frac{1}{2}\dot{y}^2 + \Omega(x,y) = \frac{1}{2}{u_0}^2 + \frac{1}{2}{v_0}^2 + \Omega(x_0,y_0)$$

$\frac{1}{2}\dot{x}^2 + \frac{1}{2}\dot{y}^2 \geq 0$, so

$$\Omega(x,y) \leq \frac{1}{2}\dot{x}^2 + \frac{1}{2}\dot{y}^2 + \Omega(x,y) = \frac{1}{2}{u_0}^2 + \frac{1}{2}{v_0}^2 + \Omega(x_0,y_0)$$

Hence, we get the inequality desired.

Programming Task:

We can turn (1) and (2) into a system of four first-order ODEs by introducing variables u and v:

$$\frac{dx}{dt} = u$$

$$\frac{dy}{dt} = v$$

$$\frac{du}{dt} = 2v - \frac{\partial\Omega}{\partial x}$$

$$\frac{dv}{dt} = -2u - \frac{\partial\Omega}{\partial y}$$

Find what $\frac{\partial\Omega}{\partial x}$ and $\frac{\partial\Omega}{\partial y}$ are by differentiating with chain rule:

$$\frac{\partial\Omega}{\partial x} = -\mu(x+1-\mu) - (1-\mu)(x-\mu) + \frac{\mu(x+1-\mu)}{{r_1}^3} + \frac{(1-\mu)(x-\mu)}{{r_2}^3}$$

$$\frac{\partial\Omega}{\partial y} = -\mu y - (1-\mu)y + \frac{\mu y}{{r_1}^3} + \frac{(1-\mu)y}{{r_2}^3}$$

The program used to solve (1) and (2) numerically, given suitable initial conditions, is **Code 1** on page 20, labelled as

```
first_order_ODEs(t,y_val,m)
```

, which defines and stores the four first order ODEs, and

```
solving_ODEs(m)
```

, which finds and plots the solution curve.

In the code, I use m to represent μ and w to represent Ω. This code is first used in **Question 3.**

In this project, I test the accuracy of my programs by plotting the allowed region on the same set of axes and showing that the condition Ω(x,y) ≤ J is always satisfied. Furthermore, I've written the codes such that it inclusively calculated the value of J for 20 points, and it outputted the same value each time, confirming the accuracy of the programs.

QUESTION 2

If $\Omega = -\frac{1}{2r_2}$, then

$$\frac{\partial\Omega}{\partial x} = \frac{x-\mu}{2{r_2}^3}$$

$$\frac{\partial\Omega}{\partial x} = \frac{y}{2{r_2}^3}$$

So (1) and (2) become:

$$\ddot{x} - 2\dot{y} = -\frac{x-\mu}{2{r_2}^3} \qquad \ddot{y} + 2\dot{x} = -\frac{y}{2{r_2}^3}$$

Use parametric equations:

$$x = \mathrm{a}\cos(\theta(t)) + 0.5 \qquad y = \mathrm{a}\sin(\theta(t))$$

Differentiate these with respect to t:

$$\dot{x} = -\mathrm{a}\sin(\theta)\frac{d\theta}{dt}$$

$$\dot{y} = \mathrm{a}\cos(\theta)\frac{d\theta}{dt}$$

$$\ddot{x} = -\mathrm{a}\cos(\theta)\left(\frac{d\theta}{dt}\right)^2 - \mathrm{a}\sin(\theta)\frac{d^2\theta}{dt^2}$$

$$\ddot{y} = -\mathrm{a}\sin(\theta)\left(\frac{d\theta}{dt}\right)^2 + \mathrm{a}\cos(\theta)\frac{d^2\theta}{dt^2}$$

Substitute these into the new differential equations:

$$-\mathrm{a}\cos(\theta)\left(\frac{d\theta}{dt}\right)^2 - \mathrm{a}\sin(\theta)\frac{d^2\theta}{dt^2} - 2\,\mathrm{a}\cos(\theta)\frac{d\theta}{dt} = -\frac{\mathrm{a}\cos(\theta)}{2a^3}$$

$$-\operatorname{asin}(\theta)\left(\frac{d\theta}{dt}\right)^2 + \operatorname{acos}(\theta)\frac{d^2\theta}{dt^2} - 2\operatorname{asin}(\theta)\frac{d\theta}{dt} = -\frac{\operatorname{asin}(\theta)}{2a^3}$$

If we divide the first equation by $-\operatorname{acos}(\theta)$ and the second equation by $-\operatorname{asin}(\theta)$, we get:

$$\left(\frac{d\theta}{dt}\right)^2 + \tan(\theta)\frac{d^2\theta}{dt^2} + 2\frac{d\theta}{dt} = \frac{1}{2a^3}$$

$$\left(\frac{d\theta}{dt}\right)^2 - \frac{1}{\tan(\theta)}\frac{d^2\theta}{dt^2} + 2\frac{d\theta}{dt} = \frac{1}{2a^3}$$

From this, we can get that $\tan(\theta)\frac{d^2\theta}{dt^2} + \frac{1}{\tan(\theta)}\frac{d^2\theta}{dt^2} = 0$, so:

$$\frac{1}{\tan(\theta)}\frac{d^2\theta}{dt^2}\left(1 + \tan^2(\theta)\right) = 0$$

Therefore, $\frac{d^2\theta}{dt^2} = 0$, which makes sense as we are working in a frame where angular velocity $\left(\frac{d\theta}{dt}\right)$ is equal to 1, so θ would be linear in t. Now substitute this to see if a solution exists for θ:

$$\left(\frac{d\theta}{dt}\right)^2 + 2\frac{d\theta}{dt} = \frac{1}{2a^3}$$

Let $z = \frac{d\theta}{dt}$:

$$z^2 + 2z = \frac{1}{2a^3}$$

$$z^2 + 2z - \frac{1}{2a^3} = 0$$

$$2a^3z^2 + 4a^3z - 1 = 0$$

$$\frac{d\theta}{dt} = z = \frac{-4a^3 \pm \sqrt{16a^6 + 8a^3}}{4a^3}$$

, which is real, so solutions for θ do exist.

Hence (1) and (2) then have analytic solutions with the spacecraft in a circular orbit of radius a about P2, where a can take any small value.

The program used to solve (1) and (2) with Ω specified by (5) instead of (3) is **Code 3** on page 21, labelled as

```
first_order_ODEs_2(t,y_val,m)
```

, which defines and stores the four first order ODEs, and

```
solving_ODEs_2(m)
```

, which finds and plots the solution curve.

I chose to check if the program matched the analytic solutions for a = 0.5, using initial conditions that would align with point (1, 0) on the circle of radius 0.5 centred at (0.5, 0): (1, 0, 0, -1), and the trajectory plotted was indeed a circle of radius 0.5 centred at (0.5, 0). Hence the modified program can reproduce the analytic solutions.

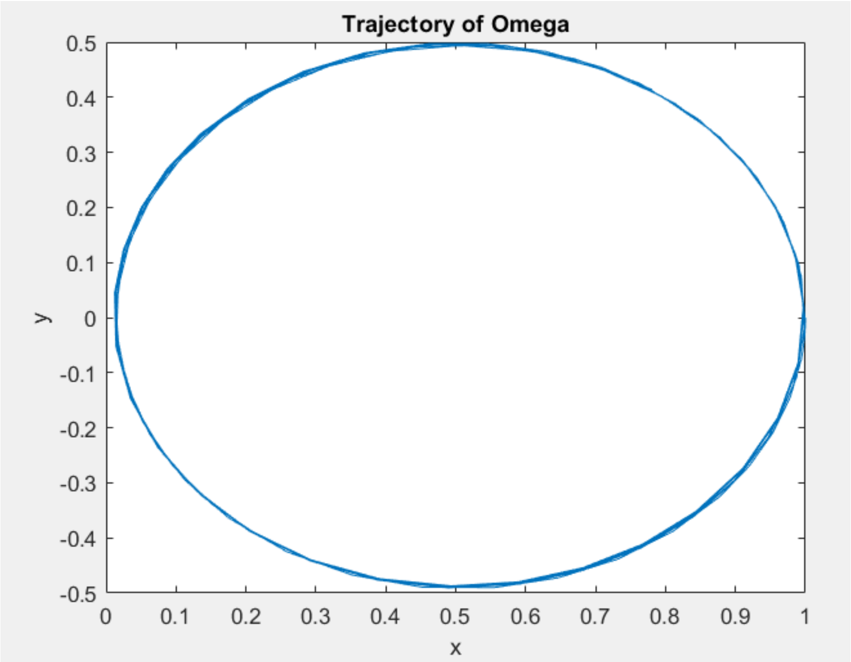

Figure 1: *A graph displaying the trajectory of the modified program specified by (1), (2) and (5), for value of radius a = 0.5.*

QUESTION 3

Using **Code 1,** we can obtain the following plots:

v_0 = -0.50:

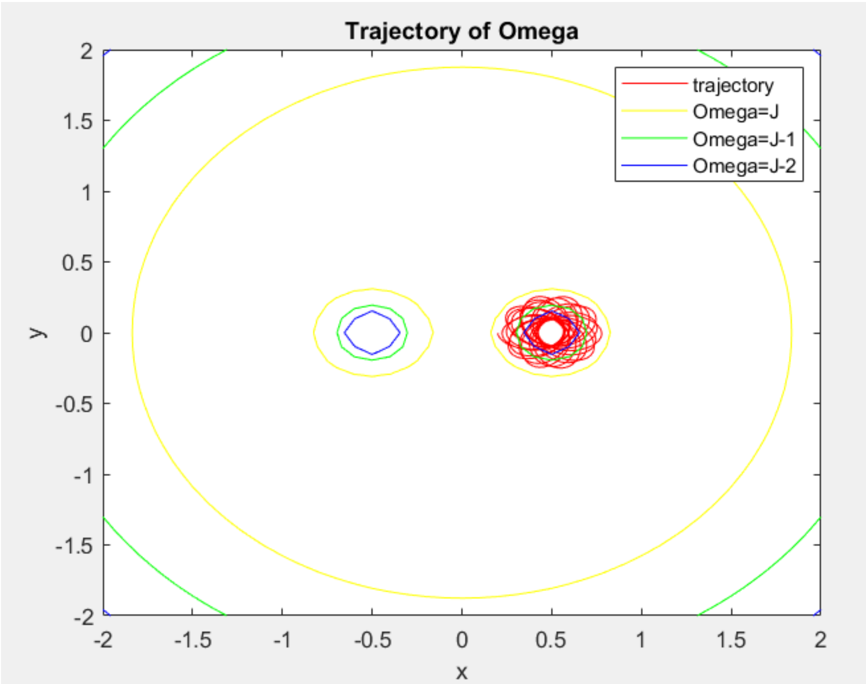

Figure 2: *A graph displaying the trajectory of the program specified by (1), (2) and (3) for v_0 = -0.5, along with the allowed region $\Omega(x, y) \leq J$.*

v_0 = -1.00:

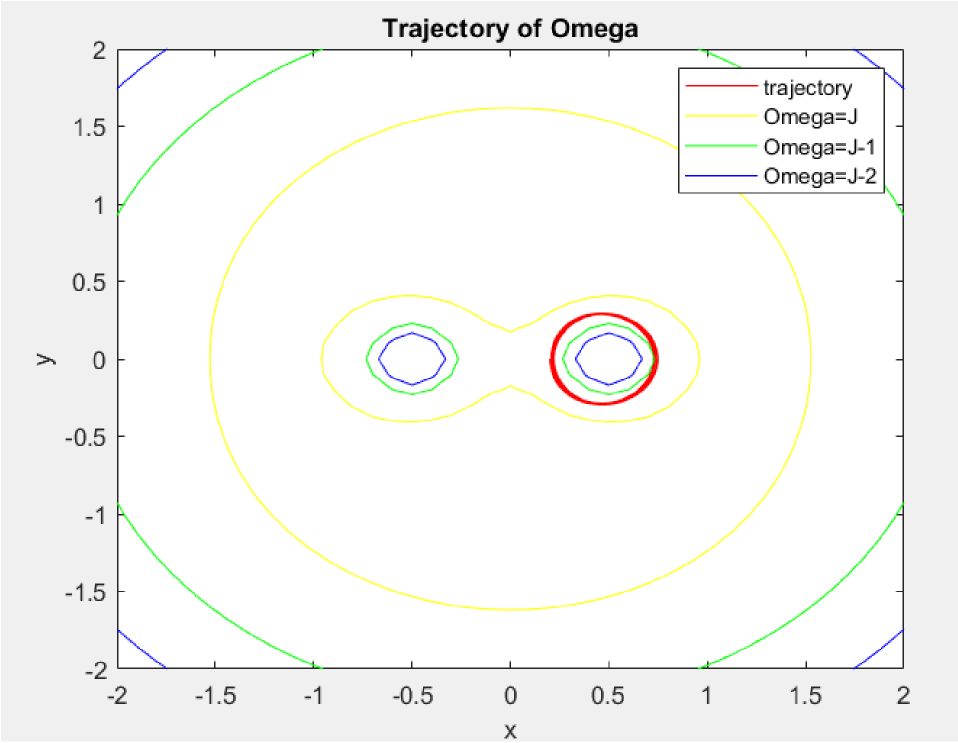

Figure 3: *A graph displaying the trajectory of the program specified by (1), (2) and (3) for v_0 = -1, along with the allowed region $\Omega(x, y) \leq J$.*

v_0 = -1.04:

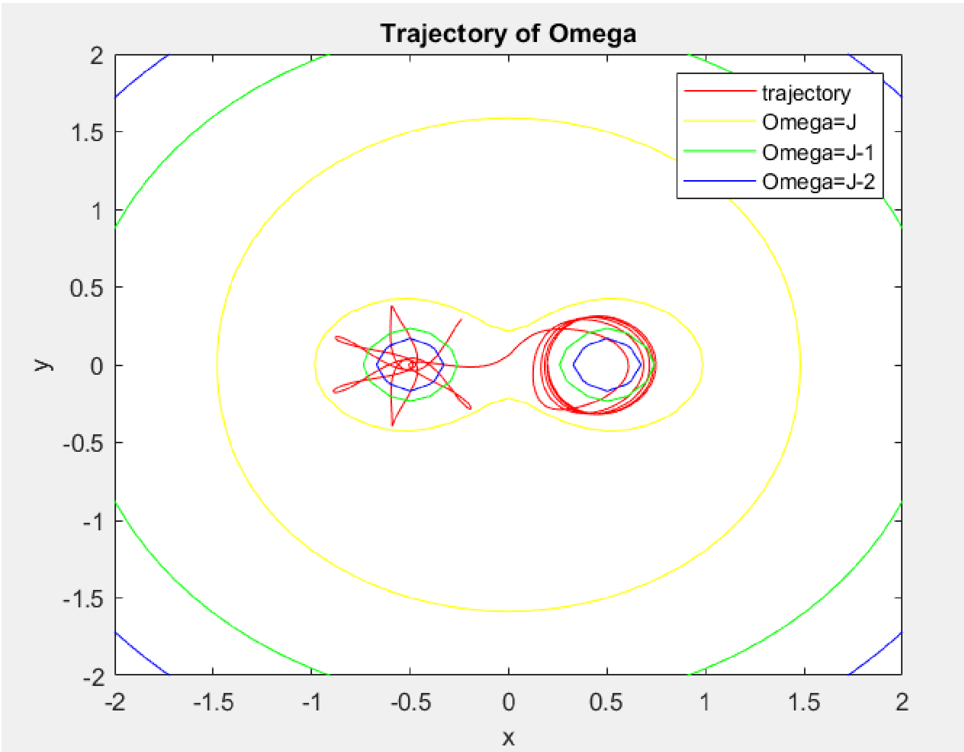

Figure 4: *A graph displaying the trajectory of the program specified by (1), (2) and (3) for v_0 = -1.04, along with the allowed region $\Omega(x, y) \leq J$.*

v_0 = -1.18:

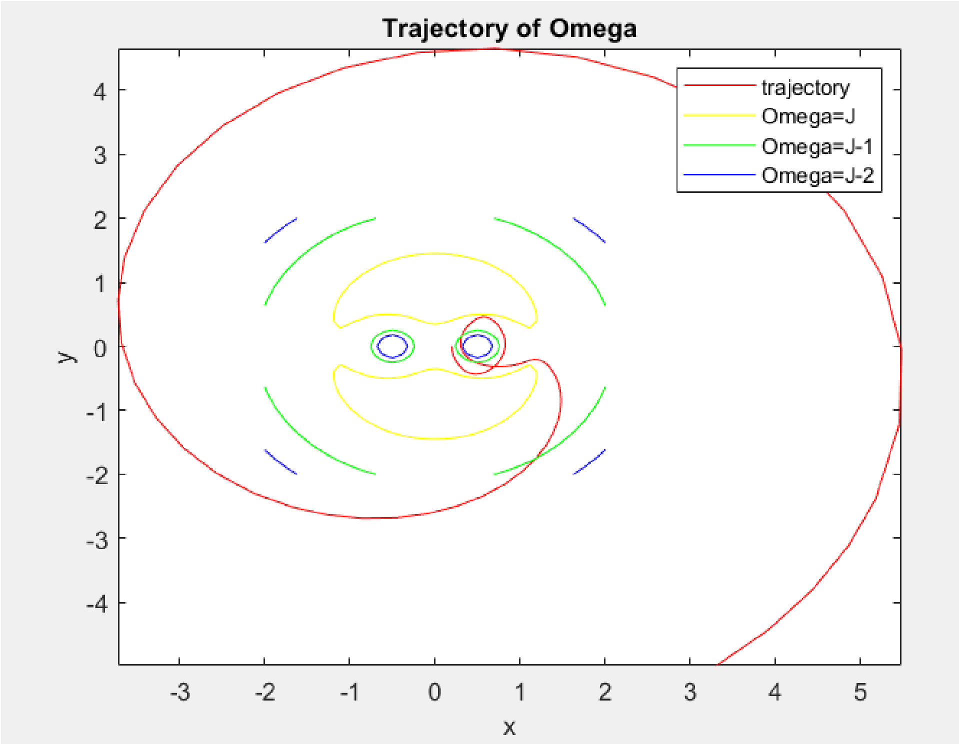

Figure 5: *A graph displaying the trajectory of the program specified by (1), (2) and (3) for v_0 = -1.18, along with the allowed region $\Omega(x, y) \leq J$.*

v_0 = -1.25:

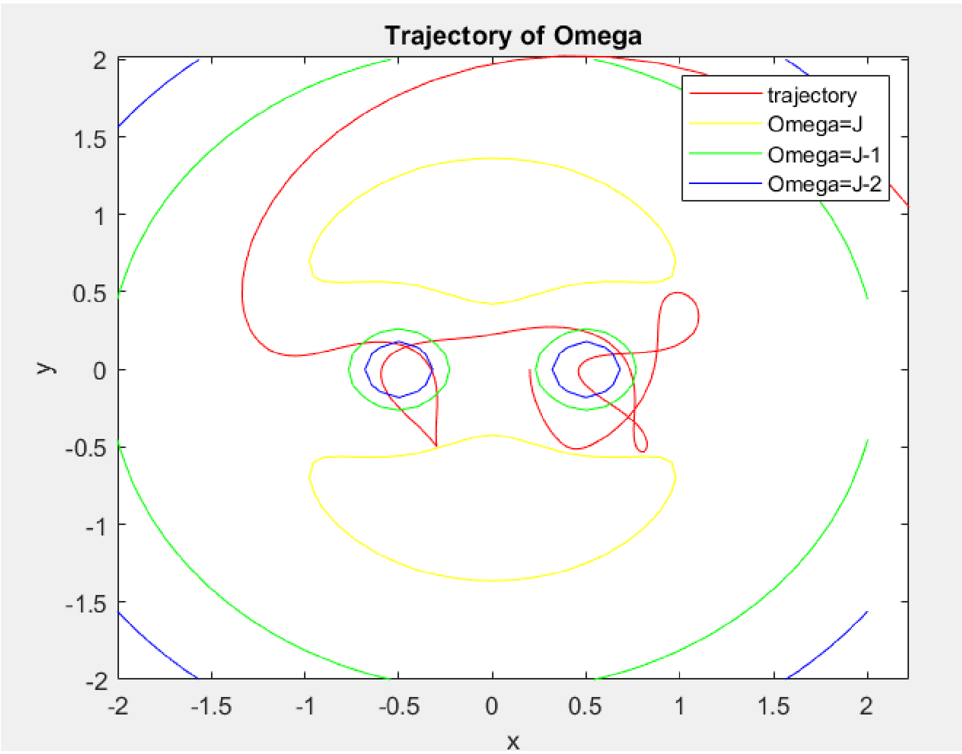

Figure 6: *A graph displaying the trajectory of the program specified by (1), (2) and (3) for v_0 = -1.25, along with the allowed region $\Omega(x, y) \leq J$.*

v_0 = -1.50:

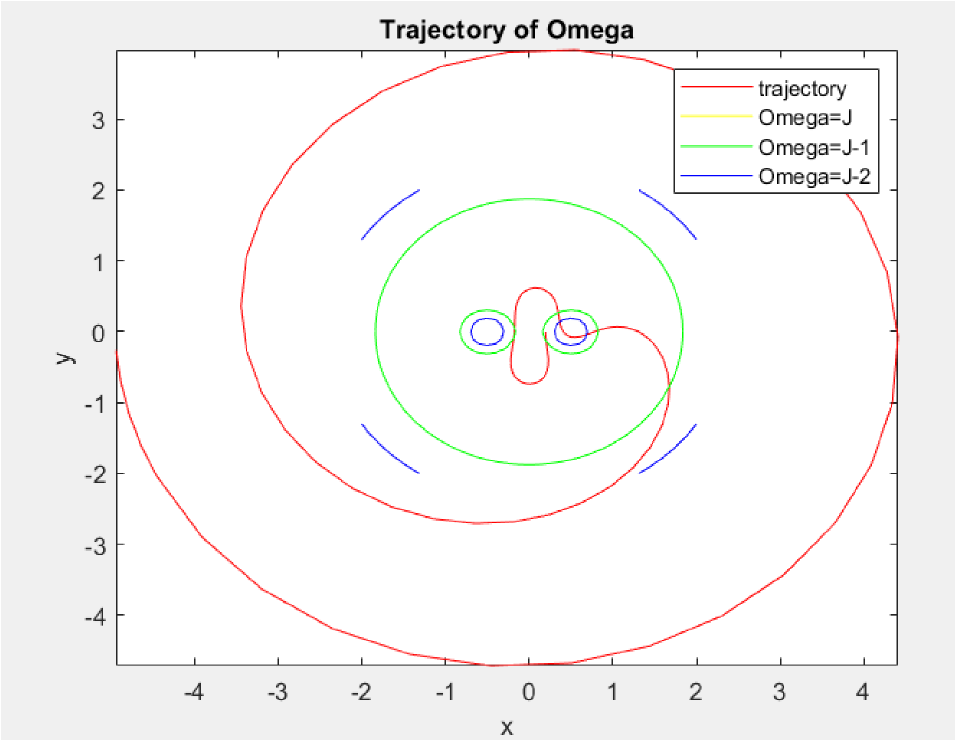

Figure 7: *A graph displaying the trajectory of the program specified by (1), (2) and (3) for v_0 = -1.5, along with the allowed region $\Omega(x, y) \leq J$.*

For each of these plots, I have not only plotted Ω(x, y) = J in yellow, but also Ω(x, y) = J – 1, J – 2 to give an indication of which side of Ω(x, y) = J is the allowed region.

The allowed regions seem to be increasing in size as the value of v_0 increases in magnitude, and the trajectories look mostly stable and orbit-like around P_2 for values of v_0 smaller in magnitude: -0.5, -1. For -1.04, it looks mostly orbit-like until it seems to spiral away from this circular orbit around P_2 to near P_1. For v_0 = -1.18, -1.25 and -1.5, the trajectory spirals away from both planets, starting at P_2 and spiralling away (except in the case v_0 = -1.25, the trajectory firstly visits P_1 and then spirals away. So generally, as v_0 increases in magnitude, the trajectories become more unstable and spiral away from the planets.

Both the trajectories and the allowed regions generally increase in size for v_0 of a higher magnitude, so it can be said that the allowed region is a useful guide to the size of the trajectory.

Close to the boundary of the allowed region, the curve of the trajectory has the same gradient as the boundary at those points, from which it then curves away from the boundary of the allowed region.

The only trajectory that travels from the neighbourhood of P_2 to the neighbourhood of P_1 and remains in that region is the trajectory with value v_0 = -1.04 (v_0 = -1.25 visits P_1, but promptly spirals away), so this would be the most suitable value of v_0 to do so.

QUESTION 4

The program used to plot the contours for various μ is **Code 2** on page 20, labelled as

`contours_2(m)`

μ = 0.25:

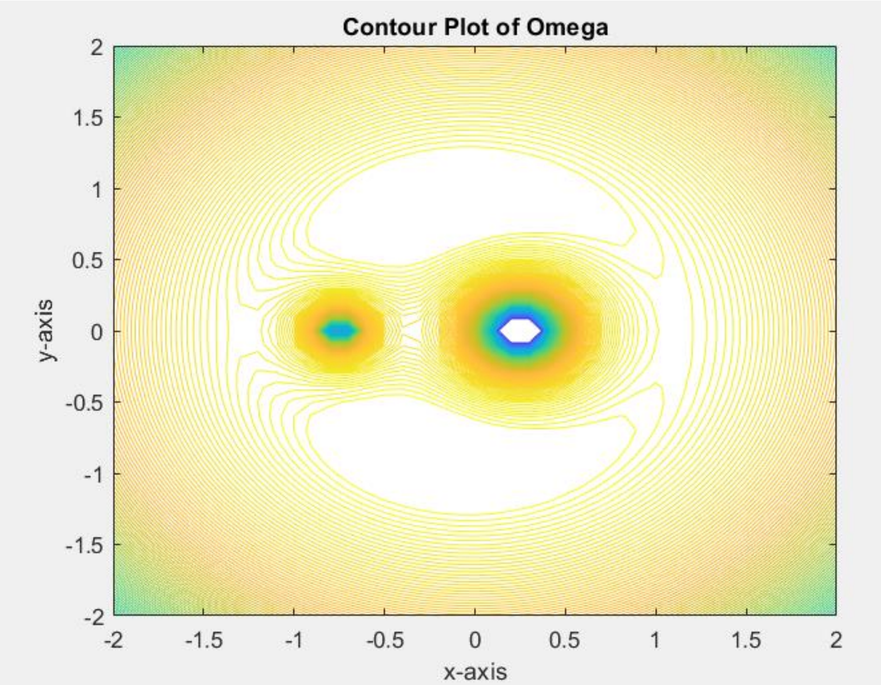

Figure 8: *A graph displaying the contours of Ω(x, y), which is specified by (3) in the project booklet, for μ=0.25.*

μ = 0.50:

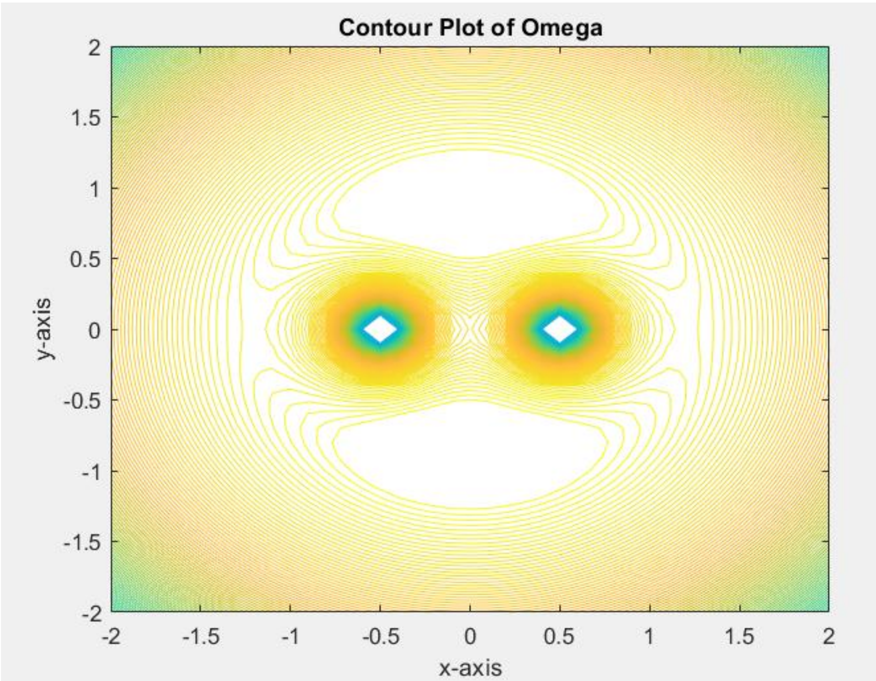

Figure 9: *A graph displaying the contours of Ω(x, y), which is specified by (3) in the project booklet, for μ=0.50.*

μ = 0.75:

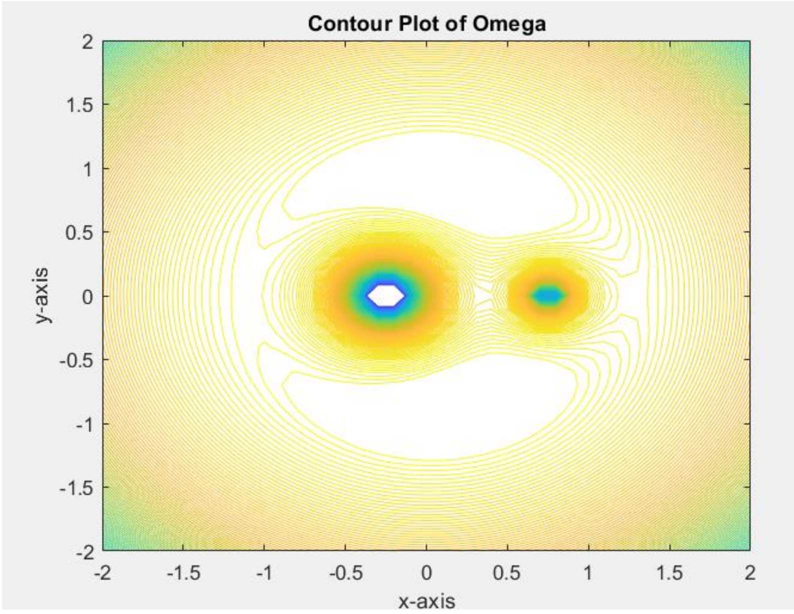

Figure 10: *A graph displaying the contours of Ω(x, y), which is specified by (3) in the project booklet, for μ=0.75.*

As the contour plots show, there are three equilibrium points on the x-axis – one in between P_1 and P_2, one to the left of P_1 and one to the right of P_2. Furthermore, we can also see ellipses formed around points above and below P_1 and P_2, at the third vertex of an equilateral triangle whose other two vertices are at P_1 and P_2. So altogether, this system generally has five equilibrium points.

Letting

$$\frac{dx}{dt} = u\big(x(t)\big) \qquad \frac{dy}{dt} = v(x(t))$$

, the given system of second-order ODEs reduces to:

$$u\frac{du}{dx} - 2v = -\frac{\partial\Omega}{\partial x} \qquad v\frac{dv}{dy} + 2u = -\frac{\partial\Omega}{\partial y}$$

So, we can see that in order to find the equilibrium points (at which u = v = 0), we must solve:

$$\frac{\partial\Omega}{\partial x} = \frac{\partial\Omega}{\partial y} = 0$$

The program used to integrate (1) – (3) numerically from t = 0 to t = 15, starting from a small perturbation away from the equilibrium points (hence determining the stability of said equilibrium point) and plot the trajectory is **Code 4** on page 22, labelled as

`stability(x0,y0,u0,v0,m)`

This code inclusively incorporates `first_order_ODEs,` just like `solving_ODEs(m,v_val)` did.

This time, I have chosen to make all of the initial values input parameters too.

Let's investigate the stability of the equilibrium points on the x-axis, starting with μ = 0.5. The coordinates of the equilibrium point in between P_1 and P_2 is clearly (0, 0). Start with initial conditions a small perturbation away from (0, 0) – let's use (0.01, 0.01) in this case. Set u0 = 0 and v0 = 0, and we obtain the following trajectory:

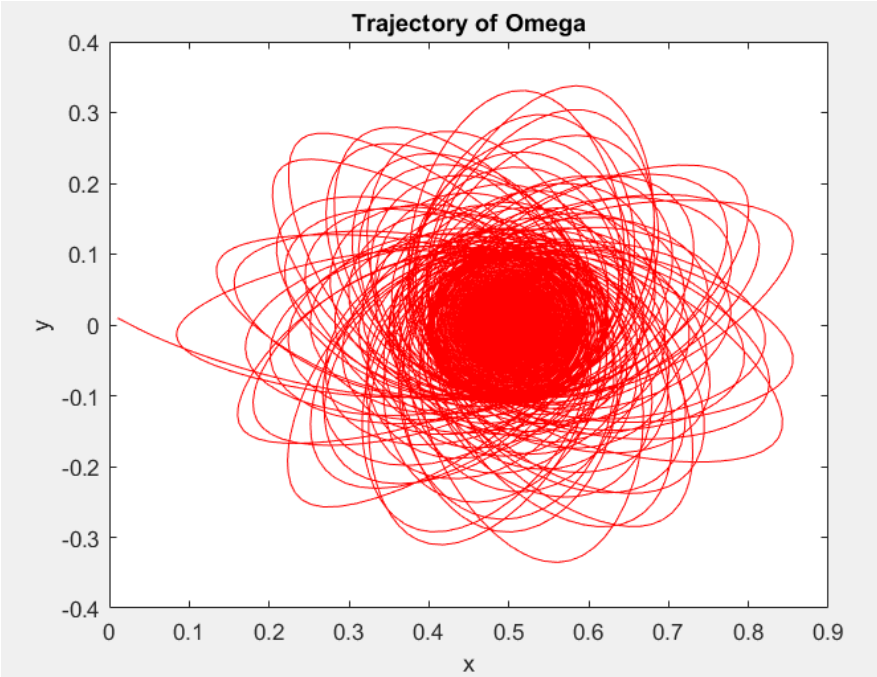

Figure 11: *A graph displaying the trajectory specified by (1)– (3), starting a small perturbation away from equilibrium point (0, 0) of Ω(x, y), for μ = 0.5.*

We can see from this that as t increases, the trajectory goes away from (0, 0) towards P_2, which suggests that it is an UNSTABLE equilibrium point.

Now investigate the stability of the two end points, starting with the one on the right. Start with initial conditions a small perturbation away from it by observing the contours – let's use (1.21, 0.01) in this case. Set u0 = 0 and v0 = 0 again, and we obtain the following trajectory:

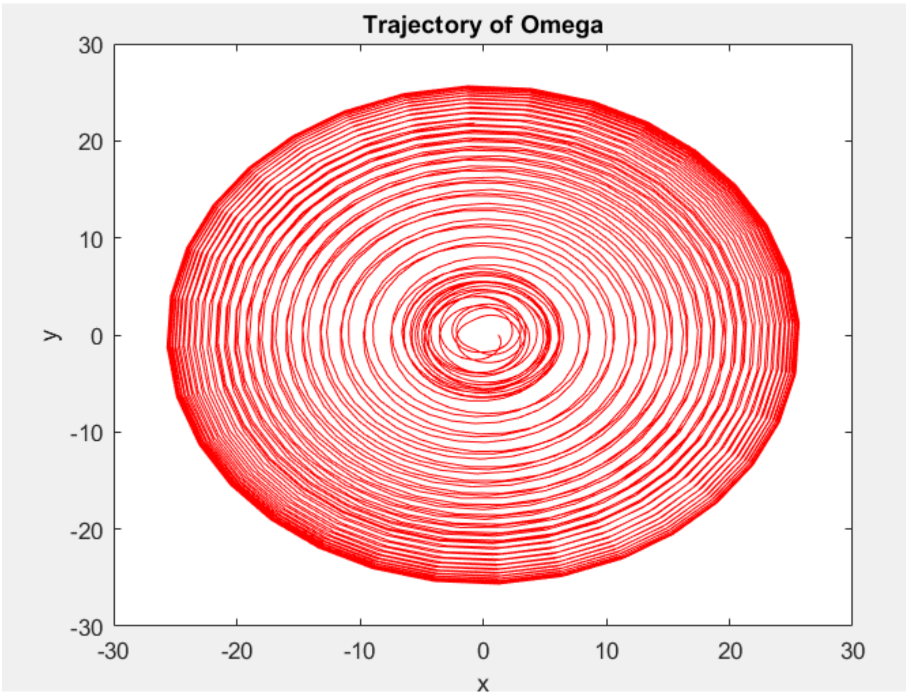

Figure 12: *A graph displaying the trajectory specified by (1)– (3), starting a small perturbation away from equilibrium point close to (1.21, 0) of Ω(x, y), for μ = 0.5.*

We can see from this that as t increases, the trajectory spirals away from the equilibrium point, which suggests that it is also an UNSTABLE equilibrium point.

Then look at the left equilibrium point on the x-axis. By observing the contours again, use initial conditions (-1.21, -0.01, 0, 0) and we obtain the following trajectory:

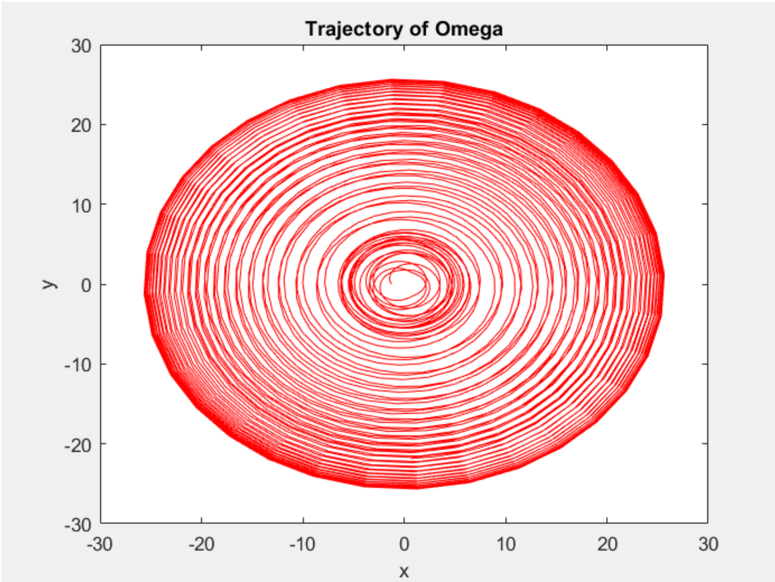

Figure 13: *A graph displaying the trajectory specified by (1)– (3), starting a small perturbation away from equilibrium point close to (-1.21, 0) of Ω(x, y), for μ = 0.5.*

The two equilibrium points not on the x-axis, at the third vertex of an equilateral triangle whose other two vertices are at P_1 and P_2, are $(0, \pm\sqrt{3}/2)$.

These are investigated in **Question 5**, so we will leave this for now.

Next, we will switch our attention to the equilibrium points on the x-axis for μ = 0.25, firstly focusing on the point in between P_1 and P_2. Use initial conditions (-0.34, 0.01, 0, 0):

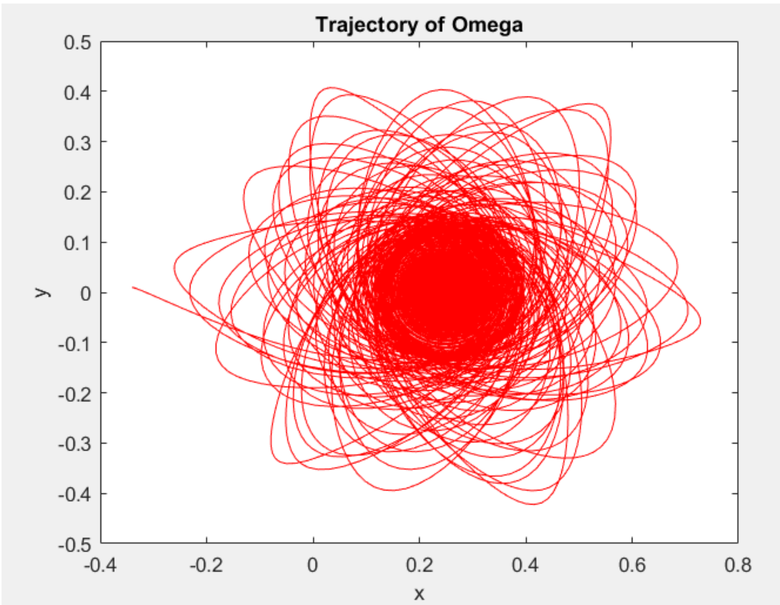

Figure 14: *A graph displaying the trajectory specified by (1)– (3), starting a small perturbation away from equilibrium point close to (-0.34, 0) of Ω(x, y), for μ = 0.25.*

We can do the same for the right and left equilibrium points on the x-axis, using initial conditions (1.11, 0.01, 0, 0) and (-1.31, 0.01, 0, 0) respectively and obtain spirals similar to those for $\mu = 0.5$.

Now focus on the equilateral points here, which are (-0.25, $\pm\sqrt{3}/2$). Limit focus to just the positive y coordinate, and use initial conditions with a smaller perturbation $(-0.2501, (\sqrt{3}/2) + 0.0001,0,0)$:

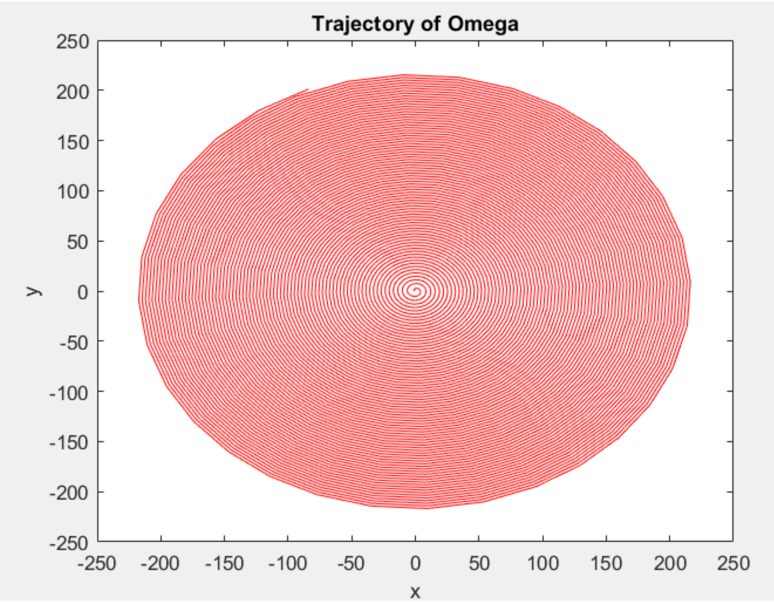

Figure 15: *A graph displaying the trajectory specified by (1)– (3), starting a very small perturbation away from equilibrium point* (-0.25, $\sqrt{3}/2$) *of Ω(x, y), for μ = 0.25.*

We can see from this that as t increases, the trajectory spirals away from the equilibrium point, which suggests that it is also an UNSTABLE equilibrium point.

We can do the same for the negative y-coordinate and get that it's unstable too.

Next, switch our attention to the equilibrium points on the x-axis for $\mu = 0.75$, firstly focusing on the point in between P_1 and P_2. Use initial conditions (0.34, 0.01, 0, 0):

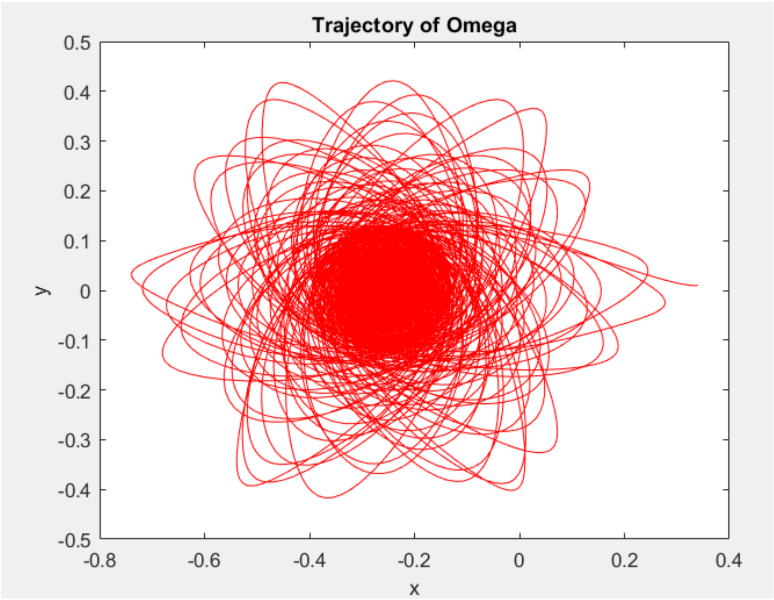

Figure 16: *A graph displaying the trajectory specified by (1)– (3), starting a small perturbation away from equilibrium point close to (0.34, 0) of Ω(x, y), for μ = 0.75.*

We can do the same for the right and left equilibrium points on the x-axis, using initial conditions (1.31, 0.01, 0, 0) and (-1.11, 0.01, 0, 0) respectively and obtain spirals similar to those for $\mu = 0.5$ and $\mu = 0.25$.

Finally, focus on the equilateral points here, which are $(0.25, \pm\sqrt{3}/2)$. Limit focus to just the negative y coordinate, and use initial conditions with a smaller perturbation $(0.2501, -(\sqrt{3}/2) - 0.0001,0,0)$:

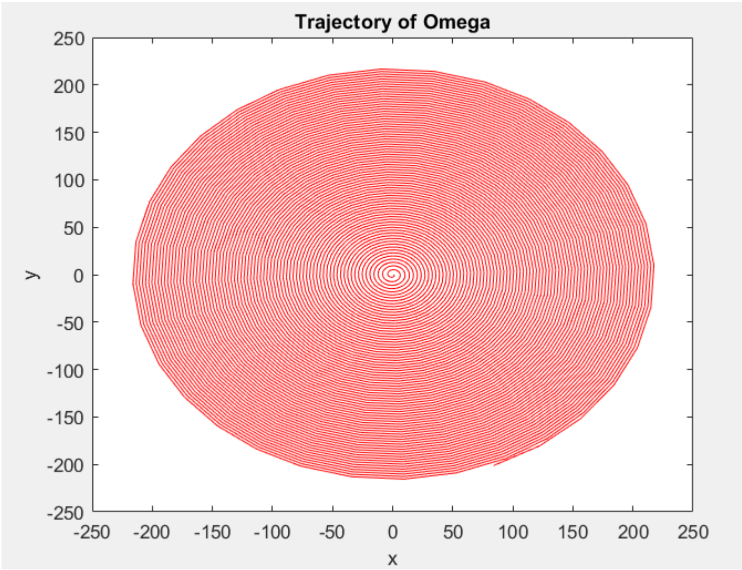

Figure 17: *A graph displaying the trajectory specified by (1)– (3), starting a very small perturbation away from equilibrium point* $(0.25, -\sqrt{3}/2)$ *of Ω(x, y), for μ = 0.75.*

This trajectory spirals away from the equilibrium point, which suggests that it is also an UNSTABLE equilibrium point. We can do the same for the positive y-coordinate and get that it's unstable too.

Generally, we can observe that the three equilibrium points on the x-axis are all unstable, and this doesn't seem to change with the value of μ.

To use linearised stability analysis, recall the system of first order differential equations:

$$\frac{dx}{dt} = u$$

$$\frac{dy}{dt} = v$$

$$\frac{du}{dt} = 2v - \frac{\partial \Omega}{\partial x}$$

$$\frac{dv}{dt} = -2u - \frac{\partial \Omega}{\partial y}$$

Recall the definition of the Jacobian in a linearisation problem:

$$\begin{pmatrix} f_x & f_y & f_u & f_v \\ g_x & g_y & g_u & g_v \\ h_x & h_y & h_u & h_v \\ k_x & k_y & k_u & k_v \end{pmatrix}$$

, where f, g, h, and k are functions equal to the derivatives of x, y, u and v respectively.

Substitute the actual expressions into the matrix, ignoring $\partial^2\Omega/\partial x\partial y$ derivatives as they cancel each other out anyways after expansion:

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\partial^2\Omega/\partial x^2 & 0 & 0 & 2 \\ 0 & -\partial^2\Omega/\partial y^2 & -2 & 0 \end{pmatrix}$$

We need to calculate the eigenvalues of this matrix – if they all have negative real parts, it's a stable point and if any of the eigenvalues have positive real parts, it's unstable.

Use formula det(I −λJ)=0:

$$\begin{vmatrix} -\lambda & 0 & 1 & 0 \\ 0 & -\lambda & 0 & 1 \\ -\partial^2\Omega/\partial x^2 & 0 & -\lambda & 2 \\ 0 & -\partial^2\Omega/\partial y^2 & -2 & -\lambda \end{vmatrix}$$

Expanding the determinant, we get:

$$-\lambda\begin{vmatrix} -\lambda & 0 & 1 \\ 0 & -\lambda & 2 \\ -\partial^2\Omega/\partial y^2 & -2 & -\lambda \end{vmatrix} + \begin{vmatrix} 0 & -\lambda & 1 \\ -\partial^2\Omega/\partial x^2 & 0 & 2 \\ 0 & -\partial^2\Omega/\partial y^2 & -\lambda \end{vmatrix} = 0$$

Expanding it further:

$$-\lambda\left(-\lambda\begin{vmatrix} -\lambda & 2 \\ -2 & -\lambda \end{vmatrix} + \begin{vmatrix} 0 & -\lambda \\ -\partial^2\Omega/\partial y^2 & -2 \end{vmatrix}\right) + \left(\lambda\begin{vmatrix} -\partial^2\Omega/\partial x^2 & 2 \\ 0 & -\lambda \end{vmatrix} + \begin{vmatrix} -\partial^2\Omega/\partial x^2 & 0 \\ 0 & -\partial^2\Omega/\partial y^2 \end{vmatrix}\right) = 0$$

Now, we get:

$$-\lambda\left[-\lambda(\lambda^2+4) - \lambda\,\partial^2\Omega/\partial y^2\right] + \left[\lambda\left(\lambda\,\partial^2\Omega/\partial x^2\right) + \partial^2\Omega/\partial x^2\,\partial^2\Omega/\partial y^2\right] = 0$$

$$\lambda^2(\lambda^2+4) + \lambda^2\,\partial^2\Omega/\partial y^2 + \lambda^2\,\partial^2\Omega/\partial x^2 + \partial^2\Omega/\partial x^2\,\partial^2\Omega/\partial y^2 = 0$$

$$\lambda^4 + \lambda^2\left(4 + \partial^2\Omega/\partial y^2 + \partial^2\Omega/\partial x^2\right) + \partial^2\Omega/\partial x^2\,\partial^2\Omega/\partial y^2 = 0$$

$$\lambda^2 = \frac{-\left(4 + \frac{\partial^2\Omega}{\partial y^2} + \frac{\partial^2\Omega}{\partial x^2}\right) \pm \sqrt{\left(4 + \frac{\partial^2\Omega}{\partial y^2} + \frac{\partial^2\Omega}{\partial x^2}\right)^2 - 4\frac{\partial^2\Omega}{\partial x^2}\frac{\partial^2\Omega}{\partial y^2}}}{2}$$

We can't immediately determine whether these are complex or real just from this, but regardless, when we take the square root of this to retrieve λ, we will get eigenvalues with both negative and/or non-negative real parts, hence it must be unstable.

QUESTION 5

Focusing on μ = 0.008 initially, still using **Code 4** on page 22**,** the equilateral Lagrange points are $(-0.492, \pm\sqrt{3}/2)$. Choose to focus on the positive y-coordinate first, and use initial conditions $(-0.4921, (\sqrt{3}/2) + 0.0001, 0, 0)$:

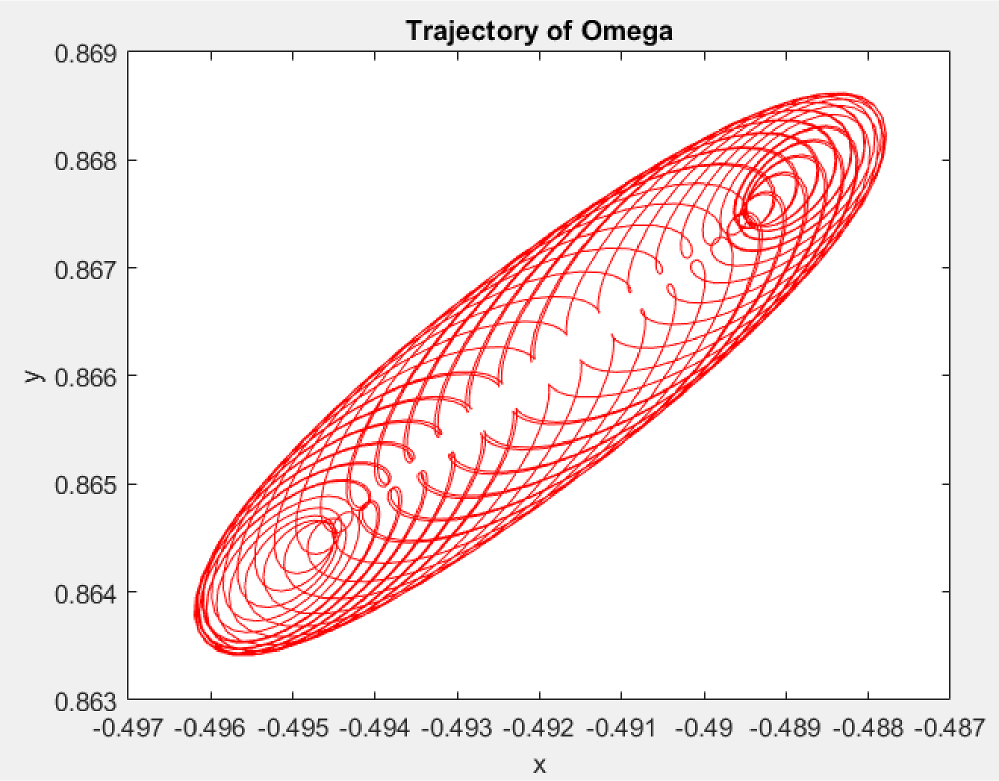

Figure 18: *A graph displaying the trajectory specified by (1)– (3), starting a very small perturbation away from equilibrium point* (-0.492, $\sqrt{3}/2$) *of Ω(x, y), for μ = 0.008.*

This trajectory keeps circling around the equilibrium point, suggesting that it is a STABLE equilibrium point. We can do the same for the negative y-coordinate and show that it is also stable.

Next, look at μ = 0.022. The equilateral Lagrange points are $(-0.478, \pm\sqrt{3}/2)$. Once again, focus on the positive y-coordinate first, and use initial conditions $(-0.4781, (\sqrt{3}/2) + 0.0001, 0, 0)$:

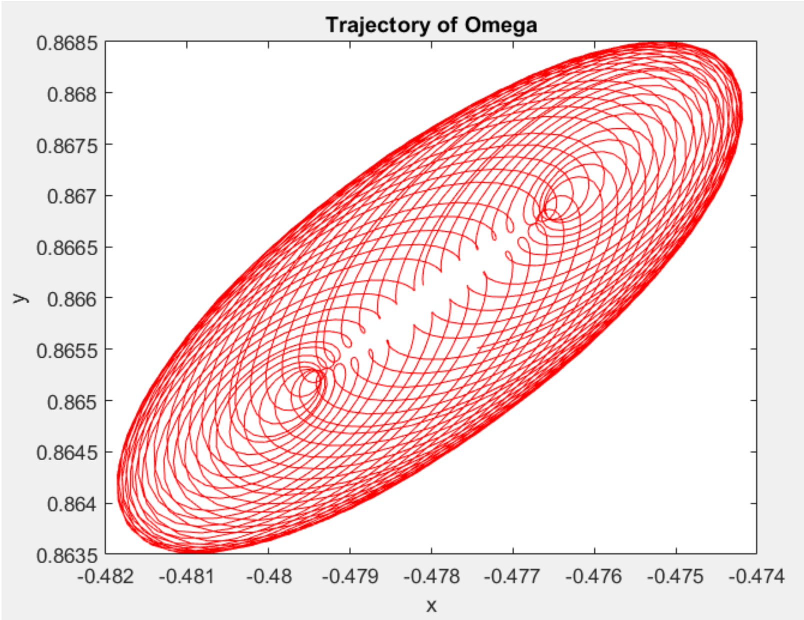

Figure 19: *A graph displaying the trajectory specified by (1)– (3), starting a very small perturbation away from equilibrium point* (-0.478, $\sqrt{3}/2$) *of Ω(x, y), for μ = 0.022.*

This trajectory keeps circling around the equilibrium point, suggesting that it is a STABLE equilibrium point. We can do the same for the negative y-coordinate and show that it is also stable.

Now look at μ = 0.044. The equilateral Lagrange points are (−0.456, ±$\sqrt{3}/2$). Once again, focus on the positive y-coordinate first and use initial conditions (−0.4561, $(\sqrt{3}/2) + 0.0001$,0,0):

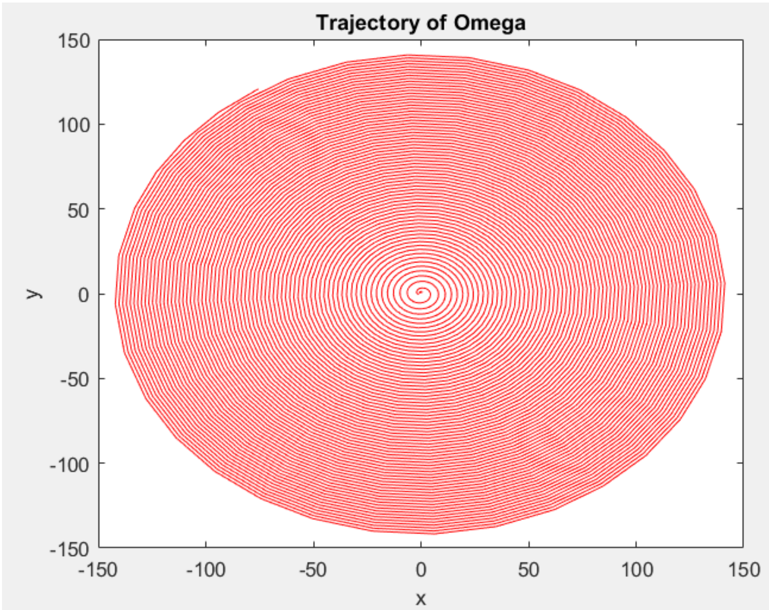

Figure 20: *A graph displaying the trajectory specified by (1)– (3), starting a very small perturbation away from equilibrium point* (-0.456, $\sqrt{3}/2$) *of Ω(x, y), for μ = 0.044.*

This trajectory spirals away from the equilibrium point, suggesting that it is an UNSTABLE equilibrium point. We can do the same for the negative y-coordinate and show that it is also unstable.

Then focus on μ = 0.08. The equilateral Lagrange points are (−0.42, ±$\sqrt{3}/2$). This time, focus on the negative y-coordinate to begin with and use initial conditions (−0.4201, $-(\sqrt{3}/2) - 0.0001$,0,0):

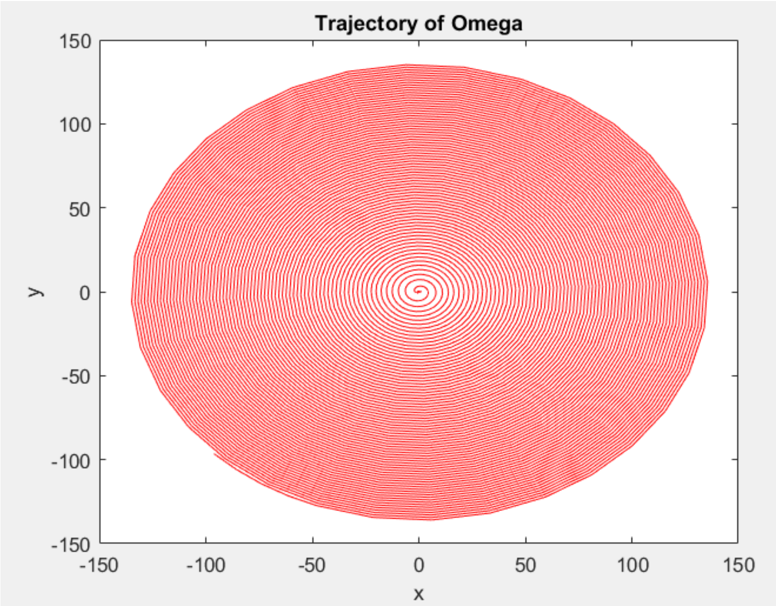

Figure 21: *A graph displaying the trajectory specified by (1)– (3), starting a very small perturbation away from equilibrium point* (-0.42, $-\sqrt{3}/2$) *of Ω(x, y), for μ = 0.08.*

This trajectory spirals away from the equilibrium point, suggesting that it is an UNSTABLE equilibrium point. We can do the same for the positive y-coordinate and show that it is also unstable.

Finally, focus on μ = 0.5. The equilateral Lagrange points are (0, $\pm\sqrt{3}/2$). Once again, focus on the negative y-coordinate to begin with and use initial conditions (-0.0001, $-(\sqrt{3}/2) - 0.0001$,0,0):

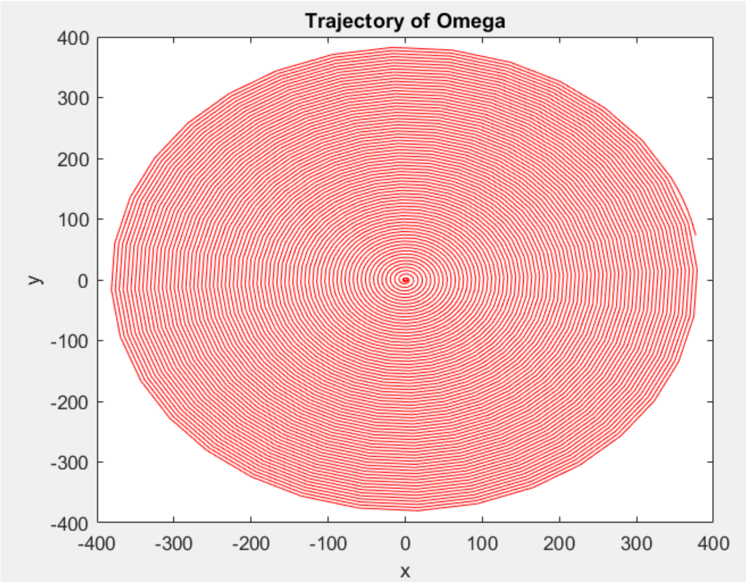

Figure 22: *A graph displaying the trajectory specified by (1)– (3), starting a very small perturbation away from equilibrium point* (0, $-\sqrt{3}/2$) *of Ω(x, y), for μ = 0.5.*

This trajectory spirals away from the equilibrium point, suggesting that it is an UNSTABLE equilibrium point. We can do the same for the positive y-coordinate and show that it is also unstable.

From this, we can identify that the equilateral Lagrange points are stable for μ = 0.008, 0.022, for which it doesn't exactly close in on the solution with spirals, like we typically expect of stable equilibrium points, but rather circles around the equilibrium points, each rotation coming closer to the equilibrium point. However, the equilateral Lagrange points are unstable for $\mu = 0.044, 0.08, 0.5$, where it very clearly spirals away from those points. Hence the critical value μ_c must lie between 0.022 and 0.044. To find μ_c, we can use numerical methods and try out values of μ between 0.022 and 0.044, shortening the interval between which μ_c must lie.

Start with $\mu = 0.033$; the equilateral points are $(-0.467, \pm\sqrt{3}/2)$. Focus on just the positive y-coordinate to begin with and use initial conditions (-0.4671, $(\sqrt{3}/2) + 0.0001$,0,0):

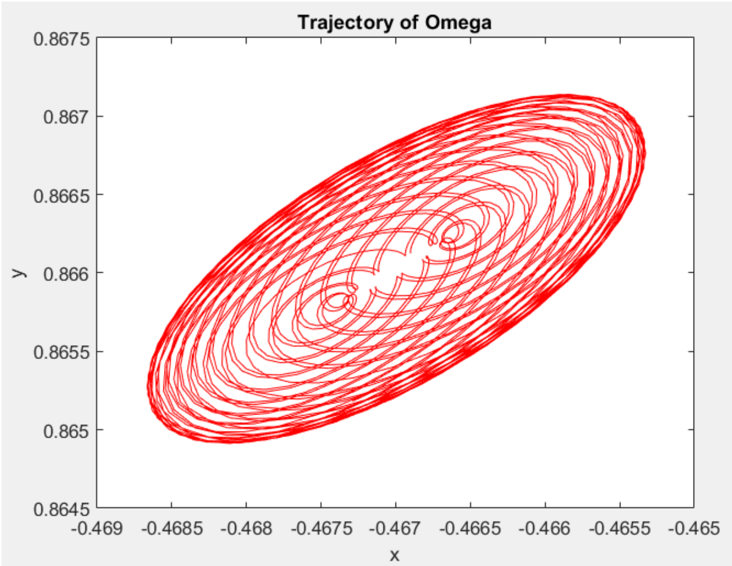

Figure 23: *A graph displaying the trajectory specified by (1)– (3), starting a very small perturbation away from equilibrium point* $(-0.467, \sqrt{3}/2)$ *of Ω(x, y), for* $\mu = 0.033$.

This is clearly a stable equilibrium point; we can do the same for the negative y-coordinate and show that it is also stable. Hence μ_c must lie between 0.033 and 0.044. Try out $\mu = 0.038$ next – the equilateral points are $(-0.462, \pm\sqrt{3}/2)$. Focus on just the positive y-coordinate to begin with and use initial conditions (-0.4621, $(\sqrt{3}/2) + 0.0001$,0,0):

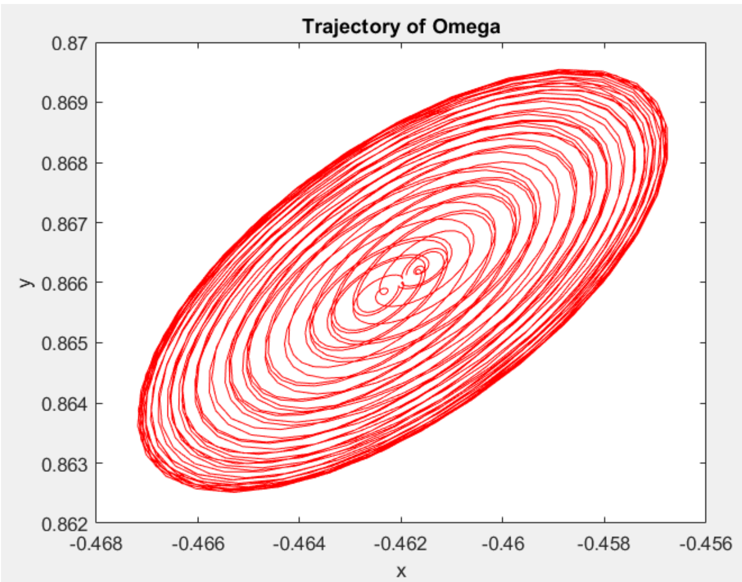

Figure 24: *A graph displaying the trajectory specified by (1)– (3), starting a very small perturbation away from equilibrium point* $(-0.462, \sqrt{3}/2)$ *of Ω(x, y), for* $\mu = 0.038$.

This is clearly a stable equilibrium point; we can do the same for the negative y-coordinate and show that it is also stable. Hence μ_c must lie between 0.038 and 0.044. Try out $\mu = 0.041$ next – the equilateral points are $(-0.459, \pm\sqrt{3}/2)$. Focus on just the positive y-coordinate to begin with and use initial conditions (-0.4591, $(\sqrt{3}/2) + 0.0001$,0,0):

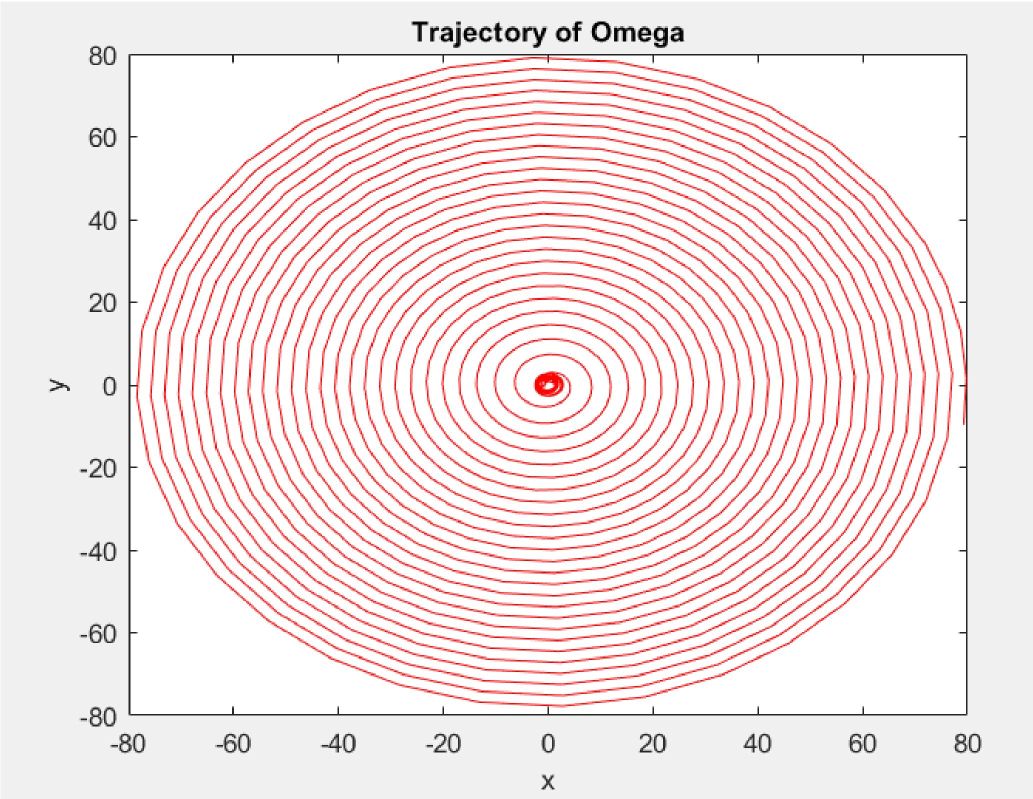

Figure 25: *A graph displaying the trajectory specified by (1)– (3), starting a very small perturbation away from equilibrium point* (-0.459, $\sqrt{3}/2$) *of Ω(x, y), for μ = 0.041.*

This is clearly an unstable equilibrium point; we can do the same for the negative y-coordinate and show that it is also unstable. Hence μ_c must lie between 0.038 and 0.041.

Through repeating the same process of testing a point halfway through the new interval and then selecting the new interval, repeating the same thing after, we eventually get that μ_c must lie between 0.0385 and 0.039, hence we get that $\mu_c = 0.039$ to two significant figures.

QUESTION 6

For the Sun-Jupiter system, $\mu = 9.54 \times 10^{-4}$. As this is less than μ_c, we do expect the equilateral Lagrange points (the location of the Trojans) to be stable equilibrium points, hence the persistence of the Trojans at this point is consistent with the findings above.

The Earth–Moon system has $\mu = 0.012141$. You would initially expect equilateral Lagrange points in the system like the Trojans in the Sun-Jupiter system, because $\mu = 0.012141 < \mu_c$ but the Sun–Jupiter system, and other bigger bodies, would have an impact on the stability of the points, and their location, hence we would not expect analogous Trojans at this point.

Programs

CODE 1

<u>Code for storing the first order ODEs:</u>

```
function [derivative] = first_order_ODEs(t,y_val,m)
x=y_val(1);
y=y_val(2);
u=y_val(3);
v=y_val(4);
r1=sqrt((x+1-m)^2 +y^2);
r2=sqrt((x-m)^2 +y^2);
dwdx=-(m*(x+1-m))-((1-m)*(x-m))+(m*(x+1-m)/(r1)^3)+((1-m)*(x-m)/(r2)^3);
dwdy=-(m*y)-((1-m)*y)+(m*y/(r1)^3)+((1-m)*y/(r2)^3);
dxdt=u;
dydt=v;
dudt=2*v-dwdx;
dvdt=-2*u-dwdy;
derivative=[dxdt; dydt; dudt; dvdt];
end
```

<u>Code for carrying out ode45 to estimate the solution:</u>

```
function solving_ODEs(m,v_val)
y0=[0.2;0;0;v_val];
r_1=sqrt((1.2-m)^2);
r_2=sqrt((0.2-m)^2);
omega=-(0.5*m*(r_1)^2)-(0.5*(1-m)*(r_2)^2)-(m/(r_1))-((1-m)/(r_2));
J=0.5*(v_val)^2+omega;
[X, Y] = meshgrid(-2:0.1:2, -2:0.1:2);
r1=sqrt((X+1-m).^2 +Y.^2);
r2=sqrt((X-m).^2 +Y.^2);
Omega =-(0.5*m*(r1).^2)-(0.5*(1-m)*(r2).^2)-(m./r1)-((1-m)./r2);
levels=[J,J];
levels_1=[J-1,J-1];
levels_2=[J-2,J-2];
tspan = [0, 15];
[~, y_val] = ode45(@(t,y_val) first_order_ODEs(t,y_val, m), tspan, y0);
for i=1:20
    R1=sqrt(((y_val(i,1)+1-m)^2)+(y_val(i,2))^2);
    R2=sqrt(((y_val(i,1)-m)^2)+(y_val(i,2))^2);
    Omega_val =-(0.5*m*(R1)^2)-(0.5*(1-m)*(R2)^2)-(m/R1)-((1-m)/R2);
    J_val=0.5*((y_val(i,3))^2)+0.5*((y_val(i,4))^2)+Omega_val;
    sprintf(num2str(J_val))
end
plot(y_val(:,1), y_val(:,2),'-r')
hold on;
contour(X, Y, Omega, levels,'-y');
hold on;
contour(X, Y, Omega, levels_1,'-g');
hold on;
contour(X, Y, Omega, levels_2,'-b');
legend('trajectory','Omega=J','Omega=J-1','Omega=J-2')
title('Trajectory of Omega')
xlabel('x')
ylabel('y');
end
```

CODE 2

```
function [contour_plot] = contours_2(m)
[X, Y] = meshgrid(-2:0.1:2, -2:0.1:2);
r1=sqrt((X+1-m).^2 +Y.^2);
r2=sqrt((X-m).^2 +Y.^2);
Omega =-(0.5*m*(r1).^2)-(0.5*(1-m)*(r2).^2)-(m./r1)-((1-m)./r2);
J=-5;
levels=zeros(1,200);
for i=1:100
    levels(2*i-1)=J+i/30;
    levels(2*i)=J-i/30;
end
contour_plot=contour(X, Y, Omega,levels);
xlabel('x-axis');
ylabel('y-axis');
title('Contour Plot of Omega');
end
```

CODE 3

<u>Code for storing the first order ODEs:</u>

```
function [derivative] = first_order_ODEs_2(t,y_val,m)
x=y_val(1);
y=y_val(2);
u=y_val(3);
v=y_val(4);
r2=sqrt((x-m)^2 +y^2);
dwdx=(x-m)/2*((r2)^3);
dwdy=y/2*((r2)^3);
dxdt=u;
dydt=v;
dudt=2*v-dwdx;
dvdt=-2*u-dwdy;
derivative=[dxdt; dydt; dudt; dvdt];
end
```

<u>Code for carrying out ode45 to estimate the solution:</u>

```
function solving_ODEs_2(m)
y0=[1;0;0;-1];
tspan = [0, 15];
[~, y_val] = ode45(@(t,y_val) first_order_ODEs_2(t,y_val, m), tspan, y0);
for i=1:20
    R1=sqrt(((y_val(i,1)+1-m)^2)+(y_val(i,2))^2);
    R2=sqrt(((y_val(i,1)-m)^2)+(y_val(i,2))^2);
    Omega_val =-(0.5*m*(R1)^2)-(0.5*(1-m)*(R2)^2)-(m/R1)-((1-m)/R2);
    J_val=0.5*((y_val(i,3))^2)+0.5*((y_val(i,4))^2)+Omega_val;
    sprintf(num2str(J_val))
end
plot(y_val(:,1), y_val(:,2))
title('Trajectory of Omega');
xlabel('x');
ylabel('y');
end
```

CODE 4

```
function[equilibrium] = stability(x0,y0,u0,v0,m)
y0=[x0;y0;u0;v0];
tspan = [0,500];
[t, y_val] = ode45(@(t,y_val) first_order_ODEs(t,y_val, m), tspan, y0);
equilibrium=table(t,y_val(:,1),y_val(:,2),'VariableNames',{'time','x','y'});
for i=1:20
    R1=sqrt(((y_val(i,1)+1-m)^2)+(y_val(i,2))^2);
    R2=sqrt(((y_val(i,1)-m)^2)+(y_val(i,2))^2);
    Omega_val =-(0.5*m*(R1)^2)-(0.5*(1-m)*(R2)^2)-(m/R1)-((1-m)/R2);
    J_val=0.5*((y_val(i,3))^2)+0.5*((y_val(i,4))^2)+Omega_val;
    sprintf(num2str(J_val))
end
plot(y_val(:,1), y_val(:,2),'-r')
title('Trajectory of Omega')
xlabel('x')
ylabel('y');
end
```