Catam Additional Projects
Computational Projects Manual
(July 2019 Edition)

4/15/2020

- Restricted Three-Body Problem
- Programs

1 Restricted Three-Body Problem

We have

$$\ddot{x} - 2\dot{y} = -\frac{\partial \Omega}{\partial x}, \ddot{y} + 2\dot{x} = -\frac{\partial \Omega}{\partial y} \tag{$*$}$$

$$\Omega = -\frac{1}{2}(x^2 + y^2) - \frac{\mu}{\sqrt{(x+1-\mu)^2 + y^2}} - \frac{1-\mu}{\sqrt{(x-\mu)^2 + y^2}}. \tag{$**$}$$

1.1 Question 1

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

$$\begin{aligned}\frac{\mathrm{d}J}{\mathrm{d}t} &= \frac{1}{2}(2\dot{x}\ddot{x} + 2\dot{y}\ddot{y}) + \frac{\mathrm{d}\Omega}{\mathrm{d}t} \\ &= \dot{x}\ddot{x} + \dot{y}\ddot{y} + \frac{\partial \Omega}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial \Omega}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t} \\ &= \dot{x}\ddot{x} + \dot{y}\ddot{y} - \ddot{x}\dot{x} + 2\dot{y}\dot{x} - \ddot{y}\dot{y} - 2\dot{x}\dot{y} \\ &= 0.\end{aligned}$$

Hence J=const.= $\frac{1}{2}(u_0^2 + v_0^2) + \Omega(x_0, y_0)$ via initial conditions.

For any x,y,

$$\begin{aligned}\Omega(x, y) &= J - \frac{1}{2}(\dot{x}^2 + \dot{y}^2) \\ &= \frac{1}{2}(u_0^2 + v_0^2) + \Omega(x_0, y_0) - \frac{1}{2}(\dot{x}^2 + \dot{y}^2) \\ &\leq \frac{1}{2}(u_0^2 + v_0^2) + \Omega(x_0, y_0)\end{aligned}$$

since $\dot{x}^2, \dot{y}^2 \geq 0$.

1.1.1 Solving the System of Equations for the Third Body: Programming Task

A program using *ode*45 is listed on page 21, named Solveode(p,a,b,x0,y0,u0,v0,rt,at). Here we use p for μ, a,b for initial and final time respectively, rt for relative tolerance and at for absolute tolerance. Other quantities used the same as in the setup.

points(t,x,y)/variables	μ	t_0	t_f	rt	at	J
(0.161,0.333,0.449)	0.4	0	5	1×10^{-8}	1×10^{-10}	0.261
(3.411,-3.154,-1.629)	0.4	0	5	1×10^{-8}	1×10^{-10}	0.261
(0.656,1.249,0.356)	0.4	0	5	1×10^{-5}	1×10^{-6}	0.261

Table 1: E_ns applying 3 methods at $x_n = 0.16$

The program gives two pictures: x,y against t and y against x. Check the results using 2 sets of inputs for 3 points to make sure the program is on the right track. Here we use $x_0 = 0.1, y_0 = 0.2, u_0 = 1, v_0 = 2$ for all three points. By choosing different points on the same plot and changing tolerance, we produce the same J value, as we expect. See Tbl.1.

1.2 Question 2

From $(*)$ and $(**)$ with Ω approximated as $-\frac{0.5}{\sqrt{(x-\mu)^2+y^2}}$, use polar coordinates then we have

$$\ddot{r}cos\theta - 2\dot{r}sin\theta\dot{\theta} - rsin\theta\ddot{\theta} - rcos\theta\dot{\theta}^2 - 2\dot{r}sin\theta - 2rcos\theta\dot{\theta} = -\frac{0.5cos\theta}{r^2} \tag{1}$$

$$\ddot{r}sin\theta + 2\dot{r}cos\theta\dot{\theta} + rcos\theta\ddot{\theta} - rsin\theta\dot{\theta}^2 + 2\dot{r}cos\theta - 2rsin\theta\dot{\theta} = -\frac{0.5sin\theta}{r^2}. \tag{2}$$

Add and Subtract (1) and (2) to get

$$\dot{\theta} = -1 \pm \frac{\sqrt{2r^3(2r^3 - 1 + 2r^2\ddot{r})}}{r^2}, \quad 2\dot{r}\dot{\theta} + r\ddot{\theta} + 2\dot{r} = 0. \tag{3}$$

$$\Rightarrow r \times \frac{\mathrm{d}\sqrt{2r^3(2r^3 - 1 + 2r\ddot{r})}}{\mathrm{d}t} = 0 \quad \Rightarrow \sqrt{2r^3(2r^3 - 1 + 2r\ddot{r})} = k, k\ const.$$

$$\Rightarrow \dot{\theta} = -1 + \frac{k}{r^2} \tag{4}$$

$$(1)(2) \Leftrightarrow (3)(4) \Leftrightarrow \ddot{r} = -V'(r), \quad where \ \ V(r) = -\frac{1}{2r} + \frac{r^2}{2} + \frac{k^2}{2r^2} + const.$$

For r(t)$\equiv$ a, $\dot{\theta}, V$ are constant. Hence (3)(4) has circular orbit solutions with k and a satisfy

$$k = \pm\sqrt{\frac{a}{2} + a^4} \quad since \ \ \dot{r} = \ddot{r} = 0 \Rightarrow V'(a) = 0.$$

1.2.1 Modifying the Program with approximated Ω: Programming Task

A program modified for Question 2 is listed on page 21, named SolveodeQ2(p,a,b,x0,y0,rt,at).

Choose a value of a, say, 0.1. Let $\mu = 0.5$, $x_0 = 0.4$, $y_0 = 0$. Then $u_0 = \dot{x}(0) = -y_0\dot{\theta}(0), v_0 = \dot{y}(0) = (x(0) - \mu)\dot{\theta}(0)$. Run the program with t final=30, relative tolerance=10^{-8}, absolute tolerance=10^{-10}. Results in Fig.1 and Fig.2.

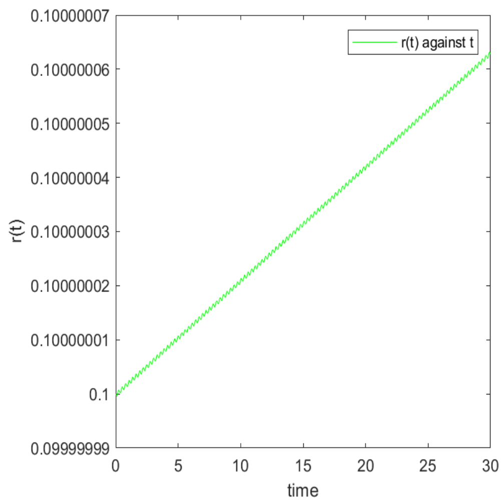


Figure 1: r against t

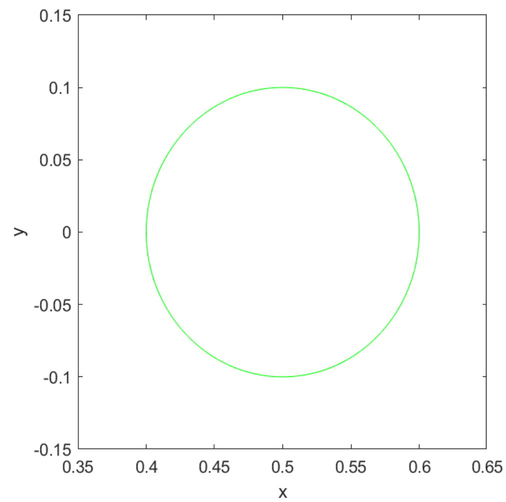


Figure 2: y against x

We can find y(x) a perfect circle centred at $(\mu, 0)$ with radius a, consistent with the analytic solutions. We also find r(t) only vary by very tiny amount, which comes from the setting of our relative and absolute tolerance. We hence conclude the modified program accurately reproduces the analytic circular-orbit solutions.

1.3 Question 3

i) Fig.3-10 shown below obtained by taking x_0=0.32, y_0=0, u_0=0 and v_0=-1.0, -1.5, -1.73, -1.78, -1.853, -1.858, -2.3, -2.31 in turn using program **SolveodeQ3ver2 (p,a,b,x0,y0,u0,v0,rt,at) (listed on page 22)** to integrate from t=0 to t=30. Trajectories in green and forbidden region in turquoise.

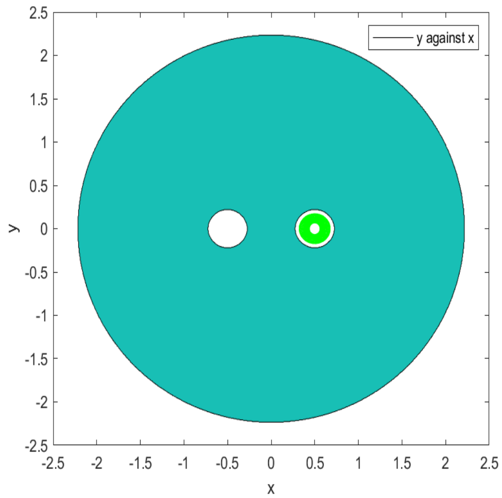


Figure 3: $v_0 = -1.0$

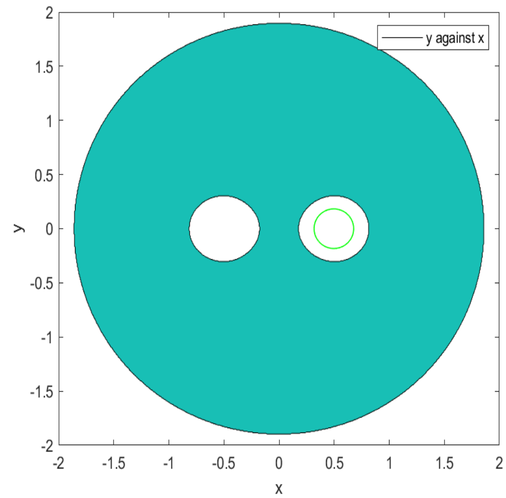


Figure 4: $v_0 = -1.5$

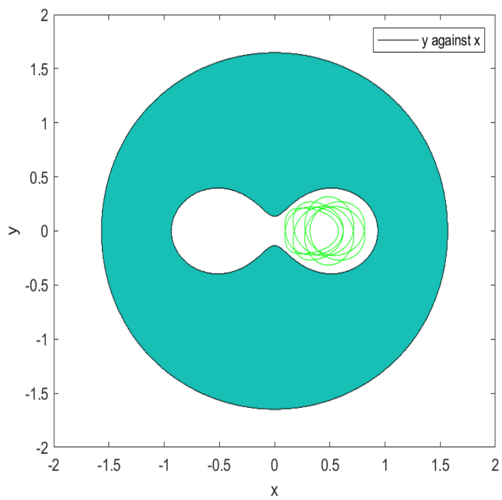


Figure 5: $v_0 = -1.73$

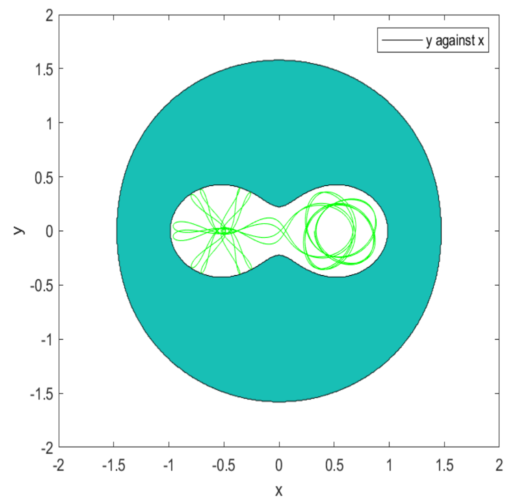


Figure 6: $v_0 = -1.78$

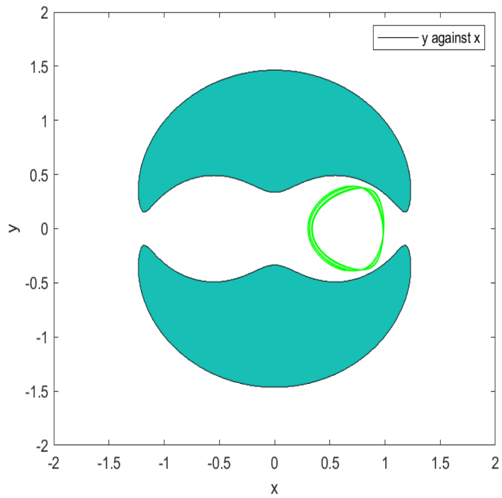


Figure 7: $v_0 = -1.853$

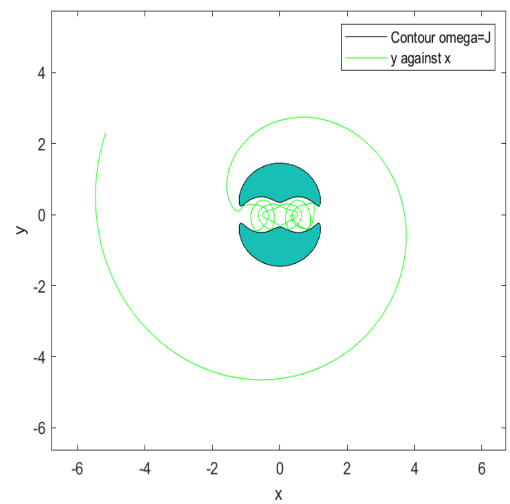


Figure 8: $v_0 = -1.858$

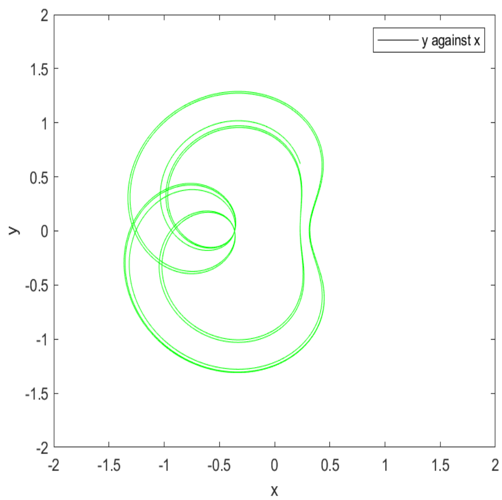


Figure 9: $v_0 = -2.3$

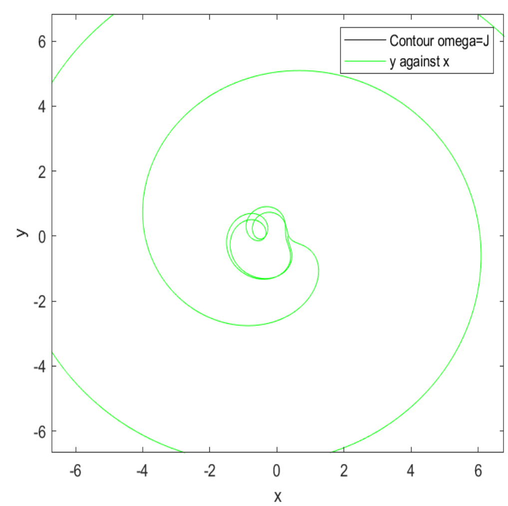


Figure 10: $v_0 = -2.31$

ii) At t=30, x and y are shown in Tbl.2.

values/v_0	-1.0	-1.5	-1.73	-1.78	-1.853	-1.858	-2.3	-2.31
x	0.4713	0.5794	0.7748	0.6896	0.8019	0.2301	1.1220	-0.9265
y	0.0589	-0.1605	-0.0553	-0.1642	-0.3766	-0.0137	2.3380	1.0891

Table 2: x and y values at t=30

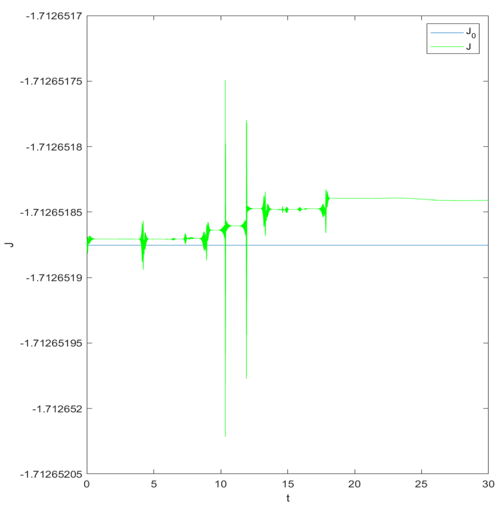


Figure 11: deviation of J compared with J_0 when $v_0 = -1.858$, t final=30

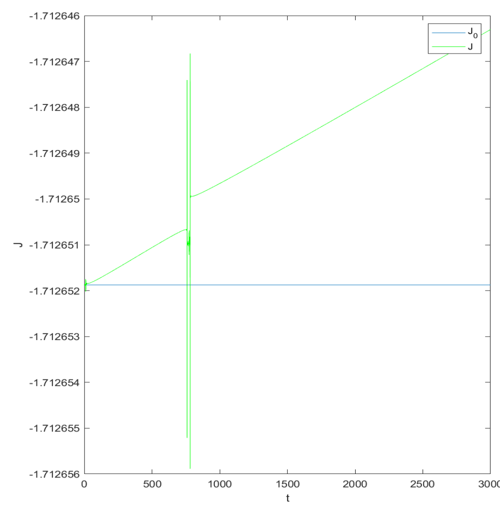


Figure 12: deviation of J compared with J_0 when $v_0 = -1.858$, t final=3000

From t=0 to t=30, J remains relatively constant (Fig.11). By t=3000, J only deviates by 0.005839% (Fig.12). Using the conclusion from Question 1 that, J is constant, conclude that the values of x and y at t=30 is rather accurate up to some level of numerical error.

Comment on Fig.3-10:

As can be seen throughout the pictures, if the third body starts from (0.32,0) with velocity (0,-1), it moves in an elliptical orbit around P_h with one of the focus at (μ,0). As -v_0 increases to 1.5, it moves rather stably in circle-like orbit with larger radius. The corresponding allowed region also **expand** with the trajectories remaining inside.

As -v_0 continues to increase, the two circular allowed region becomes more and more stretched towards each other until -v_0 reaches about 1.6963, they **touch** and become one. The third body becomes unstable and goes around P_h and P_l in turn, as can be seen in Fig.6. It goes back and forth around (μ-1,0) while circles around (μ,0).

The forbidden region breaks and becomes two halves (i.e. the allowed region becomes **connected**) when -v_0 reaches 1.8495. Then in Fig.7, the third body becomes stable and goes around P_h again. -v_0 continues increasing and it goes around P_h and P_l unstably. It finally slips out the eggshell-like forbidden region and starts to go spiral with increasing radius around P_h and P_l together.

The forbidden region vanishes (i.e. allowed region **covers all space**) when -v_0 reaches around 2.0317, where it then goes around (μ-1,0) once then passes by (μ,0) and repeats. Several turns after, it becomes spiral again. As -v_0 increases, it goes spiral more quickly.

From above we can see that the allowed region is actually a **useful** guide to the size of the trajectory. The trajectory is first confined within the circular allowed region around P_h. As allowed region expands, it becomes more freely to move and covers larger space. As allowed region vanishes, it can go to infinity.

The **most suitable** value for the third body to travel from P_h to P_l is round -1.785, where it can go to P_l from time to time and move around (μ-1,0) rather stably. The connected forbidden region also prevents it from escaping to infinity, as v_0=-1.858 would do.

1.4 Question 4

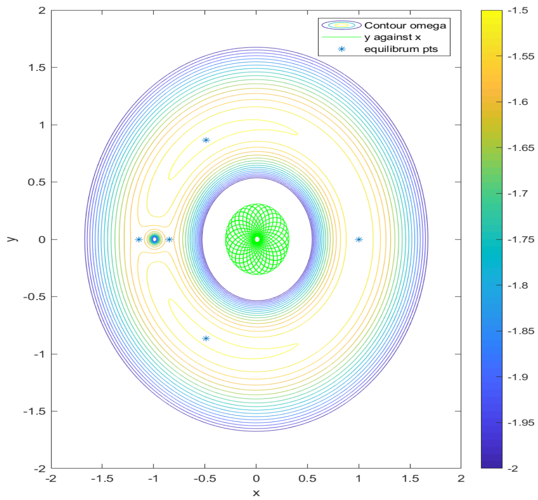


Figure 13: Ω contour plots when $\mu =$ 0.01

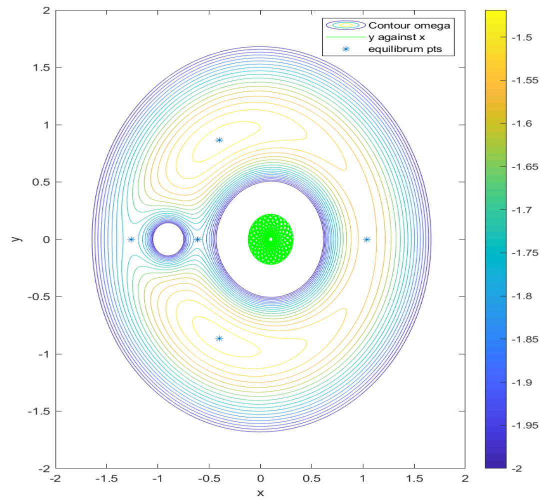


Figure 14: Ω contour plots when $\mu = 0.1$

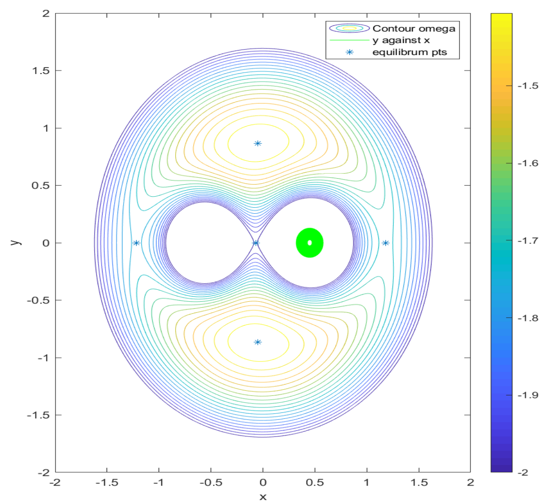


Figure 15: Ω contour plots when $\mu =$ 0.45

Each contour plot of Ω reveals 5 equilibrium points, as can be seen from Fig.13-15 above. Related programs see page 23 **SolveodeQ4ct(p,a,b,x0,y0,u0,v0,rt,at)**. Here choose μ to be 0.01, 0.10, 0.45. There are concentric circles around the equilateral lagrange points (marked points above and below x-axis), implying that they are extremum (maximum, more precisely, from the color bar). The collinear lagrange points (marked points on the x-axis) are at the intersection of contour lines, implying saddle points.

Now start trajectories a small distance away from the collinear lagrange points and integrate forward in time. Fig.16-25 below show the results using $\mu = 0.01, 0.10$ and 0.45 integrating from $t = 0$ to $t = 30$ or 60. A collection of information of the tested points is in Tbl.3. (x_0^*, y_0^*) is the equilibrium point we are examining and x_0, y_0, u_0, v_0 are the initial conditions of the tested trajectories. $'$L$''$M$''$R$'$ stand for left-most, middle, right-most collinear lagrange points respectively.

μ	position	x_0^*	y_0^*	x_0	y_0	u_0	v_0
0.01	L	-1.1468	0	-1.1467	0.0001	0	0
0.01	M	-0.8481	0	-0.8482	0.0001	0	0
0.01	R	1.0042	0	1.0041	0.0001	0	0
0.10	R	1.0416	0	1.0415	0.0001	0	0
0.45	R	1.1806	0	1.1805	0.0001	0	0

Table 3: A list of examined collinear lagrange points

The corresponding trajectories are shown in Fig.16-25, using program listed on page 25, **SolveodeQ4(p,a,b,x0,y0,u0,v0,rt,at)**.

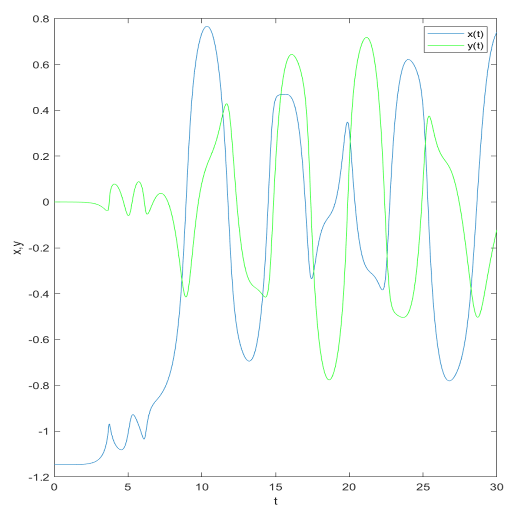


Figure 16: x(t),y(t) at the left-most collinear lagrange pt. when μ =0.01

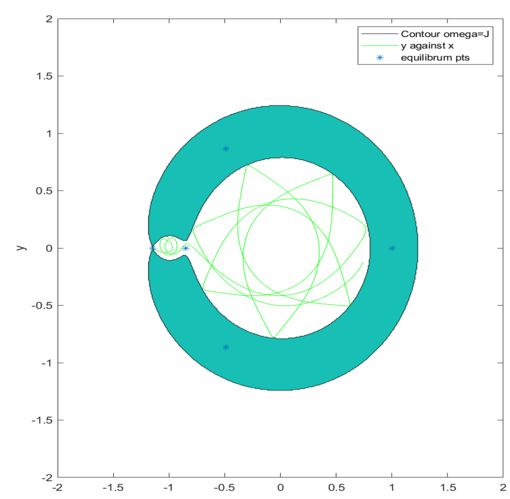


Figure 17: y(x) at the left-most collinear lagrange pt. when μ =0.01

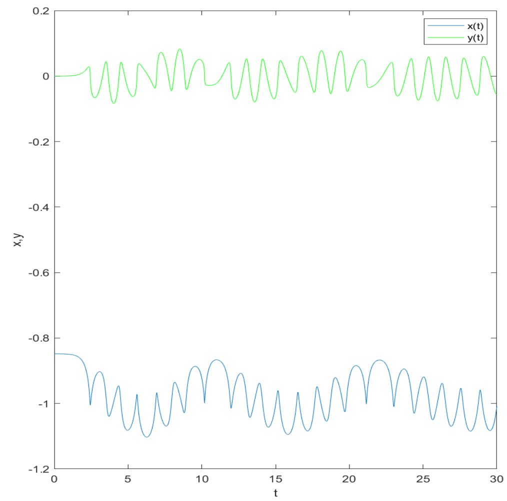


Figure 18: x(t),y(t) at the middle collinear lagrange pt. when μ =0.01

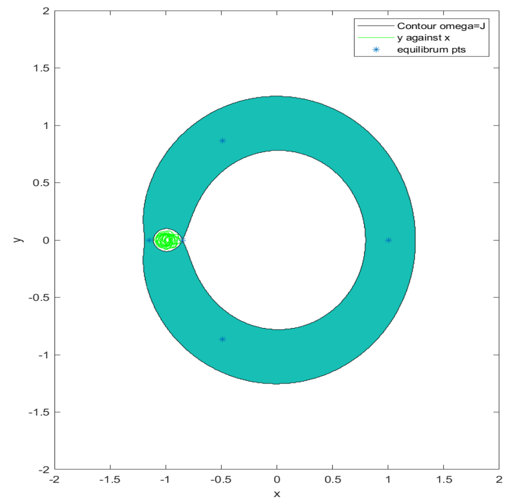


Figure 19: y(x) at the middle collinear lagrange pt. when μ =0.01

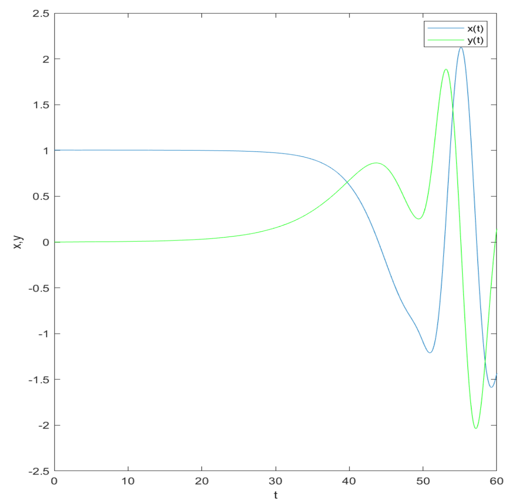


Figure 20: x(t),y(t) at the right-most collinear lagrange pt. when μ =0.01

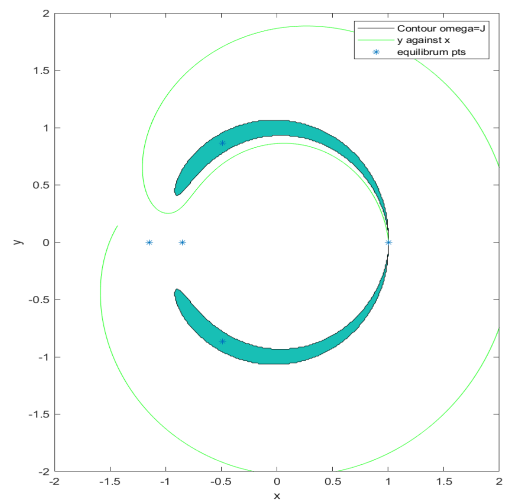
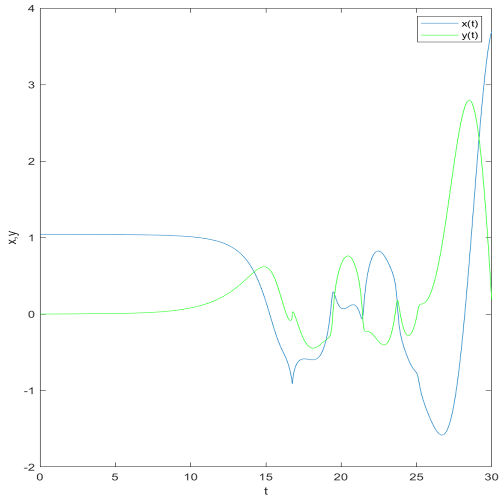


Figure 21: y(x) at the right-most collinear lagrange pt. when μ =0.01



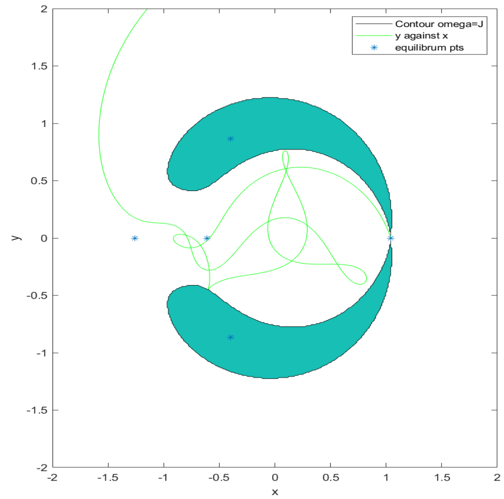
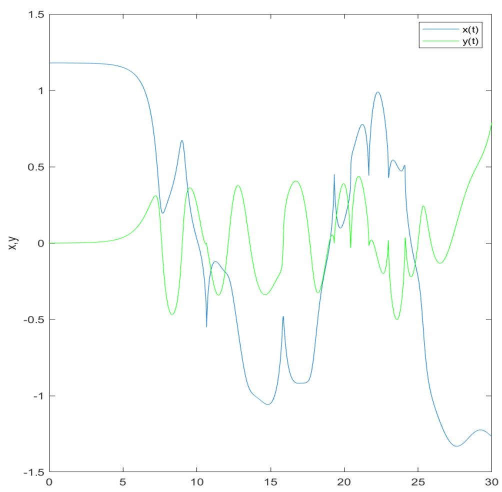


Figure 22: x(t),y(t) at the right-most collinear lagrange pt. when μ =0.1

Figure 23: y(x) at the right-most collinear lagrange pt. when μ =0.1



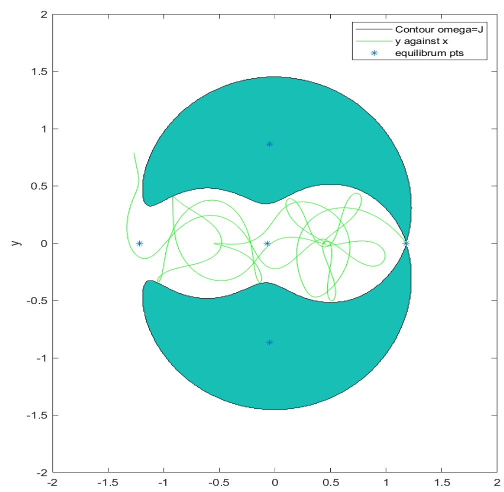


Figure 24: x(t),y(t) at the right-most collinear lagrange pt. when μ =0.45

Figure 25: y(x) at the right-most collinear lagrange pt. when μ =0.45

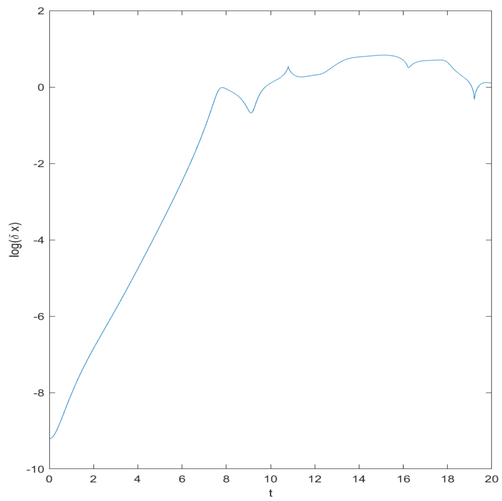


Figure 26: $\log(\delta x)$ against t starting from (1.1805,0) near equilateral lagrange pt. (1.1806,0) when μ =0.45

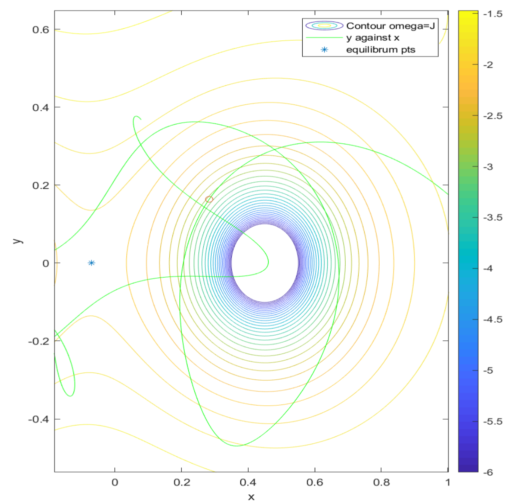


Figure 27: red pt. circled at t $\approx$ 2s around (μ,0) when μ =0.45 (enlarged)

In Tbl.3 and Fig.16-25, we used perturbation of value 0.0001. This value of δx has well captured the infinitesimal perturbations, as can be seen from Fig.26-27 above, using a slightly modified program, named **SolveodeQ4test(p,a,b,x0,y0,u0,v0,rt,at)**, (listed on page 26). Here apply $\mu = 0.45$ for example.

When we first start near the saddle point, say (1.1805,0), $\log(\delta x)$ is an increasing straight line, implying the exponential behavior of δx, i.e., the trajectory diverges, which is consistent with what we have seen in Fig.24-25. Afterwards $\log(\delta x)$ becomes oscillating, implying the third body is now away from the saddle point (see Fig.27 red circled point at $t \approx 2$).

Hence we might conclude from Fig.16-27 that no matter what value of μ is in the range (0, 1/2], the trajectory diverges around the equilateral lagrange points, i.e., the collinear lagrange points are unstable, independent of μ.

1.4.1 Linear Stability Analysis of Collinear Lagrange Points

Knowing

$$\frac{\partial \Omega}{\partial x} = -x + \frac{\mu(x+1-\mu)}{(\sqrt{(x+1-\mu)^2+y^2})^3} + \frac{(1-\mu)(x-\mu)}{(\sqrt{(x-\mu)^2+y^2})^3} \tag{4}$$

$$\frac{\partial \Omega}{\partial x} = -y + \frac{\mu y}{(\sqrt{(x+1-\mu)^2+y^2})^3} + \frac{(1-\mu)y}{(\sqrt{(x-\mu)^2+y^2})^3} \tag{5}$$

$$\frac{\partial^2\Omega}{\partial x^2} = -1+\frac{\mu((x+1-\mu)^2+y^2)-3\mu(x+1-\mu)^2}{(\sqrt{(x+1-\mu)^2+y^2})^5}+\frac{(1-\mu)((x-\mu)^2+y^2)-3(1-\mu)(x-\mu)^2}{(\sqrt{(x-\mu)^2+y^2})^5} \tag{6}$$

$$\frac{\partial^2\Omega}{\partial y^2} = -1+\frac{\mu((x+1-\mu)^2+y^2)-3\mu y^2}{(\sqrt{(x+1-\mu)^2+y^2})^5}+\frac{(1-\mu)((x-\mu)^2+y^2)-3(1-\mu)y^2}{(\sqrt{(x-\mu)^2+y^2})^5} \tag{7}$$

$$\frac{\partial^2\Omega}{\partial x\partial y} = -\frac{3\mu(x+1-\mu)y}{(\sqrt{(x+1-\mu)^2+y^2})^5}-\frac{3(1-\mu)(x-\mu)y}{(\sqrt{(x-\mu)^2+y^2})^5} \tag{8}$$

Let a lagrange point be at (x_0, y_0) where Ω=Ω_0. Let x=$x_0 + \delta$x, y=$y_0 + \delta$y. Then

$$\Omega = \Omega_0 + \frac{1}{2}\Omega_{xx}(\delta x)^2 + \frac{1}{2}\Omega_{yy}(\delta y)^2 + \Omega_{xy}\delta x\delta y \tag{9}$$

by taylor expansions, ignoring smaller terms and noticing Ω_x^0=Ω_x^0=0 at equilibrium points. With $\frac{\partial\Omega}{\partial(x_0+\delta x)} = \frac{\partial\Omega}{\partial\delta x}\frac{\partial\delta x}{\partial(x_0+\delta x)} = \frac{\partial\Omega}{\partial\delta x}$,

$$(*) \Rightarrow \delta\ddot{x} - 2\delta\dot{y} = -\Omega_{xx}^0\delta x - \Omega_{xy}^0\delta y, \quad \delta\ddot{y} + 2\delta\dot{x} = -\Omega_{yy}^0\delta y - \Omega_{xy}^0\delta x \tag{10}$$

Insert $\delta x = \delta x_0 e^{\lambda t}, \delta y = \delta y_0 e^{\lambda t}$ into (10) to get

$$\lambda^2 = \frac{-(\Omega_{xx}^0+\Omega_{yy}^0+4) \pm \sqrt{(\Omega_{xx}^0+\Omega_{yy}^0+4)^2 - 4(\Omega_{xx}^0\Omega_{yy}^0 - (\Omega_{xy}^0)^2)}}{2}. \tag{11}$$

Since y=0 implies Ω_{xy}^0=0 at collinear lagrange points,

$$\Rightarrow \Omega_{xx}^0 = -1-2A, \quad \Omega_{yy}^0 = -1+A, \quad \lambda^2 = \frac{(A-2)\pm\sqrt{(9A-8)A}}{2} \tag{12}$$

where A=$\frac{\mu}{|x+1-\mu|^3} + \frac{1-\mu}{|x-\mu|^3} > 0$. The Hessian matrix

$$\begin{pmatrix} \Omega_{xx}^0 & \Omega_{xy}^0 \\ \Omega_{xy}^0 & \Omega_{yy}^0 \end{pmatrix} = \begin{pmatrix} -1-2A & 0 \\ 0 & -1+A \end{pmatrix}. \tag{13}$$

We already know -1-2A < 0. To determine the feature of the critical points, what is left to assess is to find out the sign of -1+A. Using Ω_x^0=0 we get

$$-1+A = -1+\frac{x}{x+1-\mu}+\frac{1-\mu}{(x+1-\mu)|x-\mu|^3}.$$

This is greater than 0 for any $\mu \in (0, 1/2]$ when

$$-1+\frac{x}{x+1-\mu}+\frac{1-\mu}{(x+1-\mu)|x-\mu|^3} > -1+\frac{x}{x+1-\mu}+\frac{1-\mu}{x+1-\mu} = 0 \tag{14}$$

which is proved due following process:
• Case 1, $x < \mu - 1$, then

$$\frac{1-\mu}{x+1-\mu} < 0, \frac{1}{|x-\mu|^3} < 1 \Rightarrow (14)$$

• Case 2, $\mu - 1 < x < \mu$, then

$$\frac{1-\mu}{x+1-\mu} > 0, \frac{1}{|x-\mu|^3} > 1 \Rightarrow (14)$$

• Case 3, $x > \mu$, let

$$f(x) = \Omega_x^0 = -x + \frac{\mu(x+1-\mu)}{|x+1-\mu|^3} + \frac{(1-\mu)(x-\mu)}{|x-\mu|^3} \equiv 0. \quad (15)$$

When $\mu = \mu_{max} = 1/2$, if $x \geq \mu + 1 = 3/2$, then set $x = 3/2 + k, k \geq 0$.

$$\Rightarrow f(x) = -\frac{3}{2} - k + \frac{1}{2(2+k)^2} + \frac{1}{2(1+k)^2} < -\frac{3}{2} - k + \frac{1}{4} + \frac{1}{2} = -\frac{3}{4} - k < 0$$

Contradict to (15). So $\mu < x < \mu + 1$ when μ is at its maximum.
When $\mu \in (0,1/2]$, differentiate (15) with respect to μ on both sides to get

$$\frac{\mathrm{d}x}{\mathrm{d}\mu} = \frac{\frac{x+1+\mu}{(x+1-\mu)^3} + \frac{2-x-\mu}{(x-\mu)^3}}{1 + \frac{2\mu}{(x+1-\mu)^3} + \frac{2(1-\mu)}{(x-\mu)^3}} > 0$$

since $2 - x - \mu > 1 - 2\mu \geq 0$. Hence x is monotone decreasing as μ goes down from 1/2, i.e., x never goes beyond (or reaches) 3/2. Then $\mu < x < \mu + 1$ and (14) is true.

Hence we conclude that $-1 + A > 0$. Hessian matrix has determinant less than zero and is indefinite, meaning collinear lagrange points are at saddle points.

Now $A > 1 \Rightarrow (A-2)^2 - (9A-8)A = 4(1+2A)(1-A) < 0 \Rightarrow A - 2 + \sqrt{(9A-8)A} > 0$, $A - 2 - \sqrt{(9A-8)A} < 0$. Hence (12) yields one positive real λ. The equations

$$\delta x = \Sigma_{i=1}^4 \delta x_0 e^{\lambda_i t}, \quad \delta y = \Sigma_{i=1}^4 \delta y_0 e^{\lambda_i t} \quad (16)$$

imply Ω's divergence close to (x_0, y_0), confirming that collinear lagrange points are unstable for all $\mu \in (0,1/2]$ (does not depend on μ).

1.5 Question 5

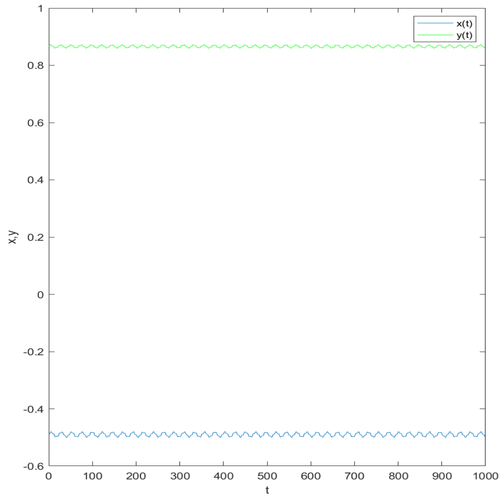


Figure 28: x(t),y(t) at the equilateral lagrange pt. when μ =0.01

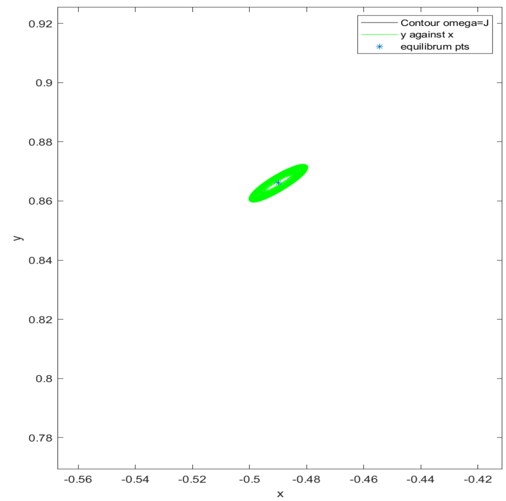


Figure 29: y(x) at the equilateral lagrange pt. when μ =0.01 (enlarged)

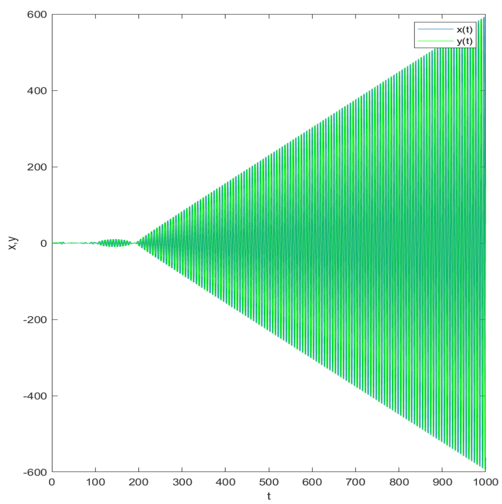


Figure 30: x(t),y(t) at the equilateral lagrange pt. when μ =0.1

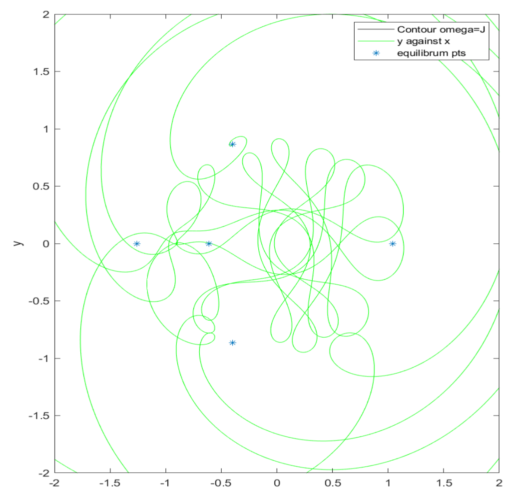


Figure 31: y(x) at the equilateral lagrange pt. when μ =0.1

μ	x_0^*	y_0^*	x_0	y_0	u_0	v_0	stability
0.01	-0.490	0.866	-0.489	0.867	0	0	stable
0.025	-0.475	0.866	-0.474	0.867	0	0	stable
0.05	-0.450	0.866	-0.449	0.867	0	0	unstable
0.1	-0.400	0.866	-0.399	0.867	0	0	unstable
0.5	0	0.866	0.001	0.867	0	0	unstable

Table 4: A list of examined equilateral lagrange points

In the program we integrate from t=0 to t=1000 and w.l.o.g. start the perturbation around the upper equilateral lagrange point (due to x- axial symmetry of the equilibrium points).

From Fig.28-31 and Tbl.4, we find that as μ rises, the equilateral lagrange points goes from stable to unstable. There is a critical value for μ between 0.025 and 0.05. To make it accurate to two decimal places, use bisection method and find the following Tbl.5. Start with $\mu = 0.025$.

μ	x_0^*	y_0^*	x_0	y_0	u_0	v_0	stability
0.025	-0.475	0.866	-0.474	0.867	0	0	stable
0.0375	-0.4625	0.866	-0.4624	0.867	0	0	stable
0.04375	-0.45625	0.866	-0.45624	0.867	0	0	unstable
0.040625	-0.45937	0.866	-0.45936	0.867	0	0	unstable

Table 5: A list of tested values of μ_c using bisection method

Hence we conclude that the critical value $\mu = 0.04$, up to two decimal places.

1.5.1 Linear Stability Analysis of Equilateral Lagrange Points

At equilateral points, $x = \mu - 1/2$ and $y = \pm\sqrt{3}/2$. Again from (6)(7)(8) we obtain

$$\Omega^0_{xx} = -\frac{3}{4}, \quad \Omega^0_{yy} = -\frac{9}{4}, \quad \Omega^0_{xy} = \pm\frac{3\sqrt{3}}{4}(1-2\mu). \tag{17}$$

The Hessian matrix is now

$$\begin{pmatrix} -\frac{3}{4} & \pm\frac{3\sqrt{3}}{4}(1-2\mu) \\ \pm\frac{3\sqrt{3}}{4}(1-2\mu) & -\frac{9}{4} \end{pmatrix} \Rightarrow |H_1| < 0, \quad |H_2| > 0 \quad \forall \mu \in (0, 1/2]. \tag{18}$$

H is negative definite and the equilateral points are at maximums.

Use (11) to get

$$\lambda^2 = -\frac{1}{2} \pm \frac{\sqrt{27(1-2\mu)^2 - 23}}{4}. \tag{19}$$

The equilateral lagrange points are stable if they have periodic orbits (centre), i.e., all λ's are pure imaginary, λ^2 is real and negative, in which case δx and δy only get oscillating sine and cosine terms.

$$\Rightarrow \lambda^2 < 0, \quad 27(1-2\mu)^2 - 23 \geq 0 \Rightarrow \mu \in (0, \frac{1}{2} - \frac{1}{2}\sqrt{\frac{23}{27}}) \approx (0, 0.03852).$$

and unstable if $\mu \in (\frac{1}{2} - \frac{1}{2}\sqrt{\frac{23}{27}}, \frac{1}{2}]$. The critical value μ_c is $\frac{1}{2} - \frac{1}{2}\sqrt{\frac{23}{27}}$. This confirms our previous estimation of μ_c accurate to 1%.

For the stable cases where λ are all pure imaginary, we get periodic harmonic motions of the third body with all sine and cosine terms.

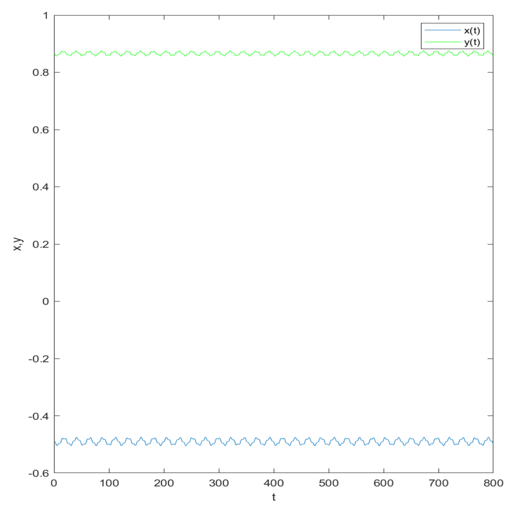


Figure 32: x(t),y(t) at the equilateral lagrange pt. when μ =0.01

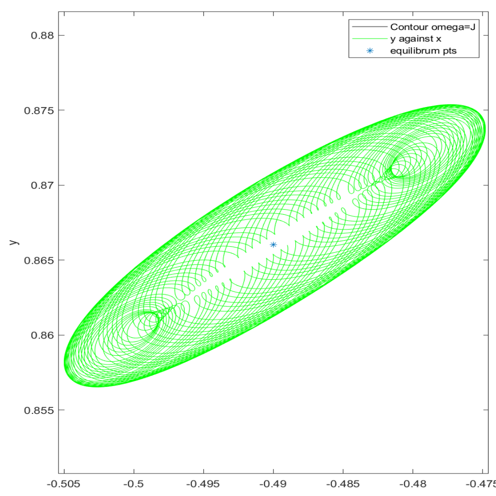


Figure 33: y(x) at the equilateral lagrange pt. when μ =0.01 (enlarged)

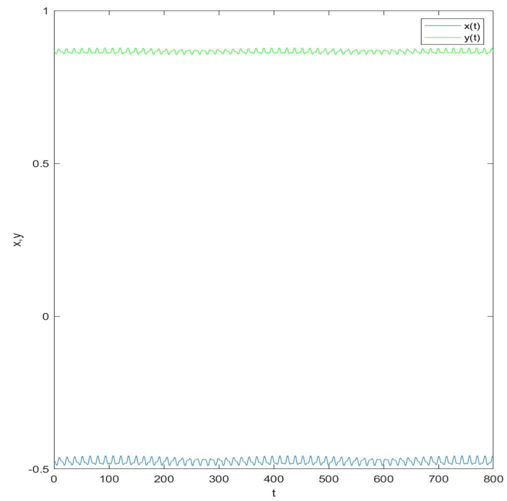


Figure 34: x(t),y(t) at the equilateral lagrange pt. when μ =0.025

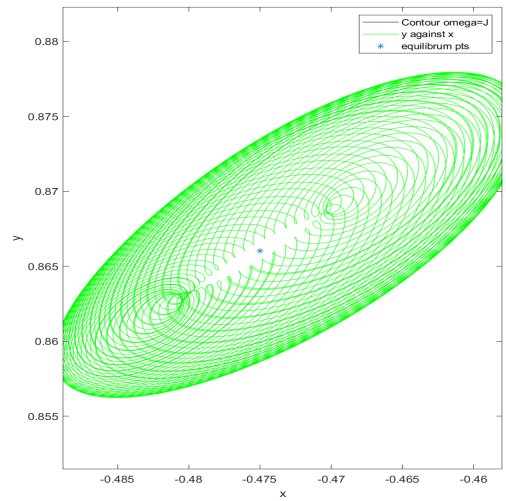


Figure 35: y(x) at the equilateral lagrange pt. when μ =0.025 (enlarged)

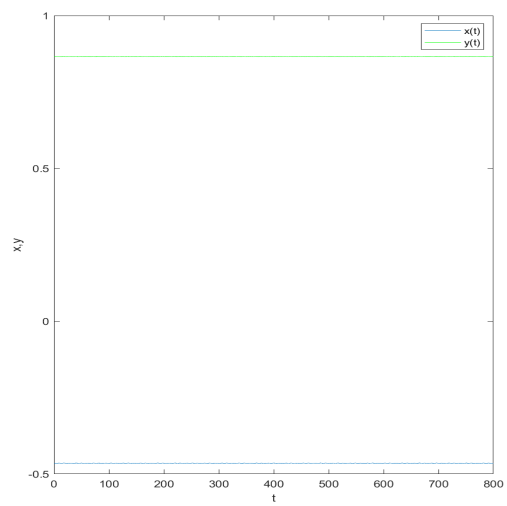


Figure 36: x(t),y(t) at the equilateral lagrange pt. when μ =0.035

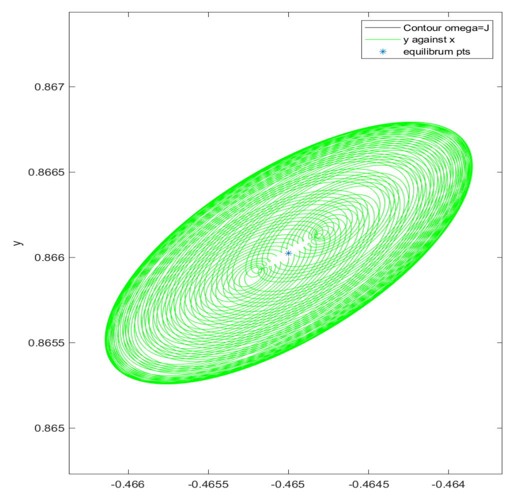


Figure 37: y(x) at the equilateral lagrange pt. when μ =0.035 (enlarged)

It can be seen from Fig.32-37 that, for the stable cases, the trajectory consists of overlapped sandglass shaped orbits and has an elliptical outline. The focus of the orbit goes through a displacement as time develops. As μ goes away from the critical value (i.e., decreases), the trajectory becomes more stretched with less stable distance to the equilibrium point, implying a less stable equilateral lagrange point.

1.6 Question 6

Since $\mu = 9.45 \times 10^{-4} \ll \mu_c$, the Sun-Jupiter equilateral lagrange points are stable. The persistence of the Trojans is then consistent with our conclusion about the stable cases in Question 5, where they perform harmonic motion and only move in a small region around these equilibrium points.

In the Earth-Moon system, however, $\mu = 0.012141$ only slightly less than μ_c. Our original setup about the restricted three-body problems, where the conditions

- third bodies are taken to be much smaller than the first two
- the first two bodies move in circular orbits around their joint centre-of-mass
- third bodies are confined to the plane of the circles

are not as well satisfied for the Earth-Moon system as for the Sun-Jupiter system, probably because of the much larger mass of Sun and Jupiter (3.3×10^5 and 318 times of the Earth respectively, much more for the Moon) as compared to the third body. The perfect planar circular motion is also not applicable to the system in reality. All these interfering factors make the linear approximation not a very valid model for the Earth-Moon system.

2 Programs

Note: Some programs listed on this pdf have $'return'$ added after excessively long texts for clarity, which needs to be removed before tested.

2.1 Question 1

```
function  Solveode(p,a,b,x0,y0,u0,v0,rt,at)
%Solve system for the motion of the third body numerically
%   e.g. Solveode(0.5,0,30,0.32,0,0,-1,1e-8,1e-10)
%         p=miu,a=t0,b=tfinal,
%         rt=relative tolerance, at=absolute tolerance.
syms t x1(t) x2(t) Y;
Dx1=diff(x1); Dx2=diff(x2); D2x1=diff(x1,2); D2x2=diff(x2,2);
Eq1=D2x1==2*Dx2+x1-p*(x1+1-p)/
((x1+1-p)^2+x2^2)^(3/2)-(1-p)*(x1-p)/((x1-p)^2+x2^2)^(3/2);
Eq2=D2x2==-2*Dx1+x2-p*x2/
((x1+1-p)^2+x2^2)^(3/2)-(1-p)*x2/((x1-p)^2+x2^2)^(3/2);
[VF]=odeToVectorField(Eq1,Eq2);
f=matlabFunction(VF,'vars',{t, Y});
options=odeset('RelTol', rt, 'AbsTol', at);
[T, Y]=ode45(@(t,Y) f(t,Y),[a,b],[y0 v0 x0 u0],options);
subplot(1,2,1)
plot(T,Y(:,1),T,Y(:,3),'g')
xlabel('time'), ylabel('x(t), y(t)')
legend('y(t) against t' , 'x(t) against t')
subplot(1,2,2)
plot(Y(:,3),Y(:,1),'g')
xlabel('x'), ylabel('y')
end
```

2.2 Question 2

```
function  SolveodeQ2(p,a,b,x0,y0,rt,at)
%UNTITLED2 Summary of this function goes here
%   e.g. SolveodeQ2(0.5,0,30,0.4,0,1e-8,1e-10) where radius=0.1
syms t x1(t) x2(t) Y;
Dx1=diff(x1); Dx2=diff(x2); D2x1=diff(x1,2); D2x2=diff(x2,2);
Eq1=D2x1==2*Dx2-p/((x1-p)^2+x2^2)^(3/2)*(x1-p);
```

```
Eq2=D2x2==-2*Dx1-p/((x1-p)^2+x2^2)^(3/2)*x2;
[VF]=odeToVectorField(Eq1,Eq2);
f=matlabFunction(VF,'vars',{t, Y});
options=odeset('RelTol', rt, 'AbsTol', at);
r0=((x0-p)^2+y0^2)^0.5;
u0=-y0*(-1+(r0/2+r0^4)^0.5/r0^2);
v0=(x0-p)*(-1+(r0/2+r0^4)^0.5/r0^2);
[T, Y]=ode45(@(t,Y) f(t,Y),[a,b],[y0 v0 x0 u0],options);
subplot(1,2,1)
plot(T,((Y(:,3)-p).^2+Y(:,1).^2).^0.5,'g')
xlabel('time'), ylabel('r(t)')
legend('r(t) against t')
subplot(1,2,2)
plot(Y(:,3),Y(:,1),'g')
xlabel('x'), ylabel('y')
end
```

2.3 Question 3

```
function  SolveodeQ3ver2(p,a,b,x0,y0,u0,v0,rt,at)
%UNTITLED2 Summary of this function goes here
%   %   e.g. SolveodeQ3ver2(0.5,0,30,0.32,0,0,-1,1e-8,1e-10)
%         p=miu,a=t0,b=tfinal,
%         rt=relative tolerance, at=absolute tolerance.
syms t x1(t) x2(t) Y;
Dx1=diff(x1); Dx2=diff(x2); D2x1=diff(x1,2); D2x2=diff(x2,2);
Eq1=D2x1==2*Dx2+x1-p*(x1+1-p)/
((x1+1-p)^2+x2^2)^(3/2)-(1-p)*(x1-p)/((x1-p)^2+x2^2)^(3/2);
Eq2=D2x2==-2*Dx1+x2-p*x2/
((x1+1-p)^2+x2^2)^(3/2)-(1-p)*x2/((x1-p)^2+x2^2)^(3/2);
[VF]=odeToVectorField(Eq1,Eq2);
f=matlabFunction(VF,'vars',{t, Y});
options=odeset('RelTol', rt, 'AbsTol', at);
[T, Y]=ode45(@(t,Y) f(t,Y),[a,b],[y0 v0 x0 u0],options);

x=linspace(min(Y(:,3))-3,max(Y(:,3))+3,1000);
y=linspace(min(Y(:,1))-3,max(Y(:,1))+3,1000);
[x,y]=meshgrid(x,y);
omega=-1./2*(x.^2+y.^2)-p./
```

```
((x+1-p).^2+y.^2).^(1/2)-(1-p)./((x-p).^2+y.^2).^(1/2);
J=1/2*(u0^2+v0^2)-1/2*(x0^2+y0^2)-p/((x0+1-p)^2+y0^2)^(1/2)-
(1-p)/((x0-p)^2+y0^2)^(1/2);
J0=zeros(1,length(T))+J;
J1=1/2*(Y(:,4).^2+Y(:,2).^2)-1/2*(Y(:,3).^2+Y(:,1).^2)-p./
((Y(:,3)+1-p).^2+Y(:,1).^2).^(1/2)-(1-p)./
((Y(:,3)-p).^2+Y(:,1).^2).^(1/2);
subplot(1,2,1)
contourf(x,y,omega>J(end),[1,1]);
hold on
plot(Y(:,3),Y(:,1),'g')
xlabel('x'), ylabel('y')
legend('Contour omega=J','y against x')
xlim([-2,2]);ylim([-2,2])
subplot(1,2,2)
plot(T,J0,T,J1,'g')
xlabel('t'), ylabel('J')
legend('J_{0}','J')
end
```

2.4 Question 4

2.4.1 SolveodeQ4ct(p,a,b,x0,y0,u0,v0,rt,at)

```
function  SolveodeQ4ct(p,a,b,x0,y0,u0,v0,rt,at)
%UNTITLED2 Summary of this function goes here
%   %   e.g. SolveodeQ4ct(0.5,0,30,0.32,0,0,-1,1e-8,1e-10)
%         p=miu, a=t0, b=tfinal,
%         rt=relative tolerance, at=absolute tolerance.
%         p=0.27712~0.27715, change of equl'm pts
syms t x1(t) x2(t) Y;
Dx1=diff(x1); Dx2=diff(x2); D2x1=diff(x1,2); D2x2=diff(x2,2);
Eq1=D2x1==2*Dx2+x1-p*(x1+1-p)/
((x1+1-p)^2+x2^2)^(3/2)-(1-p)*(x1-p)/((x1-p)^2+x2^2)^(3/2);
Eq2=D2x2==-2*Dx1+x2-p*x2/
((x1+1-p)^2+x2^2)^(3/2)-(1-p)*x2/((x1-p)^2+x2^2)^(3/2);
[VF]=odeToVectorField(Eq1,Eq2);
f=matlabFunction(VF,'vars',{t, Y});
options=odeset('RelTol', rt, 'AbsTol', at);
```

```
[T, Y]=ode45(@(t,Y) f(t,Y),[a,b],[y0 v0 x0 u0],options);
subplot(1,2,1)
plot(T,Y(:,3),T,Y(:,1),'g')
xlabel('t'), ylabel('x,y')
legend('x(t)','y(t)')

subplot(1,2,2)
x=linspace(min(Y(:,3))-3,max(Y(:,3))+3,1000);
y=linspace(min(Y(:,1))-3,max(Y(:,1))+3,1000);
[x,y]=meshgrid(x,y);
omega=-1./2*(x.^2+y.^2)-p./
((x+1-p).^2+y.^2).^(1/2)-(1-p)./((x-p).^2+y.^2).^(1/2);
J=1/2*(u0^2+v0^2)-1/2*(x0^2+y0^2)-p/((x0+1-p)^2+y0^2)^(1/2)-
(1-p)/((x0-p)^2+y0^2)^(1/2);
contour(x,y,omega,linspace(-2,-1,33));
colorbar
hold on

syms x1 x2
x=(p-1+p)/2;
[Xequ,parameter,conditon]=
solve([x-p*(x+1-p)/((x+1-p)^2+x2^2)^(3/2)-(1-p)*(x-p)/((x-p)^2+x2^2)^(3/2)
==0,x2-p*x2/((x+1-p)^2+x2^2)^(3/2)-(1-p)*x2/((x-p)^2+x2^2)^(3/2)==0],
x2,'ReturnConditions',true);
Xcol1=fzero(@(x1) x1-p*(x1+1-p)/((x1+1-p)^2+0^2)^(3/2)-
(1-p)*(x1-p)/((x1-p)^2+0^2)^(3/2),2);
Xcol2=fzero(@(x1) x1-p*(x1+1-p)/((x1+1-p)^2+0^2)^(3/2)-
(1-p)*(x1-p)/((x1-p)^2+0^2)^(3/2),x);
Xcol3=fzero(@(x1) x1-p*(x1+1-p)/((x1+1-p)^2+0^2)^(3/2)-
(1-p)*(x1-p)/((x1-p)^2+0^2)^(3/2),-2);
y=nonzeros(Xequ);
Xsol=[x x Xcol1 Xcol2 Xcol3],Ysol=[y(1) y(2) 0 0 0],
plot(Y(:,3),Y(:,1),'g',Xsol,Ysol,'*')
xlabel('x'), ylabel('y')
legend('Contour omega','y against x','equilibrum pts')
xlim([-2,2]);ylim([-2,2]);
end
```

2.4.2 SolveodeQ4(p,a,b,x0,y0,u0,v0,rt,at)

```
function  SolveodeQ4(p,a,b,x0,y0,u0,v0,rt,at)
%UNTITLED2 Summary of this function goes here
%   %   e.g. SolveodeQ4(0.5,0,30,0.32,0,0,-1,1e-8,1e-10)
%          p=miu,a=t0,b=tfinal,
%          rt=relative tolerance, at=absolute tolerance.
%          p=0.27712~0.27715, change of equl'm pts
syms t x1(t) x2(t) Y;
Dx1=diff(x1); Dx2=diff(x2); D2x1=diff(x1,2); D2x2=diff(x2,2);
Eq1=D2x1==2*Dx2+x1-p*(x1+1-p)/
((x1+1-p)^2+x2^2)^(3/2)-(1-p)*(x1-p)/((x1-p)^2+x2^2)^(3/2);
Eq2=D2x2==-2*Dx1+x2-p*x2/
((x1+1-p)^2+x2^2)^(3/2)-(1-p)*x2/((x1-p)^2+x2^2)^(3/2);
[VF]=odeToVectorField(Eq1,Eq2);
f=matlabFunction(VF,'vars',{t, Y});
options=odeset('RelTol', rt, 'AbsTol', at);
[T, Y]=ode45(@(t,Y) f(t,Y),[a,b],[y0 v0 x0 u0],options);
subplot(1,2,1)
plot(T,Y(:,3),T,Y(:,1),'g')
xlabel('t'), ylabel('x,y')
legend('x(t)','y(t)')

subplot(1,2,2)
x=linspace(min(Y(:,3))-3,max(Y(:,3))+3,1000);
y=linspace(min(Y(:,1))-3,max(Y(:,1))+3,1000);
[x,y]=meshgrid(x,y);
omega=-1./2*(x.^2+y.^2)-p./((x+1-p).^2+y.^2).^(1/2)-(1-p)./
((x-p).^2+y.^2).^(1/2);
J=1/2*(u0^2+v0^2)-1/2*(x0^2+y0^2)-p/((x0+1-p)^2+y0^2)^(1/2)-
(1-p)/((x0-p)^2+y0^2)^(1/2);
contourf(x,y,omega>J(end),[1,1]);
hold on

syms x1 x2
x=(p-1+p)/2;
[Xequ,parameter,conditon]=
solve([x-p*(x+1-p)/((x+1-p)^2+x2^2)^(3/2)-(1-p)*(x-p)/((x-p)^2+x2^2)^(3/2)
```

```
==0,x2-p*x2/((x+1-p)^2+x2^2)^(3/2)-(1-p)*x2/((x-p)^2+x2^2)^(3/2)==0],
x2,'ReturnConditions',true);
Xcol1=fzero(@(x1) x1-p*(x1+1-p)/((x1+1-p)^2+0^2)^(3/2)-
(1-p)*(x1-p)/((x1-p)^2+0^2)^(3/2),2);
Xcol2=fzero(@(x1) x1-p*(x1+1-p)/((x1+1-p)^2+0^2)^(3/2)-
(1-p)*(x1-p)/((x1-p)^2+0^2)^(3/2),x);
Xcol3=fzero(@(x1) x1-p*(x1+1-p)/((x1+1-p)^2+0^2)^(3/2)-
(1-p)*(x1-p)/((x1-p)^2+0^2)^(3/2),-2);
y=nonzeros(Xequ);
Xsol=[x x Xcol1 Xcol2 Xcol3],Ysol=[y(1) y(2) 0 0 0],
plot(Y(:,3),Y(:,1),'g',Xsol,Ysol,'*')
xlabel('x'), ylabel('y')
legend('Contour omega=J','y against x','equilibrum pts')
xlim([-2,2]);ylim([-2,2]);
end
```

2.4.3 SolveodeQ4test(p,a,b,x0,y0,u0,v0,rt,at)

```
function SolveodeQ4test(p,a,b,x0,y0,u0,v0,rt,at)
%UNTITLED2 Summary of this function goes here
%   %   e.g. SolveodeQ4test(0.5,0,30,0.32,0,0,-1,1e-8,1e-10)
%          p=miu,a=t0,b=tfinal,
%          rt=relative tolerance, at=absolute tolerance.
%          p=0.27712~0.27715, change of equl'm pts
syms t x1(t) x2(t) Y;
Dx1=diff(x1); Dx2=diff(x2); D2x1=diff(x1,2); D2x2=diff(x2,2);
Eq1=D2x1==2*Dx2+x1-p*(x1+1-p)/
((x1+1-p)^2+x2^2)^(3/2)-(1-p)*(x1-p)/((x1-p)^2+x2^2)^(3/2);
Eq2=D2x2==-2*Dx1+x2-p*x2/
((x1+1-p)^2+x2^2)^(3/2)-(1-p)*x2/((x1-p)^2+x2^2)^(3/2);
[VF]=odeToVectorField(Eq1,Eq2);
f=matlabFunction(VF,'vars',{t, Y});
options=odeset('RelTol', rt, 'AbsTol', at);
[T, Y]=ode45(@(t,Y) f(t,Y),[a,b],[y0 v0 x0 u0],options);

subplot(1,2,1)
M=log(Y(:,3)-1.1806);
plot(T,M)
xlabel('t'), ylabel('log(\delta x)')
```

```
subplot(1,2,2)
x=linspace(min(Y(:,3))-3,max(Y(:,3))+3,1000);
y=linspace(min(Y(:,1))-3,max(Y(:,1))+3,1000);
[x,y]=meshgrid(x,y);
omega=-1./2*(x.^2+y.^2)-p./((x+1-p).^2+y.^2).^(1/2)-(1-p)./
((x-p).^2+y.^2).^(1/2);
J=1/2*(u0^2+v0^2)-1/2*(x0^2+y0^2)-p/((x0+1-p)^2+y0^2)^(1/2)-
(1-p)/((x0-p)^2+y0^2)^(1/2);
contour(x,y,omega,linspace(-6,-1,33));
colorbar
hold on

syms x1 x2
x=(p-1+p)/2;
[Xequ,parameter,conditon]=
solve([x-p*(x+1-p)/((x+1-p)^2+x2^2)^(3/2)-(1-p)*(x-p)/((x-p)^2+x2^2)^(3/2)
==0,x2-p*x2/((x+1-p)^2+x2^2)^(3/2)-(1-p)*x2/((x-p)^2+x2^2)^(3/2)==0],
x2,'ReturnConditions',true);
Xcol1=fzero(@(x1) x1-p*(x1+1-p)/((x1+1-p)^2+0^2)^(3/2)-
(1-p)*(x1-p)/((x1-p)^2+0^2)^(3/2),2);
Xcol2=fzero(@(x1) x1-p*(x1+1-p)/((x1+1-p)^2+0^2)^(3/2)-
(1-p)*(x1-p)/((x1-p)^2+0^2)^(3/2),x);
Xcol3=fzero(@(x1) x1-p*(x1+1-p)/((x1+1-p)^2+0^2)^(3/2)-
(1-p)*(x1-p)/((x1-p)^2+0^2)^(3/2),-2);
y=nonzeros(Xequ);
Xsol=[x x Xcol1 Xcol2 Xcol3],Ysol=[y(1) y(2) 0 0 0],
plot(Y(:,3),Y(:,1),'g',Xsol,Ysol,'*')
P=floor(length(T)/(max(T)-min(T))*2)
plot(Y(P,3),Y(P,1),'o')
xlabel('x'), ylabel('y')
legend('Contour omega=J','y against x','equilibrum pts')
xlim([-2,2]);ylim([-2,2]);


end
```