Assignment 8a–Computation of the equilibrium points of the RTBP

Yixie Shao

April 28, 2015

1 Restricted Three-Body Problem

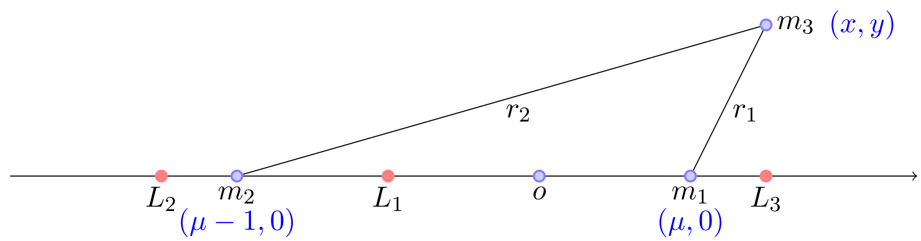

Figure 1: Restricted Three-Body Problem

The equations of motion are:

$$\begin{cases} x'' - 2y' &= \Omega_x \\ y'' + 2x' &= \Omega_y \end{cases} \tag{1}$$

And,

$$\Omega(x,y) = \frac{(x^2+y^2)}{2} + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} + \frac{\mu(1-\mu)}{2} \tag{2}$$

Here:

$$\begin{cases} \mu &= \frac{m_2}{m_1+m_2} \\ r_1 &= \sqrt{(x-\mu)^2+y^2} \\ r_2 &= \sqrt{(x-\mu+1)^2+y^2} \end{cases} \tag{3}$$

Let:

$$\begin{cases} x_1 &= x \\ x_2 &= y \\ x_3 &= x' \\ x_4 &= y' \end{cases} \tag{4}$$

The RTBP is expressed as:

$$\begin{cases} f_1 &= x_1' = x_3 \\ f_2 &= x_2' = x_4 \\ f_3 &= x_3' = 2x_4 + \Omega_{x_1} \\ f_4 &= x_4' = -2x_3 + \Omega_{x_2} \end{cases} \tag{5}$$

Where,

$$\begin{cases} \Omega_{x_1} &= x_1 - \frac{(1-\mu)(x_1-\mu)}{{r_1}^3} - \frac{\mu(x_1-\mu+1)}{{r_2}^3} \\ \Omega_{x_2} &= x_2(1 - \frac{1-\mu}{{r_1}^3} - \frac{\mu}{{r_2}^3}) \end{cases} \tag{6}$$

2 Equilibrium points—$L_1,\ L_2,\ L_3$

The position of L_1 is:

$$x_{L_1} = \mu - 1 + \xi \tag{7}$$

where,

$$f(\xi) = (\frac{\mu(1-\xi)^2}{3-2\mu-\xi(3-\mu-\xi)})^{\frac{1}{3}}$$

$$\xi_0 = (\frac{\mu}{3(1-\mu)})^{\frac{1}{3}}$$

$$\xi_{n+1} = f(\xi_n)$$

The position of L_2 is:

$$x_{L_2} = \mu - 1 - \xi \tag{8}$$

where,

$$f(\xi) = (\frac{\mu(1+\xi)^2}{3-2\mu+\xi(3-\mu+\xi)})^{\frac{1}{3}}$$

$$\xi_0 = (\frac{\mu}{3(1-\mu)})^{\frac{1}{3}}$$

$$\xi_{n+1} = f(\xi_n)$$

The position of L_3 is:

$$x_{L_3} = \mu + \xi \tag{9}$$

where,

$$f(\xi) = (\frac{(1-\mu)(1+\xi)^2}{1+2\mu+\xi(2+\mu+\xi)})^{\frac{1}{3}}$$

$$\xi_0 = 1 - \frac{7}{12}\mu$$

$$\xi_{n+1} = f(\xi_n)$$

3 Simplified Jacobi constant

Jacobi integral C is:

$$C = 2\Omega(x,y) - ({x'}^2 + {y'}^2) \tag{10}$$

Here is simplified as:

$$C = 2\Omega(x,0) = x^2 + \frac{2(1-\mu)}{|x-\mu|} + \frac{2\mu}{|x-\mu+1|} + \mu(1-\mu) \tag{11}$$

4 Results

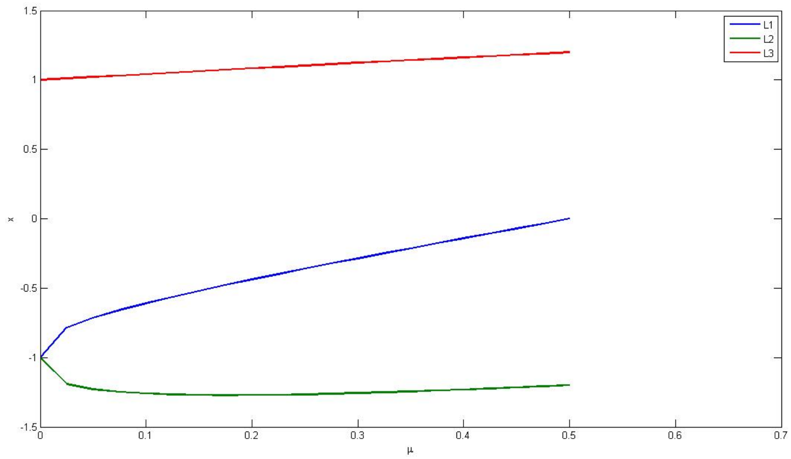

Figure 2: μ V.s. x

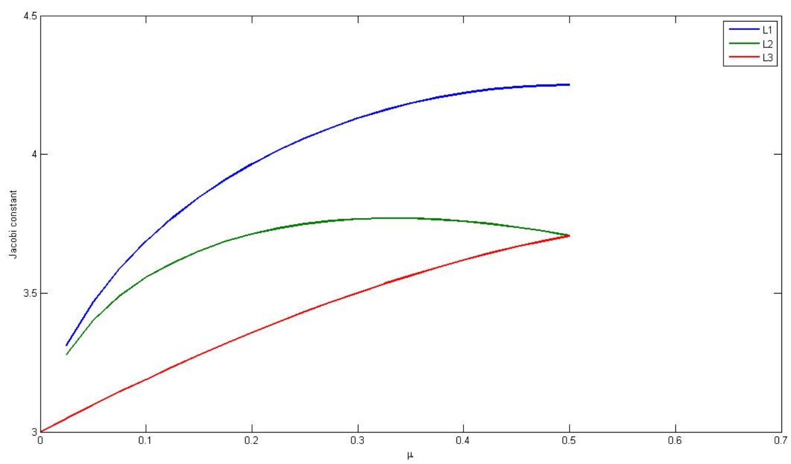

Figure 3: μ V.s. Jacobi constant

5 Code

Main function:

```
      implicit real*8 (a-h,o-z)
      parameter (n=3)
```

```
      common/param/xmu
      dimension xl(n),cl(n)
      open(10,file='orbit.d',status='unknown')
      np=20
      xstep=0.5/np
      xmu=0.d0
      do j=1,(np+1)
         call peq(xmu,xl1,xl2,xl3,cl1,cl2,cl3)
         xl(1)=xl1
         xl(2)=xl2
         xl(3)=xl3
         cl(1)=cl1
         cl(2)=cl2
         cl(3)=cl3
         write(10,*)xmu,(xl(i),i=1,n),(cl(i),i=1,n)
         xmu=xmu+xstep
      enddo
      end
```

Sub-function–JacobiSimp:

```
       subroutine jacobiSimp(xmu,x,Cja)
       IMPLICIT REAL*8 (A-H,O-Z)
       r1=sqrt((x-xmu)**2)
       r2=sqrt((x-xmu+1)**2)
       omiga=0.5*x**2+(1-xmu)/r1+xmu/r2+0.5*(1-xmu)*xmu
       Cja=2*omiga
       return
       end
```

Sub-function–calc_eqpoints:

```
       subroutine peq(xmu,xl1,xl2,xl3,cl1,cl2,cl3)
       implicit real*8(a-h,o-z)

       a=1.d0/3.d0
       i=0
c to compute L2 (on the left hand side of the small primary)
       x=xmu/(3.d0*(1.d0-xmu))
       x=x**a
 1        den=3.d0-2.d0*xmu+x*(3.d0-xmu+x)
       f=xmu*(1.d0+x)**2/den
```

```
      f=f**a
      x1=xmu-1.d0-x
      if (dabs(x-f).le.1.d-15)then
       xl2=x1
           call jacobiSimp(xmu,xl2,cl2)
           go to 3
      endif
      i=i+1
      x=f
      go to 1
 2    format(e25.16,',',e25.16,',',e25.16)
 3    continue


      i=0
      x=xmu/(3.d0*(1.d0-xmu))
      x=x**a
 10       den=3.d0-2.d0*xmu-x*(3.d0-xmu-x)
      f=xmu*(1.d0-x)**2/den
      f=f**a
      x1=xmu-1.d0+x
      if (dabs(x-f).le.1.d-15)then
       xl1=x1
           call jacobiSimp(xmu,xl1,cl1)
           go to 4
      endif
      i=i+1
      x=f
      go to 10
 4    continue
c
c L3 (on the right hand side of the big primary)
      i=0
      x=1.d0-7.d0*xmu/12.d0
 5         den=1.d0+2.d0*xmu+x*(2.d0+xmu+x)
      f=(1.d0-xmu)*(1.d0+x)**2/den
      f=f**a
      x1=xmu+x
      if (dabs(x-f).le.1.d-15)then
      xl3=x1
           call jacobiSimp(xmu,xl3,cl3)
           go to 6
      endif
      i=i+1
      x=f
      go to 5
 6    return
      end
```