Assignment 9–Computation of the equilibrium points of the circular RTBP and associated invariant manifolds

Yixie Shao

May 6, 2015

1 Restricted Three-Body Problem

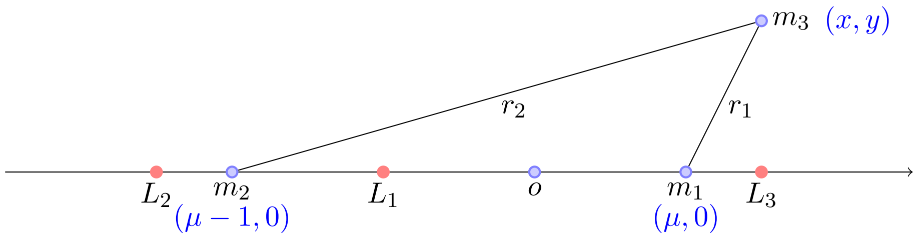

Figure 1: Restricted Three-Body Problem

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 test

The unstable manifold of L_3, with initial condition: $L_3 - 10^{-6} \cdot \overrightarrow{v}$ and $\mu = 0.008$.

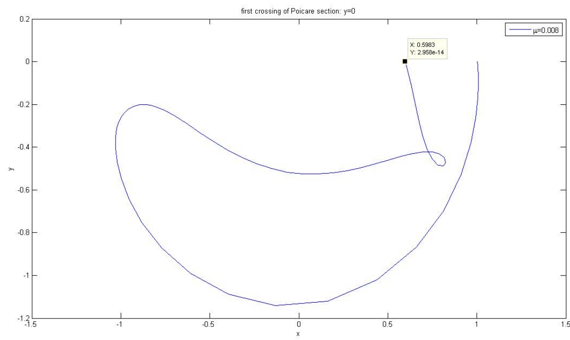

Figure 2: $\mu = 0.008, 1st$-crossing

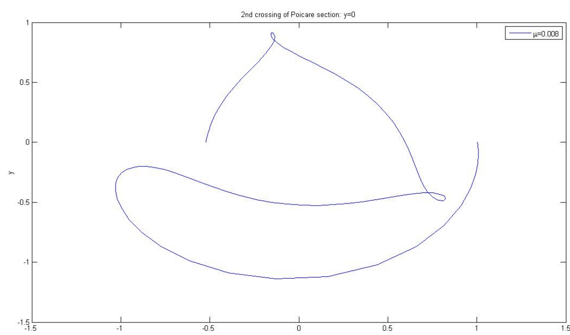

Figure 3: $\mu = 0.008, 2nd$-crossing

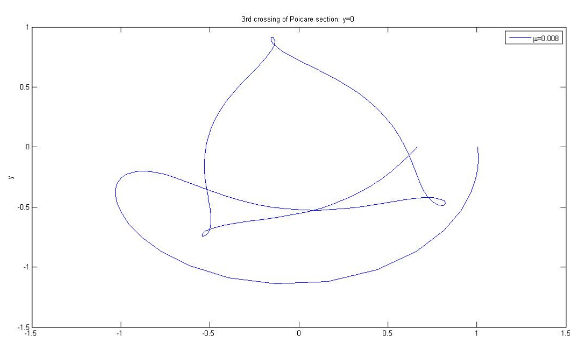

Figure 4: $\mu = 0.008, 3rd$-crossing

5 result

$\mu = 0.1$, consider L_3 L_3: (1.0416089085710609,0,0,0)
$C(L_3)$: 3.1895781504493814
The Jacobian matrix A is:

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \Omega_{x_1x_1} & \Omega_{x_1x_2} & 0 & 2 \\ \Omega_{x_1x_2} & \Omega_{x_2x_2} & -2 & 0 \end{pmatrix} \tag{12}$$

In this situation, A is:

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3.1833839612695192 & 0 & 0 & 2 \\ 0 & -9.1691980634759820^{-2} & -2 & 0 \end{pmatrix}$$

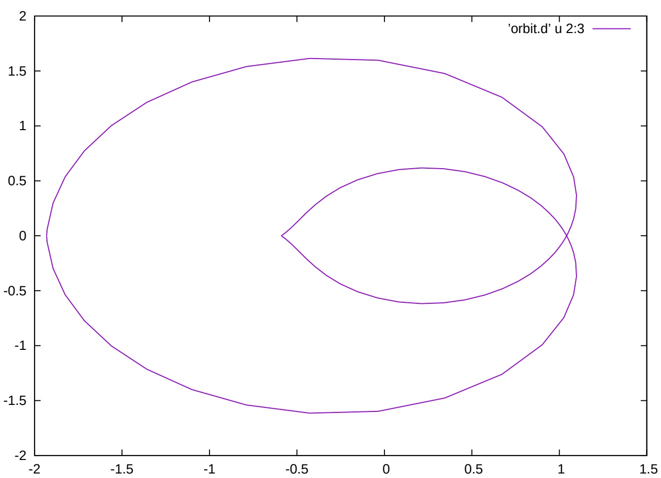

Figure 5: $\mu = 0.1, 1st$-crossing

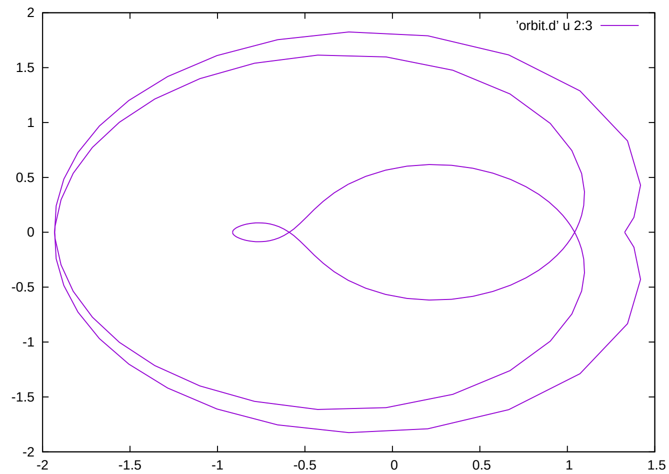

Figure 6: $\mu = 0.1, 2nd$-crossing

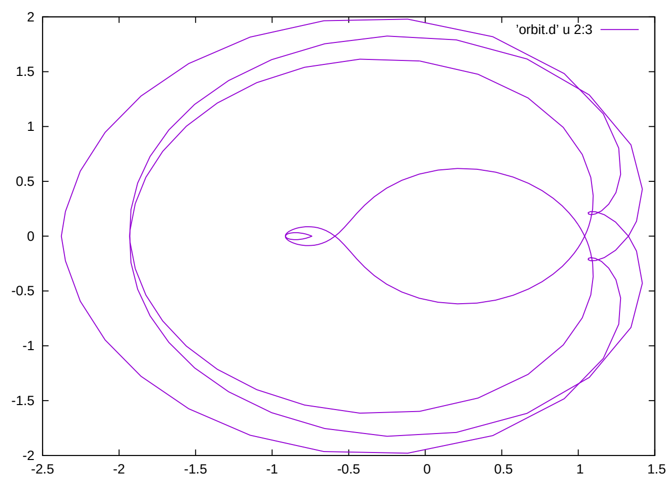

Figure 7: $\mu = 0.1, 3rd$-crossing

6 Code

Main function:

```
      implicit real*8 (a-h,o-z)
      parameter (n=4)
      common/param/xmu
      dimension x(n),oa(n,n),rr(n),ri(n),vr(n,n),vi(n,n)
```

```fortran
      open(10,file='orbit.d',status='unknown')

      write(*,*) 'mu'
      read(*,*)xmu
      call peq(xmu,xl1,xl2,xl3,cl1,cl2,cl3)
      write(*,*)'xL3:', xl3
      write(*,*)'cL3:', cl3
      call jacobiA(xmu,xl3,n,oa)
      write(*,*)'A:'
      write(*,*)(oa(1,i),i=1,n)
      write(*,*)(oa(2,i),i=1,n)
      write(*,*)(oa(3,i),i=1,n)
      write(*,*)(oa(4,i),i=1,n)
      call vapvep(oa,n,rr,ri,vr,vi)
      ti=0
      T=0.3138977039438897d01
      tmax=20.d0*T
      np=10
      p=1
      do k=1,2
         p=-1*p
         write(*,*)'p:', p
         do j=3,4
            x(1)=xl3
            x(2)=0.d0
            x(3)=0.d0
            x(4)=0.d0
            idir=dsign(1.d0,rr(j))
            write(*,*)'idir:', idir
            x(1)=p*vr(1,j)*1.d-6+x(1)
            x(2)=p*vr(2,j)*1.d-6+x(2)
            x(3)=p*vr(3,j)*1.d-6+x(3)
            x(4)=p*vr(4,j)*1.d-6+x(4)
            write(*,*)'Initial t:'
            write(*,*)ti
            write(*,*)'Initial cond:'
            write(*,*)(x(ii),ii=1,n)
            write(*,*) 'm times crossing'
            read(*,*) m
            do i=1,m
               call poinc1(n,x,tfinal,idir)
               t=tfinal+t
               write(*,*)'t:'
               write(*,*)t
            enddo
            write(10,90)
90          format()
         enddo
      enddo
```

```
      end


C
 ****************************************************************************

C
C
 ****************************************************************************



      SUBROUTINE POINC1(n,YI,tfinal,idirorig)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION YI(n),YF(n),DGG(n),F(n)
      icont=0
      idir=idirorig
      ti=0.d0
C   DETERMINATION OF THE FIRST PASSAGE OF THE ORBIT THROUGH
 y=0
      CALL SECCIO(YI,GG,DGG)
      IF(DABS(GG).LT.1.D-9)GG=0.d0
      GA=GG
      hab=.1e-16
      hre=.1e-16
      pabs=dlog10(hab)
      prel=dlog10(hre)
      istep=1
      pas=5d0
      ht=0.d0
      t=ti
1      tmax=t+idir*pas
      CALL taylor_f77_eq_rtbp_(t,yi,idir,istep,pabs,prel,
     &  tmax,ht,iordre,ifl)
      CALL SECCIO(YI,GG,DGG)
      IF(GG*GA.LT.0.D0)go to 22
      write(10,*)t,(yi(ii),ii=1,4)
      GA=GG
      GO TO 1
C
C   REFINEMENT OF THE INTERSECTION POINT YF(*) USING NEWTON
 METHOD
C   TO GET A ZERO OF THE FUNCTION GG (SEE SUBROUTINE SECCIO)
C
22    continue
      icont=icont+1
      if (icont.gt.20)then
          write(*,*)'problems finding the section'
          stop
```

```
      endif
      CALL FIELD(T,YI,N,F)
      P=0.D0
      DO 3 I=1,N
3     P=P+F(I)*DGG(I)
      H=-GG/P
      if (h.ge.0.d0)idir=1
      if (h.lt.0.d0)idir=-1
      tmax=t+h

      CALL taylor_f77_eq_rtbp_(t,yi,idir,istep,pabs,prel,
     &  tmax,ht,iordre,ifl)
      CALL SECCIO(YI,GG,DGG)
      IF(DABS(GG).GT.1.D-13)GO TO 22
      DO 4 I=1,N
4     YF(I)=YI(I)
      tfinal=t
      write(*,*)'YI:', (yi(ii),ii=1,4)
      write(*,*)'YF:', (yf(ii),ii=1,4)
      write(10,*)t,(yf(ii),ii=1,4)
      return
      end


C
   ****************************************************************************
C


   *
C     THE SURFACE g OF SECTION,IN THIS CASE
C          INPUT PARAMETERS:
C     Y(*)       POINT
C          OUTPUT PARAMETERS:
C     GG          FUNCTION THAT EQUATED TO 0 GIVES THE
   SURFACE OF
C                 SECTION
C     DGG(*)      GRADIENT OF FUNCTION GG
C

   *
C
   ****************************************************************************

      SUBROUTINE SECCIO(Y,GG,DGG)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Y(4),DGG(4)
      GG=Y(2)
      DO 1 I=1,4
```

```
 1      DGG(I)=0.D0
        DGG(2)=1.d0
        RETURN
        END


C
C
   ****************************************************************************

C
C     EQS OF MOTION IN synodical VARIABLES
C     X           TIME
C     Y(*)        POINT (Y(1),Y(2),....Y(n))
C     NEQ         NUMBER OF EQUATIONS
C          OUTPUT PARAMETERS:
C     F(*)        VECTOR FIELD
C
C
   ****************************************************************************

      subroutine field(t,x,neq,f)
      implicit real*8 (a-h,o-z)
      common/param/xmu
      dimension x(neq),f(neq)
      umu=1.-xmu
      d1=x1-xmu
      d2=x1+umu
      r12=d1*d1+x2*x2
      r22=d2*d2+x2*x2
      r0=sqrt(r12)
      r1=sqrt(r22)
      r032=r12*r0
      r132=r22*r1
      omex=x1-(umu*(-xmu+x1)/r032)-(xmu*(x1+umu)/r132)
      omey=x2*(1.-(umu/r032)-(xmu/r132))
      f(1)=x(3)
      f(2)=x(4)
      f(3)=2*x(4)+omex
      f(4)=-2*x(3)+omey
      return
      end

C
   ****************************************************************************

C
C
   ****************************************************************************
```

```fortran
      subroutine jacobiA(xmu,x,n,oa)
       implicit real*8(a-h,o-z)
       dimension oa(n,n)
       umu=1.d0-xmu
       x1=x
       xd1=dabs(x1-xmu)
       xd2=dabs(x1+umu)
       r032=xd1**3
       r132=xd2**3
       r052=xd1**5
       r152=xd2**5
       xa=3*umu*(-xmu+x1)/r052
       xb=3*xmu*(x1+umu)/r152
       omexx=1.d0-umu/r032+xa*(-xmu+x1)-xmu/r132+xb*(x1+umu
     )
       omeyy=1.d0-umu/r032-xmu/r132
       oa(1,1)=0.d0
       oa(1,2)=0.d0
       oa(1,3)=1.d0
       oa(1,4)=0.d0
       oa(2,1)=0.d0
       oa(2,2)=0.d0
       oa(2,3)=0.d0
       oa(2,4)=1.d0
       oa(3,1)=omexx
       oa(3,2)=0.d0
       oa(3,3)=0.d0
       oa(3,4)=2.d0
       oa(4,1)=0.d0
       oa(4,2)=omeyy
       oa(4,3)=-2.d0
       oa(4,4)=0.d0
       return
       end
```