Assignment 11–Computation of periodic orbits for the RTBP

Yixie Shao

May 16, 2015

1 Restricted Three-Body Problem

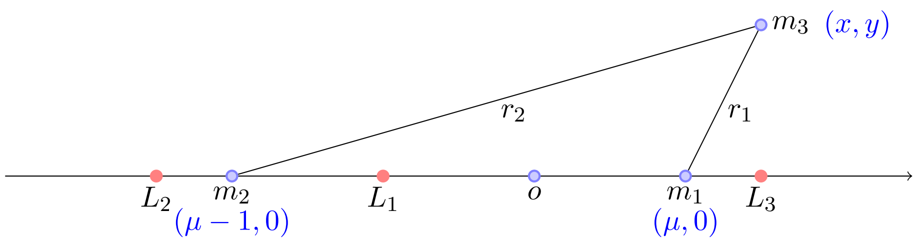

Figure 1: Restricted Three-Body Problem

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 result

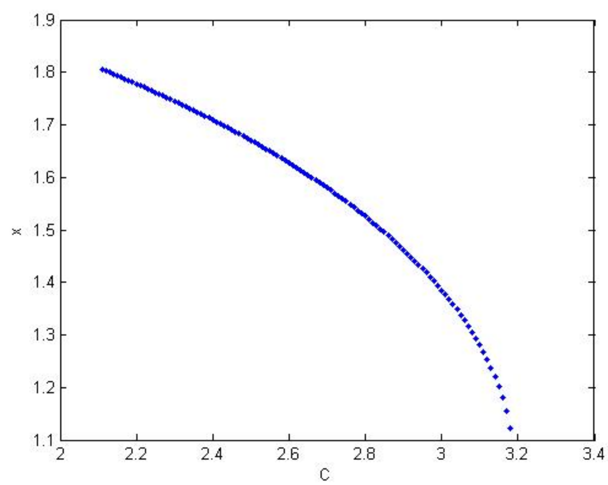

Figure 2: the curve(C,x)

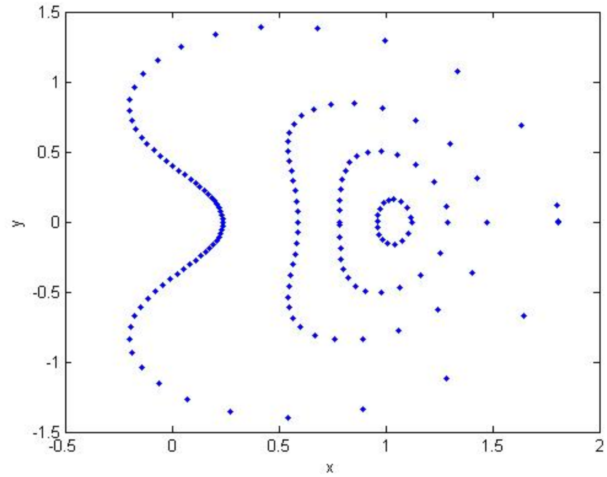

Figure 3: some orbit

4 Code

Main function to get the curve (C,v)

```
          implicit real*8 (a-h,o-z)
          parameter (n=4)
          common/param/xmu
          dimension x(n)
          open(10,file='orbit.d',status='unknown')
          open(11,file='cVsx.d',status='unknown')
          ti=0
          T=0.3138977039438897d01
          tmax=20.d0*T
          np=10
          write(*,*)'mu:'
          read(*,*)xmu

         call peq(xmu,xl1,xl2,xl3,cl1,cl2,cl3)
         write(*,*)'Jacobi integral', cl3
         xincC=1.d-2
         xincX=1.d-3
         cmin=2.1
         istep=(cl3-cmin)/xincC
         write(*,*)istep
         xc=xl3
         do i=1,istep
```

```
            const=cl3-i*xincC
            x0=xc+1.d-5
            call getYPrim(const,xmu,x0,yPrim)
            x(1)=x0
            x(2)=0.d0
            x(3)=0.d0
            x(4)=yPrim
            call getXPrim(xmu,x,xPrim)
            do j=1,10000
               xPrim0=xPrim
               xOld=x0+(j-1)*xincX
               xNew=x0+j*xincX
               call getYPrim(const,xmu,xNew,yPrim)
               x(1)=xNew
               x(2)=0.d0
               x(3)=0.d0
               x(4)=yPrim
               write(10,*)ti,(x(ii),ii=1,4)
               call getXPrim(xmu,x,xPrim)
               xPrimN=xPrim
               xM=xPrim0*xPrimN
c               write(11,*)x(1),xPrim,yPrim
               if (xM.lt.0.d0) then
c               write(*,*)'old x:', xOld
c               write(*,*)'old xPrim:',xPrim0
c               write(*,*)'current x:',x(1)
c               write(*,*)'current xPrim:',xPrimN
               call bisection(xmu,const,xOld,xNew,xc,fc)
               write(11,*)const,xc,fc
               exit
               endif
            enddo
         enddo
         end
```

Main function to get the orbits for some special Jacobi constant

```
         implicit real*8 (a-h,o-z)
         parameter (n=4)
         common/param/xmu
         dimension x(n),const(n),xc(n)
         open(10,file='orbit.d',status='unknown')
         ti=0
         T=0.3138977039438897d01
         tmax=20.d0*T
         np=10
         xmu=0.1
         idir=1
         const(1)=3.1795781504493816
```

```
        const(2)=3.0895781504493813
        const(3)=2.8795781504493814
        const(4)=2.1095781504493814
        xc(1)=1.1220670842309302
        xc(2)=1.2930576026436325
        xc(3)=1.4760758462438210
        xc(4)=1.8055710411497072
        do i=1,n
            x0=xc(i)
            con=const(i)
            write(*,*)'x0:',x0
            write(*,*)'constant:',con
            call getYPrim(con,xmu,x0,yPrim)
            x(1)=x0
            x(2)=0.d0
            x(3)=0.d0
            x(4)=yPrim
            write(10,*)ti,(x(ii),ii=1,4)
            do j=1,2
            call poinc1(n,x,tfinal,idir)
            enddo
            write(10,90)
90          format()
        enddo
        end
```

Sub-function: get y'

```
        subroutine getYprim(const,xmu,x,yPrim)
        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
        yPrim=-sqrt(2*omiga-const)
        return
        end
```

Sub-function: bisection

```
        subroutine bisection(xmu,const,xa,xb,xc,fc)
        implicit real*8 (a-h,o-z)
        parameter (n=3)
        dimension xInter(n),f(n),xi(n),xm(2),x(4)
        xInter(1)=xa
        xInter(2)=(xa+xb)/2
        xInter(3)=xb
```

```
c          write(*,*)(xInter(ii),ii=1,3)
        do k=1,50
           icont=k
           do i=1,n
              call getYPrim(const,xmu,xInter(i),yPrim)
              x(1)=xInter(i)
              x(2)=0.d0
              x(3)=0.d0
              x(4)=yPrim
              call getXPrim(xmu,x,xPrim)
              f(i)=xPrim
           enddo
           if (dabs(f(2)).lt.1.d-12) then
               xc=xInter(2)
               fc=f(2)
c               write(*,*)'c:',xc
c               write(*,*)'f(c):',fc
c               write(*,*)'icont:',icont
               return
           endif
           do i=1,2
              xm(i)=f(i)*f(i+1)
           enddo
           if (xm(1).lt.0.d0) xInter(3)=xInter(2)
           if (xm(2).lt.0.d0) xInter(1)=xInter(2)
           xInter(2)=(xInter(1)+xInter(3))/2
c           write(*,*)(xInter(ii),ii=1,3)
c           write(*,*)(f(ii),ii=1,3)
        enddo
        if (icont.eq.50) then
        write(*,*)'Problem in finding f(c)=0'
        xc=xInter(2)
        fc=f(2)
        write(*,*)'c:',xc
        write(*,*)'f(c):',fc
        return
        endif

        end
```

Sub-function: get x'

```

        subroutine getXPrim(xmu,x,xPrim)
        IMPLICIT REAL*8 (A-H,O-Z)
        parameter (n=4)
        dimension x(n)
        idir=1
        call poinc1(n,x,tfinal,idir)
```

```
      t=tfinal+t
      xPrim=x(3)
      return
      end
```

Sub-function: poinc1

```

      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'
         return
      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(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
```