Assignment 12–Computation of invariant manifolds of periodic orbits

Yixie Shao

May 20, 2015

1 Restricted Three-Body Problem

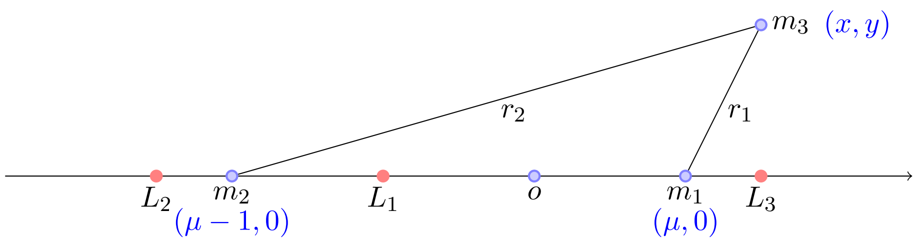

Figure 1: Restricted Three-Body Problem

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 result

Three initial points are taken.

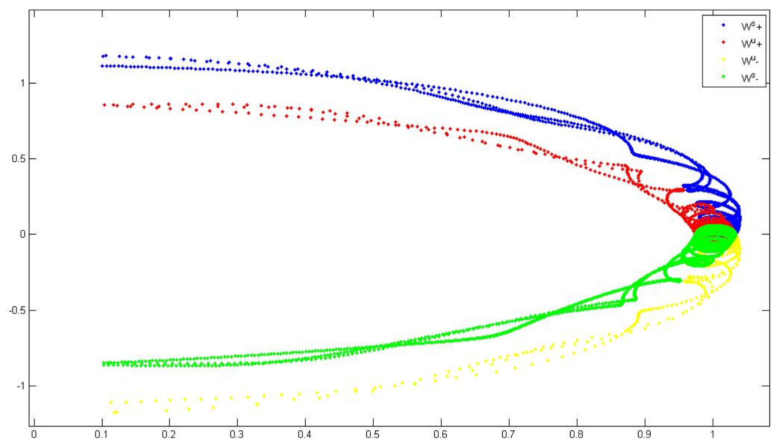

Figure 2: the whole orbits

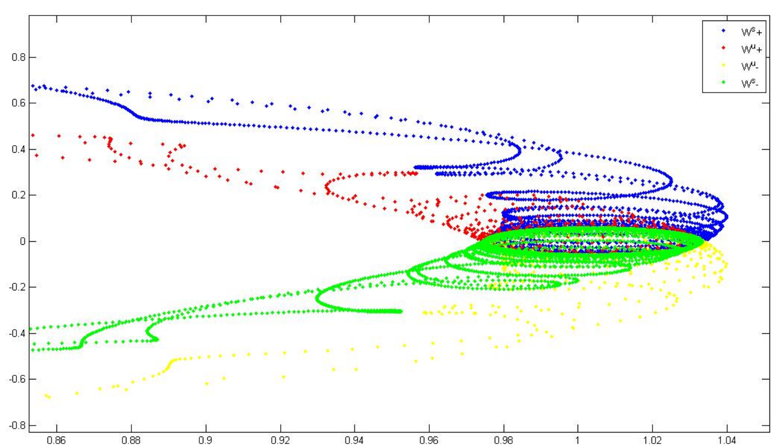

Figure 3: the zoomed in orbits

3 Code

Main function

```
implicit real*8 (a-h,o-z)
parameter (n=20,m=4)
common/param/xmu
dimension x(n),xi(n),oM(m,m),rr(m),ri(m),vr(m,m),vi(
   m,m)
open(10,file='orbit1.d',status='unknown')
ti=0
Tmid=0.3138977039438897d01
tmax=2.d0*Tmid
tfinal=2.d0*Tmid
np=10
xmu=1.d-2
xi(1)=1.033366313746765d0
xi(2)=0.d0
xi(3)=0.d0
xi(4)=-.05849376854515592d0
xi(5)=1
xi(6)=0
xi(7)=0
xi(8)=0
xi(9)=0
xi(10)=1
xi(11)=0
xi(12)=0
xi(13)=0
xi(14)=0
xi(15)=1
xi(16)=0
xi(17)=0
xi(18)=0
xi(19)=0
xi(20)=1
do i=1,n
   x(i)=xi(i)
enddo
if (tmax.ge.ti)then
    idir=1
else
    idir=-1
endif
xinctime=dabs(tmax-ti)/np
do i=1,np
   call flow(ti,n,x,idir,xinctime)
   do k=1,4
```

```
              do j=1,4
                 oM(k,j)=x(4*k+j)
              enddo
           enddo
        enddo
        call vapvep(oM,m,rr,ri,vr,vi)
         idirorig=-1
         p=1
         do k=1,2
           p=-1*p
           do j=1,2
               idirorig=-1*idirorig
               do i=1,3
               x(1)=(2*i-1)*p*vr(1,j)*1.d-6+xi(1)
               x(2)=(2*i-1)*p*vr(2,j)*1.d-6+xi(2)
               x(3)=(2*i-1)*p*vr(3,j)*1.d-6+xi(3)
               x(4)=(2*i-1)*p*vr(4,j)*1.d-6+xi(4)
               call poinc1(m,x,tfinal,idirorig)
               write(10,90)
 90            format()
               write(*,*)i
               enddo
           enddo
         enddo

        end


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

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

       subroutine flow(t,n,x,idir,xinctemps)
       IMPLICIT REAL*8 (A-H,O-Z)
       dimension x(n)
       tmax=t+idir*xinctemps
c
c parameters for the integration
c
        hab=0.1e-16
        hre=0.1e-16
        pabs=dlog10(hab)
        prel=dlog10(hre)
c Option of control of step
        istep=1
        ht=0.d0
```

```
 1         CALL taylor_f77_eq_rtbp_var_(t,x,idir,istep,pabs,
     prel,
      &  tmax,ht,iordre,ifl)
c          write(10,100) t,(x(i),i=1,n)
         if (idir.eq.1.and.t.lt.tmax)go to 1
         if (idir.eq.-1.and.t.gt.tmax)go to 1
c check t=tmax
         if (dabs(t-tmax).le.1.d-13)return
         write(*,*)'problems in taylor'
         stop
c 100      format(f15.8,2f22.15)
         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=.4d0
         ht=0.d0
         t=ti
1         tmax=t+idir*pas
         CALL taylor_f77_eq_rtbp_var_(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_var_(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(1)
        DO 1 I=1,4
 1      DGG(I)=0.D0
        DGG(1)=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=x(1)-xmu
      d2=x(1)+umu
      r12=d1*d1+x(2)*x(2)
      r22=d2*d2+x(2)*x(2)
      r0=sqrt(r12)
      r1=sqrt(r22)
      r032=r12*r0
      r132=r22*r1
      omex=x(1)-(umu*(-xmu+x(1))/r032)-(xmu*(x(1)+umu)/r132)
      omey=x(2)*(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
```