Assignment 6 and 7–RTBP and its variational equations

Yixie Shao

April 23, 2015

1 Restricted Three-Body Problem

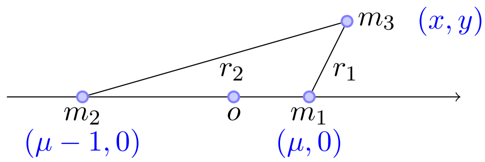

Figure 1: Restricted Three-Body Problem

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 Variational Equations

The variational equations of RTBP can be expressed as:

$$\begin{pmatrix} x_5' & x_6' & x_7' & x_8' \\ x_9' & x_{10}' & x_{11}' & x_{12}' \\ x_{13}' & x_{14}' & x_{15}' & x_{16}' \\ x_{17}' & x_{18}' & x_{19}' & x_{20}' \end{pmatrix} = M \begin{pmatrix} x_5 & x_6 & x_7 & x_8 \\ x_9 & x_{10} & x_{11} & x_{12} \\ x_{13} & x_{14} & x_{15} & x_{16} \\ x_{17} & x_{18} & x_{19} & x_{20} \end{pmatrix} \tag{7}$$

Here M is:

$$M = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \frac{\partial f_1}{\partial x_4} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \frac{\partial f_3}{\partial x_4} \\ \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial x_2} & \frac{\partial f_4}{\partial x_3} & \frac{\partial f_4}{\partial x_4} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \Omega_{x_1x_1} & \Omega_{x_1x_2} & 0 & 2 \\ \Omega_{x_1x_2} & \Omega_{x_2x_2} & -2 & 0 \end{pmatrix} \tag{8}$$

Where,

$$\begin{cases} \Omega_{x_1x_1} &= 1 - \frac{1-\mu}{{r_1}^3} - \frac{3(1-\mu)(x_1-\mu)^2}{{r_1}^5} - \frac{\mu}{{r_2}^3} + \frac{3\mu(x_1-\mu+1)^2}{{r_2}^5} \\ \Omega_{x_1x_2} &= x_2(\frac{3(1-\mu)(x_1-\mu)}{{r_1}^5} + \frac{3\mu(x_1-\mu+1)}{{r_2}^5}) \\ \Omega_{x_2x_2} &= 1 - \frac{1-\mu}{{r_1}^3} - \frac{\mu}{{r_2}^3} + {x_2}^2(\frac{3(1-\mu)}{{r_1}^5} + \frac{3\mu}{{r_2}^5}) \end{cases} \tag{9}$$

3 The implementation of the system

```
/* eqs of the planar RTPB pla in synodical coordinates */
extern MY_FLOAT xmu;
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;
r052=r12*r032;
r152=r22*r132;
diff(x1,t)=x3;
diff(x2,t)=x4;
        a=3*umu*(-xmu+x1)/r052;
        b=3*xmu*(x1+umu)/r152;
        c=3*umu/r052;
        d=3*xmu/r152;
        omex=x1-(umu*(-xmu+x1)/r032)-(xmu*(x1+umu)/r132);
        omey=x2*(1.-(umu/r032)-(xmu/r132));
        omexx=1-umu/r032-a*(-xmu+x1)-xmu/r132+b*(x1+umu);
        omexy=x2*(a+b);
        omeyy=1-umu/r032-xmu/r132+x2*x2*(c+d);
diff(x3,t)=2.*x4+omex;
diff(x4,t)=-2.*x3+omey;
diff(x5,t)=x13;
diff(x6,t)=x14;
diff(x7,t)=x15;
diff(x8,t)=x16;
diff(x9,t)=x17;
diff(x10,t)=x18;
diff(x11,t)=x19;
diff(x12,t)=x20;
diff(x13,t)=omexx*x5+omexy*x9+2*x17;
diff(x14,t)=omexx*x6+omexy*x10+2*x18;
diff(x15,t)=omexx*x7+omexy*x11+2*x19;
diff(x16,t)=omexx*x8+omexy*x12+2*x20;
diff(x17,t)=omexy*x5+omeyy*x9-2*x13;
diff(x18,t)=omexy*x6+omeyy*x10-2*x14;
diff(x19,t)=omexy*x7+omeyy*x11-2*x15;
diff(x20,t)=omexy*x8+omeyy*x12-2*x16;
```

4 Assignment 6–Integrate the RTBP and variational equations using Taylor

4.1 check the Jacobi integral

It is done in assignment 5.
Subroutine Jacobi:

```
      subroutine jacobi(n,x,Cja)
      IMPLICIT REAL*8 (A-H,O-Z)
      common/param/xmu
      dimension x(n)
      r1=sqrt((x(1)-xmu)*(x(1)-xmu)+x(2)*x(2))
      r2=sqrt((x(1)-xmu+1)*(x(1)-xmu+1)+x(2)*x(2))
      a=x(1)*x(1)+x(2)*x(2)
      omiga=0.5*a+(1-xmu)/r1+xmu/r2+0.5*(1-xmu)*xmu
      Cja=2*omiga-(x(3)*x(3)+x(4)*x(4))
      return
      end
```

4.2 check the determinant of matrix M

The initial value of matrix M is an identity matrix, i.e,

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The initial determinant of matrix M is 1.
we set $\mu = 9.53875 \times 10^{-4}$.
And the difference between determinant of every step and the initial value is less than 10^{-8}.
The final determinant of matrix M is 1.0000000000000007.

4.3 Code

Main function:

```
c
   *****************************************************************
c  MAIN_RTBP_VAR.F
```

```
c   input: xi,ti,tmax,idir,np
c
    *****************************************************************

         implicit real*8 (a-h,o-z)
         parameter (n=20)
         common/param/xmu
         dimension xi(n),x(n),oM(4,4)
         open(10,file='orbit.d',status='unknown')
         write(*,*) 'xmu'
         read(*,*) xmu
         xi(1)=0.1001005021494284d01
         xi(2)=0
         xi(3)=0
         xi(4)=-.1215976572734674d-02
         ti=0
         T=0.3138977039438897d01
         tmax=2.d0*T
         np=10
         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
         if (tmax.ge.ti)then
             idir=1
         else
             idir=-1
         endif
         do i=1,n
          x(i)=xi(i)
         enddo
          write(*,*)'Initial t:'
          write(*,*)ti
          write(*,*)'Initial cond:'
          write(*,*)(x(i),i=1,n)
         call jacobi(4,x,Cja)
          Cini=Cja
```

```
      write(*,*)Cini, 'initial value of Jacobi constant'
       do k=1,4
         do j=1,4
           oM(k,j)=x(4*k+j)
         enddo
       enddo
       write(*,*)'Initial matrix M:'
       write(*,*)(oM(i,1),i=1,4)
       write(*,*)(oM(i,2),i=1,4)
       write(*,*)(oM(i,3),i=1,4)
       write(*,*)(oM(i,4),i=1,4)
       call det(oM,detD,4)
       detDini=detD
       write(*,*)'Initial determinant:'
       write(*,*)detD
       write (10,*)ti,(x(ii),ii=1,n),Cini,detDini
       xinctime=dabs(tmax-ti)/np
       do 20 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
        call jacobi(4,x,Cja)
        deltaC=Cja-Cini
        if (dabs(deltaC).gt.1.d-12)then
                write(*,*)'problems in C'
                stop
        endif
        call det(oM,deta,4)
        write (10,*)ti,(x(ii),ii=1,n),Cja,deta
        difDeta=dabs(deta-1.d0)
        if(difDeta.gt.1D-8)then
         write (*,*) 'problem in determinant'
         stop
        endif
20     continue
       write(10,90)
90     format()
       write(*,*)'Final t:'
       write(*,*)ti
       write(*,*)'Final point:'
       write(*,*)(x(i),i=1,n)
      end


     subroutine flow(t,n,x,idir,xinctemps)
     IMPLICIT REAL*8 (A-H,O-Z)
```

```
      dimension x(n)
       tmax=t+idir*xinctemps
       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)
       if (idir.eq.1.and.t.lt.tmax)go to 1
       if (idir.eq.-1.and.t.gt.tmax)go to 1
       if (dabs(t-tmax).le.1.d-13)return
       write(*,*)'problems in taylor'
       stop
       return
       end
```

5 Assignment 7–Computation of a Poincare section for the RTBP

5.1 some test

Computation of a Poincare section y=0 for the RTBP with its variational equations.
The determinant of the matrix M is 1. And the Jacobi constant is 3.0019064499999999.
The period is $T = 3.138977039438897$.

Inicitial condition	nth-crossing	idir	Final Time	Final Point
(1.0010050214942841,0)	1	$+1$	3.1389770393838394	$x = 0.99978987398753205$ $y = -2.375154 \times 10^{-19}$
		-1	-3.1389770393838394	$x = 0.99978987398753205$ $y = 2.375154 \times 10^{-19}$
	2	$+1$	6.2779540784752941	$x = 1.0010050214942856$ $y = -5.035643 \times 10^{-20}$
		-1	-6.2779540784752941	$x = 1.0010050214942856$ $y = 5.035643 \times 10^{-20}$

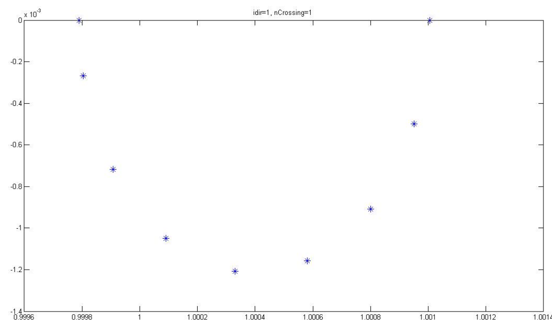

Figure 2: 1st-crossing, forward

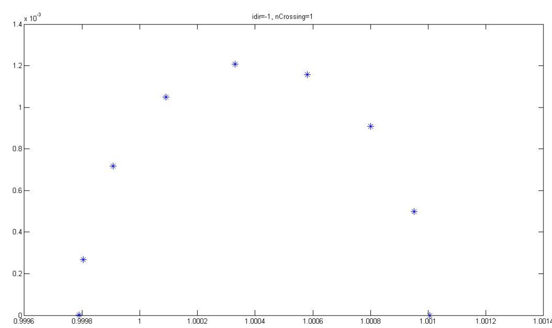

Figure 3: 1st-crossing, backward

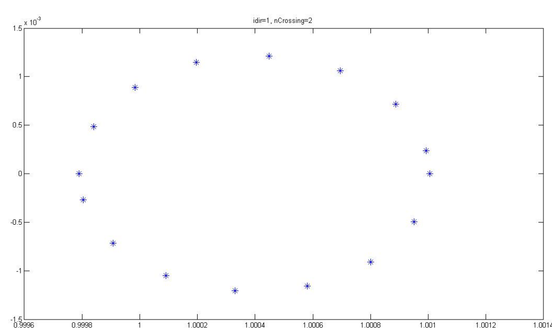

Figure 4: 2nd-crossing, forward

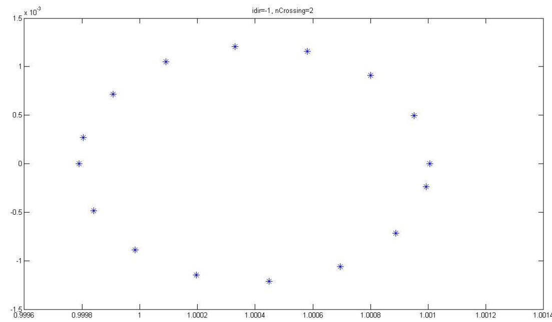

Figure 5: 2nd-crossing, backward

5.2 Code

Main function:

```
cc
c  MAIN_OS_SEC1_var.f
c
c
    ***************************************************************
c       We integrate the harmonic oscillator field with
    Taylor
c       up to the   n  crossing with the Poincare section: y
    =0
c                   -----
c
c !!!!   You should enter the code to integrate up to a
    given
c   'm_crossing'  crossing with the Poincare section: y=0
    !!!!!
c
c
    ***************************************************************

      implicit real*8 (a-h,o-z)
      parameter (n=20)
      common/param/xmu
      dimension yf(n),xi(n),x(n),oM(4,4)
      open(10,file='orbit.d',status='unknown')
      write(*,*) 'xmu'
      read(*,*) xmu
      xi(1)=0.1001005021494284d01
```

```
xi(2)=0
xi(3)=0
xi(4)=-.1215976572734674d-02
ti=0
T=0.3138977039438897d01
tmax=20.d0*T
np=10
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

write(*,*) 'idir?'
read(*,*) idir
write(*,*) 'm times crossing'
read(*,*) m


do i=1,n
 x(i)=xi(i)
enddo
write(*,*)'Initial t:'
 write(*,*)ti
 write(*,*)'Initial cond:'
 write(*,*)(x(i),i=1,n)
call jacobi(4,x,Cja)
 Cini=Cja
write(*,*)Cini, 'initial value of Jacobi constant'
 do k=1,4
   do j=1,4
     oM(k,j)=x(4*k+j)
   enddo
 enddo
 write(*,*)'Initial matrix M:'
 write(*,*)(oM(i,1),i=1,4)
 write(*,*)(oM(i,2),i=1,4)
 write(*,*)(oM(i,3),i=1,4)
```

```
         write(*,*)(oM(i,4),i=1,4)
         call det(oM,detD,4)
         detDini=detD
         write(*,*)'Initial determinant:'
         write(*,*)detD
         write (10,*)ti,(x(ii),ii=1,4),Cini,detDini
         t=0
        do i=1,m
          call poinc1(n,x,yf,tfinal,idir)
          x=yf
          t=tfinal+t
          write(*,*)'t:'
          write(*,*)t
          do k=1,4
           do j=1,4
             oM(k,j)=x(4*k+j)
           enddo
          enddo
          call jacobi(4,x,Cja)
          deltaC=Cja-Cini
          if (dabs(deltaC).gt.1.d-12)then
                  write(*,*)'problems in C'
                  stop
          endif
          call det(oM,deta,4)
          difDeta=dabs(deta-1.d0)
          if(difDeta.gt.1D-8)then
           write (*,*) 'problem in determinant'
           stop
          endif
        enddo

        end


        SUBROUTINE POINC1(n,YI,YF,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_var_(t,yi,idir,istep,pabs,
         prel,
     &  tmax,ht,iordre,ifl)
c     computation of first integral to be done
      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     REFINEMENT OF THE INTERSECTION POINT YF(*) USING
   NEWTON METHOD
C     TO GET A ZERO OF THE FUNCTION GG (SEE SUBROUTINE
   SECCIO)
 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_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(*,*)'tfinal point time: ',tfinal
      write(*,*)(yf(ii),ii=1,n)
      write(10,*)t,(yf(ii),ii=1,4)
      return
      end
```

```
C      THE SURFACE G OF SECTION.
        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      EQS OF MOTION IN synodical VARIABLES
      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
```