ASSIGNMENT 6 Integrating the RTBP using Taylor with Variational Equations

Ignacio Coca

In Assay 5 we checked that the Jacobi constant in the periodic orbit of the RTBP was conserved along the solutions. In assay 6 also checked that the determinant of the Jacobi matrix is 1 once we do a period along the orbit.

Now we are interested in finding the Poincare section of the orbit.

To do so, we just have to implement the Poincare routine in the program done in the previous assays, modifiing the equations to compute the 20 new ones.

```
 /* 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;
        omex=x1-(umu*(-xmu+x1)/r032)-(xmu*(x1+umu)/r132);
        omey=x2*(1.-(umu/r032)-(xmu/r132));

omexx=1-(umu*((r0*r0)-3.*d1)/(r0*r0*r0*r0*r0))-
(xmu*((r1*r1)-(3*(umu+x1)*(umu+x1)))/(r1*r1*r1*r1*r1));

omexy=x2*(((3*umu*d1)/(r0*r0*r0*r0*r0))+
(3*xmu*(x1+umu))/(r1*r1*r1*r1*r1));

omeyy=(1-(umu/(r0*r0*r0))-(xmu/(r1*r1*r1)))+
(x2*((3*umu*x2)/(r0*r0*r0*r0*r0))+(xmu*3*x2)/(r1*r1*r1*r1*r1));
```

```
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)=x5*omexx+x9*omexy+2.*x17;
diff(x14,t)=x6*omexx+x10*omexy+2.*x18;
diff(x15,t)=x7*omexx+x11*omexy+2.*x19;
diff(x16,t)=x8*omexx+x12*omexy+2.*x20;

diff(x17,t)=x5*omexy+x9*omeyy-2.*x13;
diff(x18,t)=x6*omexy+x10*omeyy-2.*x14;
diff(x19,t)=x7*omexy+x11*omeyy-2.*x15;
diff(x20,t)=x8*omexy+x12*omeyy-2.*x16;
```

The modified program is:

```
c************************************************************
c
c  MAIN_OS_FLOW.f
c
c       We integrate the harmonic oscillator field with Taylor
c       from t=ti up to t=tmax
c       idir= +1 (integration forward in time); =-1 (backward)
c       np= number of intermediate points (apart from the initial one)
c           that we want to write on the file orbit.d. If np=1
c           only the initial and final points are written
c
```

```
c  input: xi,ti,tmax,idir,np
c**************************************************************
        implicit real*8 (a-h,o-z)
        parameter (n=20)
        dimension yf(n)
        dimension xi(n),x(n)
        dimension oM(4,4)
        common/param/xmu
        write(*,*) 'xmu'
        read(*,*) xmu
        open(10,file='orbit.d',status='unknown')
C        write(*,*) 'Initial condition x(1),x(2)'
C        read(*,*) (xi(i),i=1,4)
        xi(1)=.1001005021494284d1
        xi(2)=0.
        xi(3)=0.
        xi(4)=-.1215976572734674d-2

C        write(*,*) 'ti,tmax,np (number of points)'
C        read(*,*)ti,tmax,np
        write(*,*) 'idir'
        read(*,*)idirorig
        write(*,*) 'ncross'
        read(*,*)ncross
c particular example integration up to t=pi
c        pi=4.d0*datan(1.d0)
        ti=0.
        tmax=2d0*.3138977039438897d1
        np=100

       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.

         r1=dsqrt((xi(1)-xmu)*(xi(1)-xmu)+xi(2)*xi(2))
         r2=dsqrt((xi(1)-xmu+1.d0)*(xi(1)-xmu+1.d0)+xi(2)*xi(2))
         omega=0.5d0*(xi(1)*xi(1)+xi(2)*xi(2))+(1.d0-xmu)/r1
      .  +xmu/r2+0.5d0*(1.d0-xmu)*xmu
         C_initial=2.d0*omega-(xi(3)*xi(3)+xi(4)*xi(4))


c         if (tmax.ge.ti)then
c          'idir (=1 forward in time, =-1 backward)'
c              idir=1
c         else
c              idir=-1
c         endif
         do i=1,n
          x(i)=xi(i)
         enddo
c          write(*,*)ti,'   initial t, initial cond:'
c          write(*,*)(x(i),i=1,n)
c REMARK: xinctime positive
         xinctime=dabs(tmax-ti)/np
          write (10,*)ti,(x(ii),ii=1,n)
         do 20 i=1,np
          call flow(ti,n,x,idirorig,xinctime)
          write (10,*)ti,(x(ii),ii=1,n)


                 r1=dsqrt((x(1)-xmu)*(x(1)-xmu)+x(2)*x(2))
         r2=dsqrt((x(1)-xmu+1.d0)*(x(1)-xmu+1.d0)+x(2)*x(2))
      omega=0.5d0*(x(1)*x(1)+x(2)*x(2))+(1.d0-xmu)/r1
      . +xmu/r2+0.5d0*xmu*(1.d0-xmu)
      C=2.d0*omega-(x(3)*x(3)+x(4)*x(4))

      if (dabs(C-C_initial).ge. 10d-12)then
```

```
      write(*,*) 'not same C'
     stop
    endif

 20    continue
c       write(*,*)ti,'  final t,  final point:'
c       write(*,*)(x(i),i=1,n)

       oM(1,1)=x(5)
       oM(1,2)=x(6)
       oM(1,3)=x(7)
       oM(1,4)=x(8)

       oM(2,1)=x(9)
       oM(2,2)=x(10)
       oM(2,3)=x(11)
       oM(2,4)=x(12)

       oM(3,1)=x(13)
       oM(3,2)=x(14)
       oM(3,3)=x(15)
       oM(3,4)=x(16)

       oM(4,1)=x(17)
       oM(4,2)=x(18)
       oM(4,3)=x(19)
       oM(4,4)=x(20)

       call DET(oM,ddet,4)

C      if(dabs(det-1).ge. 10d-12)then
C         write(*,*) 'determinant no 1'
C         stop
C      endif

    write(*,*) ddet

    DO i=1,ncross
        call poinc1(n,x,yf,tfinal,idirorig)
        x=yf
```

```
    end DO
        end
```

```
       subroutine flow(t,n,x,idirorig,xinctemps)
       IMPLICIT REAL*8 (A-H,O-Z)
       common/param/xmu
       dimension x(n)
       tmax=t+idirorig*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_(t,x,idirorig,istep,pabs,prel,
     &  tmax,ht,iordre,ifl)
c        write(10,100) t,(x(i),i=1,n)
        if (idirorig.eq.1.and.t.lt.tmax)go to 1
        if (idirorig.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
```

```
C***************************************************************************
c Input:
c n dimension of the vectors yi and yf
c yi initial point
c idirorig: +1 integration forwards in time; -1 backwards
c yf final point
c tfinal final time
c
C***************************************************************************
        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
c
c  we assume initial time t=0.
c
        ti=0.D0
C   DETERMINATION OF THE FIRST PASSAGE OF THE ORBIT THROUGH y=0
C
        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
c reasonable step:
        pas=0.4d0
        ht=0.d0
        t=ti
c |tmax| must be big enough
1        tmax=t+idir*pas
        CALL taylor_f77_eq_rtbp_(t,YI,idir,istep,pabs,prel,
     &  tmax,ht,iordre,ifl)
```

```
C
        CALL SECCIO(YI,GG,DGG)
        IF(GG*GA.LT.0.D0)go to 22
        GA=GG
        GO TO 1
C
C   REFINEMENT OF THE INTERSECTION POINT YF(*) USING NEWTON'S 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
c check p is not (or very close to) 0:  to be done
        if (h.ge.0.d0)idir=1
        if (h.lt.0.d0)idir=-1
        tmax=t+h
c          write(*,*)icont,'  refining: h and time ',h,tmax
c          write(*,*)'refining  t point ',t,yi(1),yi(2)
        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+tfinal
        write(*,*)'tfinal point time ',tfinal
        write(*,*)(yf(ii),ii=1,4)
C        write(10,*)tfinal,(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(2),DGG(2)
      GG=Y(2)
      DO 1 I=1,2
 1    DGG(I)=0.D0
      DGG(2)=1.d0

      RETURN
      END


C
C FIELD.F
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(20),f(20)
c

      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=dsqrt(r12)
      r1=dsqrt(r22)

      r032=r12*r0
      r132=r22*r1
      r052=r12*r032
      r152=r22*r132

      omex=x(1)-(umu*(-xmu+x(1))/r032)-(xmu*(x(1)+umu)/r132)
      omey=x(2)*(1.-(umu/r032)-(xmu/r132))

     omexx=1.-(umu*((r0*r0)-3.*d1)/(r0*r0*r0*r0*r0))
     .    -(xmu*( (r1*r1)-(3.*(umu+x(1))*(umu+x(1))) )/(r1*r1*r1*r1*r1))
     omexy=x(2)*(((3.*umu*d1)/(r0*r0*r0*r0*r0))
     .    +(3.*xmu*(x(1)+umu))/ (r1*r1*r1*r1*r1))
     omeyy=(1.-(umu/(r0*r0*r0))-(xmu/(r1*r1*r1))) + ( x(2)* ((3.
     .    *umu*x(2))/ (r0*r0*r0*r0*r0) )+ (xmu*3.*x(2))
     .      / (r1*r1*r1*r1*r1) )



      f(1)=x(3)
      f(2)= x(4)
      f(3)=2.*x(4)+omex
      f(4)=-2.*x(3)+omey

      f(5)=x(13)
      f(6)=x(14)
      f(7)=x(15)
```

```
      f(8)=x(16)

      f(9)=x(17)
      f(10)=x(18)
      f(11)=x(19)
      f(12)=x(20)

      f(13)=x(5)*omexx+x(9)*omexy+2.*x(17)
      f(14)=x(6)*omexx+x(10)*omexy+2.*x(18)
      f(15)=x(7)*omexx+x(11)*omexy+2.*x(19)
      f(16)=x(8)*omexx+x(12)*omexy+2.*x(20)

      f(17)=x(5)*omexy+x(9)*omeyy-2.*x(13)
      f(18)=x(6)*omexy+x(10)*omeyy-2.*x(14)
      f(19)=x(7)*omexy+x(11)*omeyy-2.*x(15)
      f(20)=x(8)*omexy+x(12)*omeyy-2.*x(16)

      return
      end
```

The initial conditions are:

```
periodic orbit in the  RTBP


    xmu=9.538750000000000E-004        C=3.001906450000000
    x=.1001005021494284E+01, y=x'=0,  y'=-.1215976572734674E-02,
    period/2=.3138977039438897E+01
```

The outputs for $ncross = 2$ are, for $idir = 1$ and $idir = -1$ respectively:

```
IDIR<-1

 xmu
```

```
9.538750000000000d-4
 idir
1
 ncross
2
  0.99999999999997780
 tfinal point time    3.1389770392961545
  0.99978987398753738        1.7854751755472226E-016   1.4745768539613613E-013   1.21
 tfinal point time    6.2779540780292766
   1.0010050214942907       -1.7784471843651761E-016   2.0556444066950157E-013  -1.21



 IDIR<--1

  xmu
9.538750000000000d-4
 idir
-1
 ncross
2
   1.0000000000000011
 tfinal point time   -3.1389770392961545
  0.99978987398753738       -1.7854751755472226E-016  -1.4745768539613613E-013   1.21
 tfinal point time   -6.2779540780292766
   1.0010050214942907        1.7784471843651761E-016  -2.0556444066950157E-013  -1.21
```

Where we can see that the determinant is 0.99999999999999523, which is almost 1, that the final point is the initial one, and that the final time is the period.