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 this assay we want also to check that the determinant of the Jacobi matrix is 1 once we do a period along the orbit.

We first compute the Jacobi matrix, and implement them in the program. To do this we just have to find the derivative of the equations with respect to each variable.

Once implemented these variational equations our 20 equations read:

```
 /* 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;
```

We now modify the program in Assay5 in order to compute this determinant. Is is done calling the det function in paquet_alg. The final 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 xi(n),x(n)
        dimension oM(4,4)
        common/param/xmu
        write(*,*) 'xmu'
        read(*,*) xmu
        open(10,file='orbit.d',status='unknown')
        write(*,*) 'Initial condition x(1),x(2)'
        read(*,*) (xi(i),i=1,4)


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

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,idir,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)



write(*,*) ddet
       end


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

This way at the end of the call of the program we obtain the value of the determinant, if C is conserved along the solutions (if C is not conserved the program will stop running and will send an error message).

We must note that the initial conditions of the variational equations are the identity matrix. We call the program using the following initial conditions for the remaining variables:

```
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 output of our program is:

```
 xmu
9.53875d-4
 Initial condition x(1),x(2)
.1001005021494284d1
0
0
-.1215976572734674d-2
 ti,tmax,np (number of points)
0
.313897703943889d1
100
 idir
1
  0.99999999999999523
```

Where we can see that the determinant is 0.99999999999999523, which is almost 1.