Assignment 10

Henry Mauricio Ortiz Osorio

May, 2015

Contents

1 Computation of homoclinic orbits of equilibrium points - CODE

```
c***************************************************************
c
c  MAIN_RTBP_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(ah,oz)
      parameter (n=4,m=4)

      dimension xi(n),x(n),O(m,m),mat(n,n),RR(n),RI(n),VR(n,n),VI(n,n)
      dimension v(n),p(2),yf(n)
      common/param/xmu


      open(10,file='orbit.d',status='unknown')
         write(*,*)'iregion'
      read(*,*)iregion
         write(*,*)'idir? (1 or -1)'
      read(*,*)idir
         write(*,*)'ncrossing?'
      read(*,*)ncrossing
      ti=0.d0
      tmax=6.28
      np=30
         write(*,*)'xmu?'
      read(*,*)xmu
      CALL peq(xmu,xl1,xl2,xl3,cl1,cl2,cl3)
```

```
   write(*,*)'xl3',xl3
   write(*,*)'cl3',cl3
C=cl3
x(1)=xl3
x(2)=0
x(3)=0
x(4)=0
CALL jacmat(n,x,xmu,mat)
   write(*,*)'mat'
doi=1,n
   write(*,*)(mat(i,j),j=1,n)
enddo
CALL vapvep(mat,n,RR,RI,VR,VI)
if(idir.gt.0)then
do i=1,n
if (RI(i).gt.0)then
k=i
endif
enddo
else
do i=1,n
if(RI(i).lt.0)then
k=i
endif
enddo
endif
k=1
do i=1,n
v(i)=VR(i,k)
enddo
p(1)=RR(k)
   write(*,*)'Eigenvalue',p(1)
   write(*,*)'v',(v(i),i=1,n)
s=1.d6
if(iregion.lt.0)then
s=s
endif
x=x+s*v
```

```
   write(*,*)'initialpoint',(x(i),i=1,n)
CALL jacobi(x,C,xmu,n)
ti=0
doj=1,ncrossing
t=0.d0
   write(10,*)t,(x(i),i=1,n)
CALL poinc1(j,xmu,n,m,x,yf,tfinal,idir,ti)
ti=ti+tfinal
end do
end



SUBROUTINE POINC1(j,xmu,n,m,YI,YF,tfinal,idirorig,ti)
IMPLICIT REAL*8(AH,OZ)
DIMENSION YI(n),YF(n),DGG(n),F(n)
icont=0
idir=idirorig



CALL SECCIO(YI,GG,DGG)
IF(DABS(GG).LT.1.D9)GG=0.d0
GA=GG
hab=.1e16
hre=.1e16
pabs=dlog10(hab)
prel=dlog10(hre)
istep=1


pas=0.4d0
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,n)
      GA=GG
      GO TO 1

  22    continue
      icont=icont+1
      if(icont.gt.20)then
         write(*,*)'problems finding the section'
      stop
      end if
      CALL FIELD(xmu,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


      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.D13) GO TO 22
      DO 4 I=1,N
  4   YF(I)=YI(I)
      tfinal=t


         write(*,*)'tfinalpointtime',tfinal
         write(*,*)(yf(ii),ii=1,n)
         write(10,*)t,(yf(ii),ii=1,n)
```

```
      return
      t=tfinal
      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
```

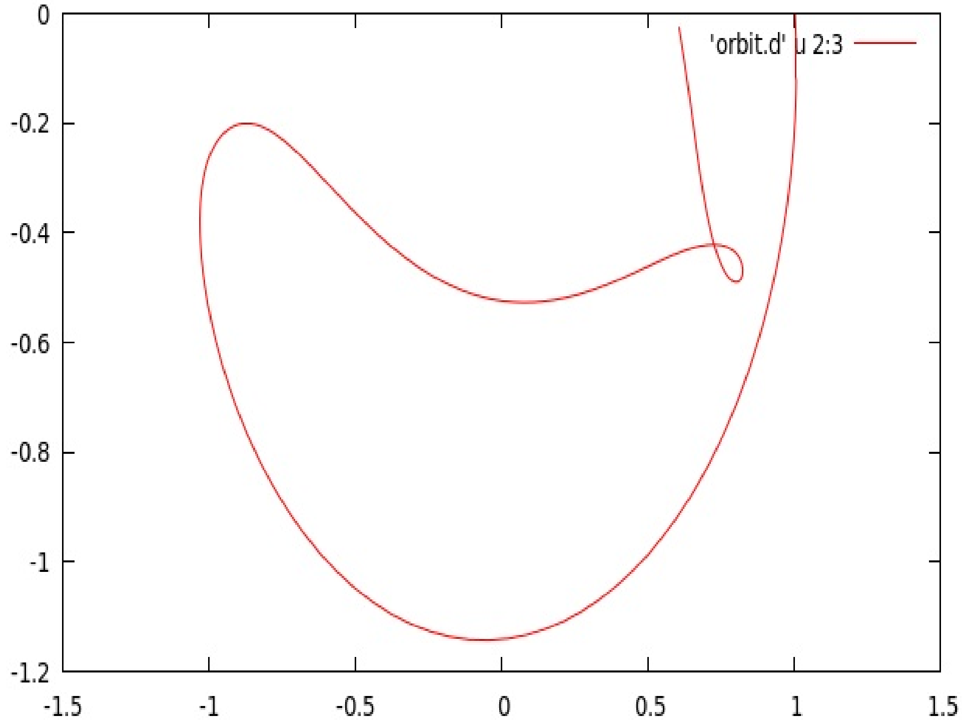
```
C
C***************************************************************************
      subroutine field(t,x,neq,f)
      implicit real*8 (a-h,o-z)
      common/param/xmu
      dimension x(20),f(20)
c
      umu=1.-xmu
      d1=x(1)-xmu
      d2=x(1)+umu
      r12=d1*d1+x(2)*x(2)
      r22=d2*d2+x(2)*x(2)
      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)
```

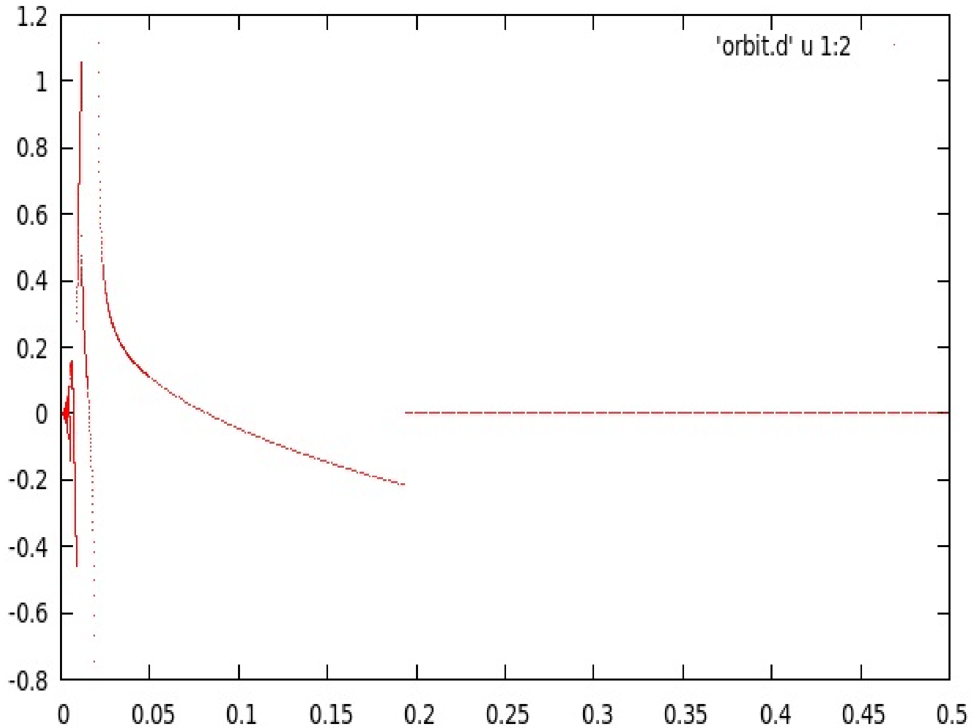
```
f(11)=x(19)
f(12)=x(20)
f(13)=x(5)*omexx+x(9)*omexy+2.*x(17)
f(14)=x(6)*omexx+x(10)*omexy+2.*x(18)
f(15)=x(7)*omexx+x(11)*omexy+2.*x(19)
f(16)=x(8)*omexx+x(12)*omexy+2.*x(20)
f(17)=x(5)*omexy+x(9)*omeyy-2.*x(13)
f(18)=x(6)*omexy+x(10)*omeyy-2.*x(14)
f(19)=x(7)*omexy+x(11)*omeyy-2.*x(15)
f(20)=x(8)*omexy+x(12)*omeyy-2.*x(16)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

2 PLOTS

2.1 Plot 1: (Orbit for $\mu = 0,008$)



2.2 Plot 2: (xmu,x') for xmu in (0,001,0.5)



2.3 Plot 2: (xmu,x’) for xmu in (0,001,0.1)