Numerics of Dynamical Systems

Assignment 11

Conny Schweigert

Barcelona
22.05.2015

1 Explanation

In this assignment, the familiy of Lyapunov periodic orbits around L3 was calculated. The Jacobi constant was varied in the interval (2.1,3.189) in 1000 steps. An initial x_1 was guessed ($x_1 = 1.9$). For this initial x_1 a x_4 was calculated according to the formula for the Jacobi constant with fixed constant C and fixed μ, while $x_2 = x_3 = 0$. This was done in *Comutey.f*.
Now to this initial value x the value $F(x)$ was calculated with *Ass10change.f*. *Ass10change2.f* is just another version which does not write data on 'orbit.d'.
Then x was varied to find $x1$ and $x2$ such that $F(x)$, i.e. x_3^{fin}, changes sign. Of course, either x_1 or x_2 is equal to the inital x. This was done in *varyx.f*.
Then x_1^{init} was refined with a bisection method in *bipoin.f* until a x_1^{init} was found with $x_3^{fin} = 0$
The programs are summarized in *Ass11.f* and the main program is *Lyapunov.f*.

2 Programme

Listing 1: Lyapunov.f

```
c*****************************************************************
c
c   Lyapunov.f
c
c  input:xmu
c*****************************************************************
         implicit real*8 (a-h,o-z)
         common/param/xmu
         parameter (n=4)
         dimension x(n),xfinal(n)
         m=1000
c           m=1
         do i = 1,(m+1)
         cinc = (3.189-2.10)/1000.d0
          xinc=1.9
         C = 3.189 - (i-1)*cinc
c         C=2.2
         call Ass11(C,xinc)
         enddo
          end
```

Listing 2: Ass11.f

```
c**********************************************************************
c
c   Ass11.f
c
c  input:xmu
c**********************************************************************
        subroutine Ass11(C,xinc)
        implicit real*8 (a-h,o-z)
        common/param/xmu
        parameter (n=4)
        dimension x(n),xfinal(n),xsfin(n)
        xmu=0.1
        open(20,file='orbit2.d',status='unknown')
         x(1)= xinc
         x(2) = 0
         x(3) = 0

         xin = x(1)
         sg = 1
         call computedy(xin,xmu,sg,C,dy)
         x(4)= dy

         write(*,*) 'C', C
        write(*,*) 'initial point', (x(i),i=1,n)

         call ass10change2(xin,dy,xfinal)
         write(*,*) 'final point', (xfinal(i),i=1,n)

         write(*,*) 'xin', xin
         call varyx(xin,xfinal,xmu,sg,C,x1,x2)
          write(*,*) 'x1,x2', x1,x2

         xmuh=xmu
         call bipoin(xin,xmuh,sg,C,x1,x2,xs,xsfin,dys)
           write(*,*) 'xs', xs
            write(*,*) 'dys', dys
           write(*,*) 'xsfinal', (xsfin(i),i=1,4)
         write(20,*) C, xs, xsfin(1)
         call ass10change(xs,dys,xfinal)
```

```
        end
```

Listing 3: Computey.f

```
c         Ass 11B
c       Given initial valuex, C and a sign of y',
c      sg must be +1 or -1
c      Output: y'




        subroutine computedy(xin,xmu,sg,C,dy)
        implicit real*8 (a-h,o-z)
        ro = dsqrt((xin - xmu)*(xin - xmu))
        rt = dsqrt((xin - xmu + 1.d0)*(xin - xmu + 1.d0))
        ome =0.5d0*(xin*xin)+(1.d0-xmu)/ro+xmu/rt+0.5d0*xmu*(1.d0-xmu)
        dy = sg*dsqrt(2*ome - C)
        end
```

Listing 4: Ass10change.f

```
c****************************************************************
c
c   Ass10change.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****************************************************************
        subroutine ass10change(xin,dy,xfinal)
        implicit real*8 (a-h,o-z)
        parameter (n=4,m=4)
        dimension xi(n),x(n),O(m,m),A(n,n),RR(n),RI(n),VR(n,n),VI(n,n)
        dimension yf(n),xfinal(n)
        common/param/xmu
        open(10,file='orbit.d',status='unknown')
        idir=1
```

```
      ncrossing=1
      ti=0.d0
      tmax = 6.28
      np=30
      xmu=0.1

      x(1)=xin
      x(2)=0
      x(3)=0
      x(4)=dy
      call JacobiMatrix(n,x,xmu,A)
      call vapvep(A,n,RR,RI,VR,VI)
      ti=0
      do j = 1,ncrossing
      t=0.d0
        write(10,*)t,(x(i),i=1,n)
        call poinc1(j,xmu,n,m,x,yf,tfinal,idir,ti)
        xfinal = yf
        ti = ti + tfinal
      end do
      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(j,xmu,n,m,YI,YF,tfinal,idirorig,ti)
      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
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_var_(t,yi,idir,istep,pabs,prel,
     &  tmax,ht,iordre,ifl)
c computation of first integral to be done
C
          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
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(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
c           write(*,*)icont,'   refining: h and time ',h,tmax
c           write(*,*)'refining   t  point  ',t,yi(1),yi(2)
         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,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(xmu,t,x,neq,f)
      implicit real*8 (a-h,o-z)
      dimension x(neq),f(neq)
c
      umu=1.d0-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
      c1=-umu/r032-xmu/r132
      c2=3.d0*umu/r052
      c3=3.d0*xmu/r152
        omex=x(1)-(umu*(-xmu+x(1))/r032)-(xmu*(x(1)+umu)/r132)
        omey=x(2)*(1.d0-(umu/r032)-(xmu/r132))
        omexx=c1+c2*d1*d1+c3*d2*d2+1.d0
        omexy=c2*d1*x(2)+c3*d2*x(2)
        omeyy=c1+(c2+c3)*x(2)*x(2)+1.d0
      f(1)=x(3)
      f(2)=x(4)
      f(3)=2.*x(4)+omex
      f(4)=-2.*x(3)+omey
      return
      end
```

```
c**********Jacobi-Matrix of x
      subroutine JacobiMatrix(n,x,xmu,A)
      implicit real*8 (a-h,o-z)
      dimension x(n),A(n,n)

      return
      end
```

Listing 5: varyx.f

```
c        Ass  11D
c      Given  initial  valuex ,  C  and  a  sign  of  y',
c     sg must be +1 or -1
c     Output: x1,x2, s.t. f(x1) and f(x2) have different signs



       subroutine varyx(xin,xfinal,xmu,sg,C,x1,x2)
       implicit real*8 (a-h,o-z)
       parameter (n=4)
       dimension x(n),xfinal(n),xnew(n),xnewfinal(n)
       xnew(1)=xin
       xnew(2)=0
       xnew(3)=0
       e = 1
       if(xfinal(3)>0) then
         a=1
         else
         a = -1
         endif

         x2 = xin
         do i = -1,1,2
           do j = 1,200
           xnew(1) = xin + i*j*0.01d0
           xn = xnew(1)
           call computedy(xn,xmu,sg,C,dy)
          xnew(4)= dy
          call Ass10change2(xn,dy,xnewfinal)
            if(a*xnewfinal(3)<0) then
            x1 = xnew(1)
```

```
             exit
             endif
          enddo
          if(a*xnewfinal(3)<0) then
            x1 = xnew(1)
             exit
             endif
        enddo
        if(x1>x2) then
        swap = x1
        x1=x2
        x2=swap
        endif
       end
```

Listing 6: bipoin.f

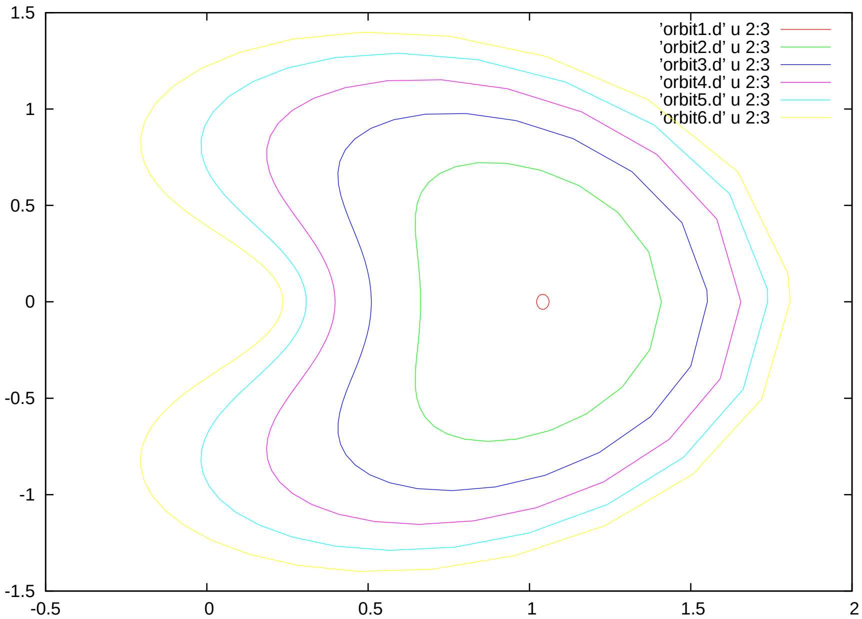
```
c Input: function f, points a < b with f(a)<0 and f(b)>0
c routine to compute  x with f(x)=0
c
        subroutine bipoin(xin,xmuh,sg,C,x1,x2,xs,xsfin,dys)
        implicit real*8 (a-h,o-z)
        common/param/xmu
        parameter (n=4)
        dimension x(n),xfinal(n),xsfin(n)
        xmu=xmuh
        tol = 0.000001
        nmax = 50
        e=1
        i=0
        do while(i.lt.nmax)
        xin = x1
        call computedy(xin,xmu,sg,C,dy)
        call Ass10change2(xin,dy,xfinal)
        fx1=xfinal(3)
c         write(*,*) fx1
        g = 0.5*(x1+x2)
        d = dabs(0.5*(x1-x2))
        xin=g
        call computedy(xin,xmu,sg,C,dy)
        call Ass10change2(xin,dy,xfinal)
```

```
fg=xfinal(3)
if(dabs(xfinal(3))<tol.OR.d<tol) then
xs=xin
dys = dy
endif

if(sign(e,fx1)==sign(e,fg)) then
x1=g
else
x2=g
endif
i=i+1
end do
call Ass10change(xs,dys,xsfin)
end
```

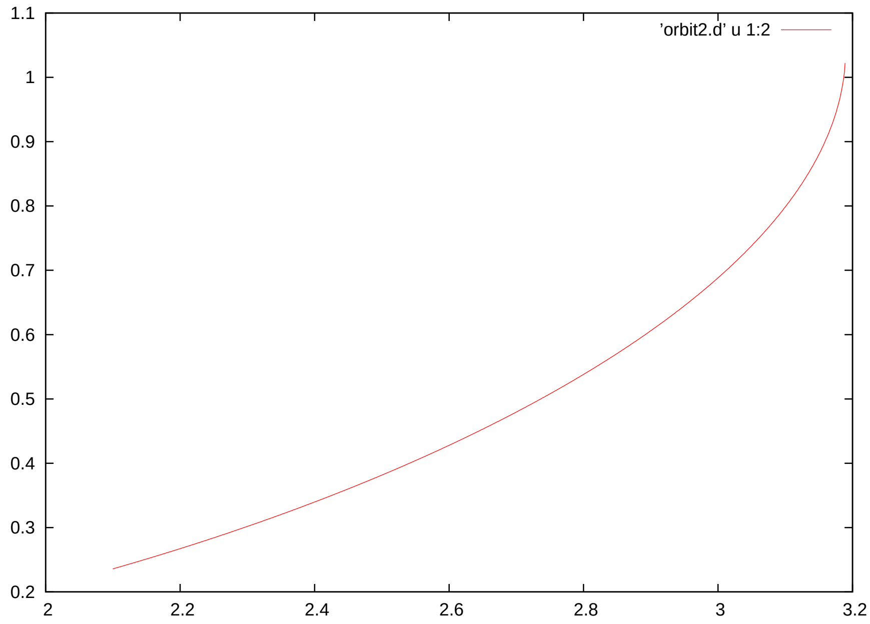
3 Plots

3.1 Graph of Lyapunov periodic orbits with C ∈ (2.1,3.189).



The inner orbit is the one with $C = 3.189$, the remotest is for $C = 2.1$.

3.2 Graph (x_1, C) of initial x_1 (with $x_3^{fin} = 0$) vs. C.



3.3 Graph (x_1, C) of final x_1 (after the first crossing) vs. C.