Numerics of Dynamical Systems

Assignment 7

Conny Schweigert

Barcelona
28.05.2015

1 Programme

```
c
c   MAIN7.f
c
c***************************************************************
c         We integrate the harmonic oscillator field with Taylor
c         up to the   FIRST  crossing with the Poincare section: y=0
c                     -----
c
c  !!!!   You should enter the code to integrate up to a given
c    'n_crossing'  crossing with the Poincare section: y=0   !!!!!
c
c***************************************************************
          implicit real*8 (a-h,o-z)
          parameter (n=20,m=4)
          dimension yf(n),x(n)
          common/param/xmu
c           write(*,*)'xmu'
c           read(*,*) xmu
c           write(*,*)'C'
c           read(*,*) C
          xmu = 9.538750000000000E-004
          C=3.001906450000000
          open(10,file='orbit.d',status='unknown')
          write(*,*) 'Initial condition x(1),...,x(n)'
          read(*,*) (x(i),i=1,n)
          write(*,*) 'idir?'
          read(*,*) idir
          write(*,*) 'ncrossing?'
          read(*,*) ncrossing
           xinit = x(1)
           yinit = x(2)
           dxinit = x(3)
           dyinit = x(4)
c
c we assume initial time t=0.d0
c
c

c          i = 0
```

```
        call jacobi(x,C,xmu,n)
        call matrix(x,m,n)
        ti=0
        do j = 1,ncrossing
        t=0.d0
          write(10,*)t,(x(i),i=1,n)
          call jacobi(x,C,xmu,n)
          call poinc1(j,xmu,n,m,x,yf,tfinal,idir,ti)
          ti = ti + tfinal
c         i = i + 1
        end do
c******Check on initial and final points**************************************
       xfinal = x(1)
       yfinal = x(2)
       dxfinal = x(3)
       dyfinal = x(4)
       xdiff = abs(xinit - xfinal)
       ydiff = abs(yinit - yfinal)
       dxdiff = abs(dxinit - dxfinal)
       dydiff = abs(dyinit - dyfinal)
       write(*,*)'xdiff,ydiff,dxdiff,dydiff'
       write(*,*) xdiff, ydiff, dxdiff, dydiff
        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
c check first integral: to be done
        write(*,*)'tfinal point time ',tfinal
        call checkperiod(j,tfinal)
        write(*,*)(yf(ii),ii=1,n)
        write(10,*)t,(yf(ii),ii=1,n)
        call matrix(yf,m,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.-xmu
      d1=x(1)-xmu
      d2=x(1)+umu
      r12=d1*d1+x(2)*x(2)
      r22=d2*d2+x(2)*x(2)
      r0=sqrt(r12)
      r1=sqrt(r22)
      r032=r12*r0
      r132=r22*r1
      r052=r12*r032
      r152=r22*r132
c1=-umu/r032-xmu/r132;
c2=3.*umu/r052;
c3=3.*xmu/r152;
        omex=x1-(umu*(-xmu+x1)/r032)-(xmu*(x(1)+umu)/r132)
        omey=x2*(1.-(umu/r032)-(xmu/r132))
        omexx=c1+c2*d1*d1+c3*d2*d2+1.
```

```
    omexy=c2*d1*x(2)+c3*d2*x(2)
    omeyy=c1+(c2+c3)*x(2)*x(2)+1.
  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)=omexx*x(5)+omexy*x(9)+2.*x(17);
  f(14)=omexx*x(6)+omexy*x(10)+2.*x(18);
  f(15)=omexx*x(7)+omexy*x(11)+2.*x(19);
  f(16)=omexx*x(8)+omexy*x(12)+2.*x(20);
  f(17)=omexy*x(5)+omeyy*x(9)-2.*x(13);
  f(18)=omexy*x(6)+omeyy*x(10)-2.*x(14);
  f(19)=omexy*x(7)+omeyy*x(11)-2.*x(15);
  f(20)=omexy*x(8)+omeyy*x(12)-2.*x(16);
  return
  end

c***************************************************************************
   subroutine jacobi(x,C,xmu,n)
   implicit real*8 (a-h,o-z)
   dimension x(n)
   ro = dsqrt((x(1) - xmu)*(x(1) - xmu)+x(2)*x(2))
   rt = dsqrt((x(1) - xmu + 1.d0)*(x(1) - xmu + 1.d0)+x(2)*x(2))
   rest = 0.5d0 * xmu * (1.d0-xmu)
   ome = 0.5d0*(x(1)*x(1) + x(2)*x(2)) + (1.d0 - xmu)/ro + xmu / rt + rest
   Cnew = 2*ome - (x(3)*x(3) + x(4)*x(4))
    Cdiff = dabs(C - Cnew)
   if (Cdiff.gt.1.d-3) then
   write(*,*) 'Jacobi integral not conserved'
   endif
   end
c*************************************************************************
  subroutine checkperiod(j,tfinal)
```

```
      implicit real*8 (a-h,o-z)
      tabs = dabs(j*3.1415926535897932-dabs(tfinal))
      if (tabs.gt.1.d-2) then
       write(*,*) 'Time of integration is not equal to period.'
       write(*,*) tfinal, tabs
       endif
       end

c******** Test on determinant of M******************************************
        subroutine matrix(x,m,n)
       implicit real*8 (a-h,o-z)
       dimension x(n),O(m,m)
        do k=1,m
          do l=1,m
            O(k,l) = x(4*k+l)
          end do
        end do
c       write(*,*) O
        call DET(O,DETA,m)
        z = dabs(DETA - 1.d0)
        if(z.gt.1.d-2) then
          write(*,*) 'Determinant of M is not equal to one'
        endif
        end
```

1.1 Output forward in time

```
conny.schweigert@fme-desktop:~$ ./main7
 Initial condition x(1),...,x(n)
.1001005021494284E+01,0,0,-.1215976572734674E-02,   1.,0.,0.,0.,0.,1.,0.,0.,0.,(
 idir?
1
 ncrossing?
2
 tfinal point time    3.13897693350963 23
  0.99978987394644436         1.7912643810439097E-016  -6.4576384369388235E-011
1.2163463701885659E-003    7.0153665494799284         -5.2489979930799482E-003
1.0475060671352612E-002    4.0055577223722585         -18.837593423133107
1.0093230246602709        -4.0104327089144531         -9.4138001693745235
4.2890525399306149E-002   -3.3595800315628873E-003  -0.99086803598092688
```

```
2.1433939975484675E-002  -12.023378577508536        7.8749066594723284E-003
-1.5715457148851950E-002  -7.0091084271627802
 tfinal point time    6.2779542924614660
   1.0010050214929316        -1.7708477106415289E-016  -1.3062932759911169E-010
-1.2159765707076262E-003   1.1981155379664261         -1.0637317840478481E-002
2.1228155898936184E-002   9.9005221323208575E-002  -38.094402657804849
1.0496114964583574        -9.9005218846910117E-002  -19.037097054844697
4.0438335724669157E-002  -5.2663247359112523E-005   1.0001050953216579
2.0208548163620457E-002 -0.29695541479532023        1.5944277538473613E-002
-3.1818993950939137E-002  0.85160105629994187
 xdiff , ydiff , dxdiff , dydiff
   1.3524736885983657E-012   1.7708477106415289E-016   1.3062932759911169E-010
2.0270478004458248E-012
```

1.2 Output backward in time

```
conny.schweigert@fme-desktop:~$ ./main7
 Initial condition x(1),...,x(n)
.1001005021494284E+01,0,0,-.1215976572734674E-02,    1.,0.,0.,0.,0.,1.,0.,0.,0.,0
 idir?
-1
 ncrossing?
2
 tfinal point time   -3.1389769335096323
  0.99978987394644436        -1.7912643810439097E-016   6.4576384369388235E-011
1.2163463701885659E-003   7.0153665494799284         5.2489979930799482E-003
-1.0475060671352612E-002   4.0055577223722585         18.837593423133107
1.0093230246602709        -4.0104327089144531         9.4138001693745235
-4.2890525399306149E-002  -3.3595800315628873E-003 -0.99086803598092688
-2.1433939975484675E-002  -12.023378577508536        -7.8749066594723284E-003
1.5715457148851950E-002  -7.0091084271627802
 tfinal point time   -6.2779542924614660
   1.0010050214929316         1.7708477106415289E-016   1.3062932759911169E-010
-1.2159765707076262E-003   1.1981155379664261          1.0637317840478481E-002
-2.1228155898936184E-002   9.9005221323208575E-002   38.094402657804849
1.0496114964583574        -9.9005218846910117E-002   19.037097054844697
-4.0438335724669157E-002  -5.2663247359112523E-005   1.0001050953216579
-2.0208548163620457E-002 -0.29695541479532023        -1.5944277538473613E-002
3.1818993950939137E-002  0.85160105629994187
 xdiff , ydiff , dxdiff , dydiff
```

1.3524736885983657E−012 1.7708477106415289E−016 1.3062932759911169E−010
2.0270478004458248E−012

Abbildung 1: Retrograde orbit

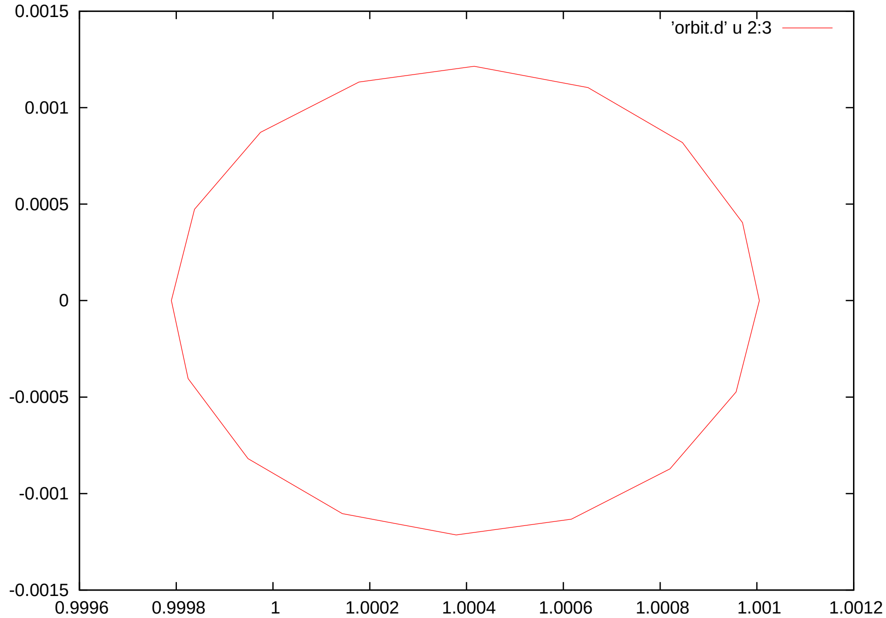

Abbildung 2: Direct orbit