Numerics of Dynamical Systems

Assignment 8

Conny Schweigert

Barcelona
28.05.2015

1 Programme

Listing 1: main_rtbp_flow8.f

```
c***********************************************************************
c
c   MAIN_RTBP_FLOW8.f
c
c  input:xmu
c***********************************************************************
        implicit real*8 (a-h,o-z)
        parameter (n=20,m=100)
        dimension xi(n),x(n)
        common/param/xmu
c         write(*,*) 'xmu'
c         read(*,*) xmu
        open(10,file='orbit.d',status='unknown')
         do i=1,m
         xmu = i*(0.5/m)
         call peq(xmu,xl1,xl2,xl3,cl1,cl2,cl3)
         write (10,*) xmu, xl1, xl2, xl3, cl1, cl2, cl3
         enddo

         write(*,*)  'xl1,xl2,xl3', xl1,xl2,xl3
         write(*,*)  'cl1,cl2,cl3', cl1,cl2,cl3

         end
```

Listing 2: main_rtbp_flow8.f

```
c
c You must write 2 parts of the code:
c
c 1. main program: enter as input: xmu,
c    obtain as output xl1,xl2,xl3,Cl1,Cl2,Cl3
c   Typical call to the routine for the computation of this output:
c 1) we compute L1,L2,L3
c  1            call peq(xmu,xl1,xl2,xl3,cl1,cl2,cl3)
c
c 2. A routine to compute the Jacobi integral 2*Omega(x,y)-(x'^2+y'^2)=C
c  BUT for a collinear equilibrium point, it is simply
c        C=2*Omega(x,0)
```

```
c
c
c
c
c routine to compute xl1, xl2, xl3, Cl2, Cl2, Cl3
c
        subroutine peq(xmu, xl1, xl2, xl3, cl1, cl2, cl3)
        implicit real*8(a-h,o-z)
        a=1.d0/3.d0
        i=0
        umu=1.d0-xmu
c to compute L2 (on the left hand side of the small primary)
        x=xmu/(3.d0*(1.d0-xmu))
        x=x**a
  1        den=3.d0-2.d0*xmu+x*(3.d0-xmu+x)
        f=xmu*(1.d0+x)**2/den
        f=f**a
        x1=xmu-1.d0-x
        if (dabs(x-f).le.1.d-15)then
c CALL .... and compute C(L2)
         xl2=X1
         xl=xl2
             call c(xmu, xl, cl)
             xl2=xl
             cl2=cl
             go to 3
        endif
        i=i+1
        x=f
        go to 1
   2    format(e25.16,',',e25.16,',',e25.16)
   3    continue
c
c L1 (between the primaries)
c
        i=0
        x=xmu/(3.d0*(1.d0-xmu))
        x=x**a
  10        den=3.d0-2.d0*xmu-x*(3.d0-xmu-x)
        f=xmu*(1.d0-x)**2/den
        f=f**a
```

```
      x1=xmu-1.d0+x
      if (dabs(x-f).le.1.d-15)then
c CALL .... and compute C(L1)
       XL1=X1
       xl=xl1
          call c(xmu,xl,cl)
          xl1=xl
          cl1=cl
          go to 4
      endif
      i=i+1
      x=f
      go to 10
 4    continue
c
c L3 (on the right hand side of the big primary)
c
      i=0
      x=1-(7/12)*xmu
      x=x**a
12        den=1+2.d0*xmu+x*(2.d0+xmu+x)
      f=((1-xmu)*(1.d0+x)*(1.d0+x))/den
      f=f**a
      X1=xmu+x
      if (dabs(x-f).le.1.d-15)then
c CALL .... and compute C(L2)
       xl3=X1
       xl=xl3
          call c(xmu,xl,cl)
          xl3=xl
          cl3=cl
          go to 11
      endif
      i=i+1
      x=f
      go to 12
11    continue
      end

      subroutine c(xmu,xl,cl)
      implicit real*8(c-h,o-z)
```

```
        ro = dabs(xl-xmu)
        r1 = dabs(xl-xmu+1)
        cl = (xl*xl)+2.d0*((1.d0-xmu)/ro)+2.d0*(xmu/r1)+(xmu*(1.d0-xmu))
        end
```

Abbildung 1: μ vs. xl

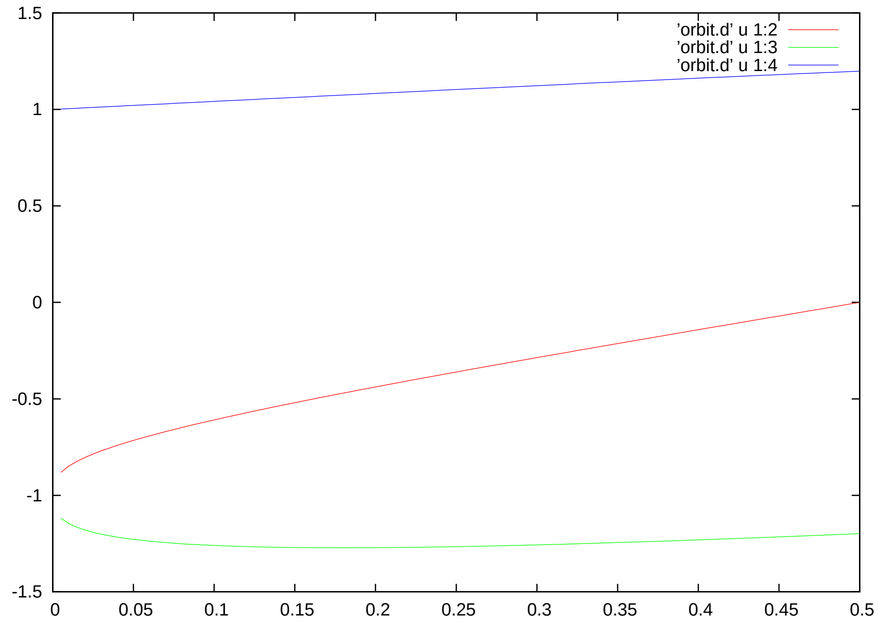

Abbildung 2: μ vs. cl