Assignment 4–Integrate the RTBP using Taylor

Yixie Shao

April 8, 2015

1 Restricted Three-Body Problem

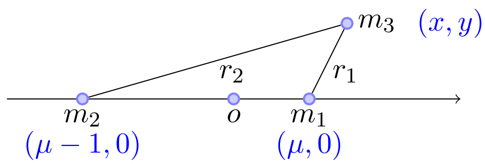

Figure 1: Restricted Three-Body Problem

The equations of motion are:

$$\begin{cases} x'' - 2y' &= \Omega_x \\ y'' + 2x' &= \Omega_y \end{cases} \tag{1}$$

And,

$$\Omega(x, y) = \frac{(x^2 + y^2)}{2} + \frac{1 - \mu}{r_1} + \frac{\mu}{r_2} + \frac{\mu(1 - \mu)}{2} \tag{2}$$

Here:

$$\begin{cases} \mu &= \frac{m_2}{m_1 + m_2} \\ r_1 &= \sqrt{(x - \mu)^2 + y^2} \\ r_2 &= \sqrt{(x - \mu + 1)^2 + y^2} \end{cases}$$

Let:

$$\begin{cases} x_1 &= x \\ x_2 &= y \\ x_3 &= x' \\ x_4 &= y' \end{cases}$$

The RTBP is expressed as:

$$\begin{cases} \dot{x}_1 &= x_3 \\ \dot{x}_2 &= x_4 \\ \dot{x}_3 &= 2x_4 + \Omega_{x_1} \\ \dot{x}_4 &= -2x_3 + \Omega_{x_2} \end{cases} \tag{3}$$

Where,

$$\begin{cases} \Omega_{x_1} &= x_1 - \frac{(1-\mu)(x_1-\mu)}{{r_1}^3} - \frac{\mu(x_1-\mu+1)}{{r_2}^3} \\ \Omega_{x_2} &= x_2(1 - \frac{1-\mu}{{r_1}^3} - \frac{\mu}{{r_2}^3}) \end{cases}$$

2 Periodic Orbit

Here, we set $\mu = 9.53875 \times 10^{-4}$.

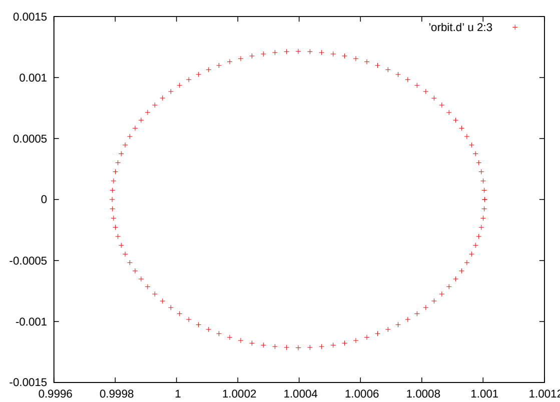

Figure 2: $idir = 1$

Initial time	0	Final time	6.2779540788778050
$x_i(1)$	1.0010050214942841	$x_f(1)$	1.0010050214942869
$x_i(2)$	0	$x_f(2)$	-4.9694609863780437E-013
$x_i(3)$	0	$x_f(3)$	-1.5562634595725142E-013
$x_i(4)$	-1.2159765727346740E-003	$x_f(4)$	-1.2159765727387431E-003

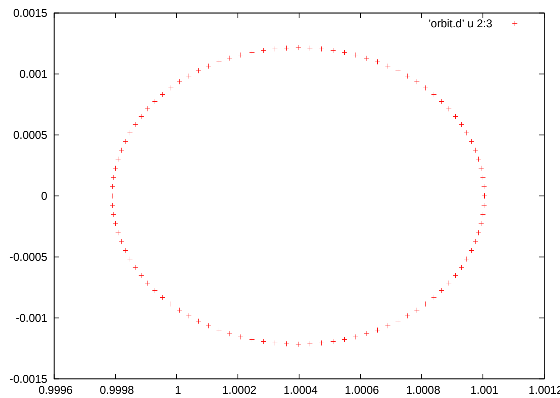

Figure 3: $idir = -1$

Initial time	0	Final time	-6.2779540788778050
$x_i(1)$	1.0010050214942841	$x_f(1)$	1.0010050214942869
$x_i(2)$	0	$x_f(2)$	4.9694609863780437E-013
$x_i(3)$	0	$x_f(3)$	1.5562634595725142E-013
$x_i(4)$	-1.2159765727346740E-003	$x_f(4)$	-1.2159765727387431E-003

3 Jacobi integral

Jacobi integral C is:

$$C = 2\Omega(x, y) - (x'^2 + y'^2) \tag{4}$$

Jacobi integral of the initial point is $C_{ini} = 3.0009534848775155$. And the $\Delta C = C - C_{ini}$ in every point is smaller than 10^{-12}. So the Jacobi integral is conserved.

4 code

```
c
  *****************************************************************
c
c   MAIN_RTBP_FLOW.f
c
c
c   input: xi,ti,tmax,idir,np,xmu
c
  *****************************************************************

      implicit real*8 (a-h,o-z)
      parameter (n=4)
      common/param/xmu
      dimension xi(n),x(n)
      open(10,file='orbit.d',status='unknown')
      write(*,*) 'xmu'
      read(*,*) xmu

      xi(1)=0.1001005021494284d01
      xi(2)=0
      xi(3)=0
      xi(4)=-.1215976572734674d-02
      ti=0
      T=0.3138977039438897d01
c     tmax=2.d0*T
      tmax=-2.d0*T
      np=100

      if (tmax.ge.ti)then
          idir=1
      else
          idir=-1
      endif
      do i=1,n
       x(i)=xi(i)
      enddo
       write(*,*)ti,'   initial t, initial cond:'
       write(*,*)(x(i),i=1,n)
     call jacobi(n,x,Cja)
     Cini=Cja

     xinctime=dabs(tmax-ti)/np
       write (10,*)ti,(x(ii),ii=1,n),Cini
     do 20 i=1,np
       call flow(ti,n,x,idir,xinctime)
```

```
         call jacobi(n,x,Cja)
         deltaC=Cja-Cini
         write (10,*)ti,(x(ii),ii=1,n),Cja
         if (dabs(deltaC).gt.1.d-12)then
                  write(*,*)'problems in C'
                  stop
         endif
20     continue
       write(*,*)ti,'  final t,  final point:'
       write(*,*)(x(i),i=1,n)
       end


       subroutine flow(t,n,x,idir,xinctemps)
       IMPLICIT REAL*8 (A-H,O-Z)
       dimension x(n)
        tmax=t+idir*xinctemps
        hab=0.1e-16
        hre=0.1e-16
        pabs=dlog10(hab)
        prel=dlog10(hre)
        istep=1
        ht=0.d0
1         CALL taylor_f77_eq_rtbp_(t,x,idir,istep,pabs,prel,
     & tmax,ht,iordre,ifl)
        if (idir.eq.1.and.t.lt.tmax)go to 1
        if (idir.eq.-1.and.t.gt.tmax)go to 1
        if (dabs(t-tmax).le.1.d-13)return
        write(*,*)'problems in taylor'
        stop
        return
        end


        subroutine jacobi(n,x,Cja)
        IMPLICIT REAL*8 (A-H,O-Z)
        common/param/xmu
        dimension x(n)
        r1=sqrt((x(1)-xmu)*(x(1)-xmu)+x(2)*x(2))
        r2=sqrt((x(1)-xmu+1)*(x(1)-xmu+1)+x(2)*x(2))
        omiga=0.5*(x(1)*x(1)+x(2)*x(2))+(1-xmu)/r1+xmu/r2
            +0.5*(1-xmu)*xmu
        Cja=2*omiga-(x(3)*x(3)+x(4)*x(4))
        end
```