LAGRANGE RK4 EXAMPLE

```
;X acceleration;X acceleration

pro visorbit, inx,iny

nx=n_elements(inx)

window,xsize=1000,ysize=1000

plot,inx,iny,xrange=[.2,1],yrange=[-.5,.5],ystyle=1,xstyle=1,psym=3

FOR i=0,nx-1 DO BEGIN

oplot,inx[i],iny[i],psym=3,thick=4

wait=.1

ENDFOR

END


FUNCTION xddot, vy,x,u,r1,r2

term1=2.0*vy+x

term2=-1.0*u*(x-1.0+u)/r1^3.

term3=-1.0*(1.0-u)*(x+u)/r2^3.

return,term1+term2+term3

END


;Y acceleration

FUNCTION yddot,vx,y,u,r1,r2

term1=y-2.0*vx

term2=-1.0*u*y/r1^3.

term3=-1.0*(1.0-u)*y/r2^3.

return, term1+term2+term3

END
```

```
FUNCTION rad1,x,y,u
return, sqrt((x-1.0+u)^2.+y^2.)
END

FUNCTION rad2,x,y,u
return, sqrt((x+u)^2.+y^2.)
END

FUNCTION lpts,a,u
;calculate lagrange points as a function of mu where mu = m1 and m1 < m2
IF a eq 1 THEN BEGIN
alpha=(u/(3d0*(1.0-u)))^(1./3.)
r2=alpha-alpha^2/3d0-alpha^3./9d0-23d0*alpha^3./81d0
return, [(1d0-u)-r2,0d0]
ENDIF

IF a eq 2 THEN BEGIN
alpha=(u/(3d0*(1.0-u)))^(1./3.)
r2=alpha+alpha^2/3d0-alpha^3./9d0-31d0*alpha^3./81d0
return,[r2+(1d0-u),0d0]
ENDIF

IF a eq 3 THEN BEGIN
rat=u/(1d0-u)
b=-7d0*rat/12d0+7d0*rat^2./12d0-13223d0*rat^3./20736
```

```
r2=2d0+b
return,[(1d0-u)-r2,0]
ENDIF

IF a eq 4 THEN BEGIN
Return, [.5d0-u,sqrt(3d0)/2.]
ENDIF
IF a eq 5 THEN BEGIN
Return, [.5d0-u,-1.0*sqrt(3d0)/2.]
ENDIF

END
PRO lagrange,sep,msmall,mbig, values

 ;a is separation between m1 and m2 in m
 a=1.49598d11
 a=sep*a
 ms=1.9891d30
 ;m2>m1 and on negative x for this formulation to work
 m1=msmall*ms
 m2=mbig*ms
;radial separation from CM_system
 m=m2/(m1+m2)
 r1=m*a
 r2=(m-1.0)*a
 bounds=[r1,r2]
 kmin=min(bounds)*1.5
```

```
 kmax=max(bounds)*1.5
 kmin=-2.0*a
 kmax=2.0*a
 ;gravitational constant
 G=-6.67428d-11
 norm=G*(m1+m2)/a
 ;array size-keep this even so we kill zeros
 n=1000
 ;make kx and ky and r array where r is distance from center of mass
 k = dindgen(n)*(kmax-kmin)/(n-1.)+kmin
 kx =k#make_array(n,value=1.0)
 ky =make_array(n,value=1.0)#k
 kmag2xy = kx*kx+ky*ky
 r=sqrt(kmag2xy)
 ;need s1,s2 = |r-r1|, |r-r2|
 ;s2=sqrt((kx-r2)^2.+ky^2.)
 ;s1=sqrt((kx-r1)^2.+ky^2.)
 s2=sqrt(r2^2.+r^2.0-2.0*r2*kx)
 s1=sqrt(r1^2.+r^2.0-2.0*r1*kx)
 pot1= G*m1/s1
 pot2= G*m2/s2
 pot3= G*.5*(m1+m2)*r^2./a^3.
 potential=pot1+pot2+pot3
potnorm=potential/(-1.0*norm)
dx=abs(kx[2]-kx[1])
dy=dx
grad_x=(SHIFT(potential, -1, 0) - SHIFT(potential, 1, 0))/(2 * dx)
```

```
grad_y=(SHIFT(potential, 0, -1) - SHIFT(potential, 0, 1))/(2 * dy)

 values =
{r1:r1,r2:r2,kx:kx,ky:ky,kmag2xy:kmag2xy,pot:potential,potnorm:potnorm,m1:m1,
m2:m2, a:a, grad_x:grad_x, grad_y:grad_y}


END


PRO l4orbit, t,x,y

;returns x and y orbit of a particle librating about l4

x=(3.54d-4)*sin(.268*t)-(9.85d-5)*sin(.963*t)

y=(6.23d-5)*cos(.268*t)-(4.86d-5)*cos(.963*t)

END


PRO orbits,x,y,vx,vy,mu,dt,tstop,outarray

;initialize arrays

;l4/l5 y=+-sqrt(3.0)/2. x=.5-mu


  i=0L

  dt=double(dt)

  t=dt

  tarray=t

  mu=double(mu)

  x0=double(x)

  y0=double(y)

  vx0=double(vx)

  vy0=double(vy)
```

```
 xarray=x0
 yarray=y0
 vxarray=vx0
 vyarray=vy0
 parray=-mu/rad1(x0,y0,mu)^2.-(1.0-mu)/rad2(x0,y0,mu)^2. -.5*(x0^2.+y0^2.)

;m1<m2
;mu = M1
;1-mu = M2
;M2+M1=1
;r1=1-mu
;r2=-mu
;G=omega=1


;MAIN INTEGRATION LOOP
 WHILE t lt tstop  DO BEGIN

 ;print,"At t = ", t," x0 = ",x0," y0 = ",y0," vx0 = ",vx0," vy0 = ",vy0
;UPDATE x,y,vx,vy - FIRST pass - x1=vx0*dt, v1=a0*dt
  kx1=vx0*dt ;x1
  ky1=vy0*dt ;y1
  kvx1=xddot(vy0,x0,mu,rad1(x0,y0,mu),rad2(x0,y0,mu))*dt ;vx1
  kvy1=yddot(vx0,y0,mu,rad1(x0,y0,mu),rad2(x0,y0,mu))*dt ;vy1
  ;print,"At t = ", t," kx1 = ",kx1," ky1 = ",ky1," kvx1 = ",kvx1," kvy1 = ",kvy1
```

```
;UPDATE x,y,vx,vy - SECOND pass, use at x0+k1/2 so new x is x0+vx0dt, new v is vx0+a*dt

   kx2=(vx0+kvx1/2.)*dt ;x2

   ky2=(vy0+kvy1/2.)*dt ;y2

kvx2=xddot(vy0+kvy1/2.,x0+kx1/2.,mu,rad1(x0+kx1/2.,y0+ky1/2.,mu),rad2(x0+kx1/2.,y0+ky1/2.,mu))*dt ;vx2

kvy2=yddot(vx0+kvx1/2.,y0+ky1/2.,mu,rad1(x0+kx1/2.,y0+ky1/2.,mu),rad2(x0+kx1/2.,y0+ky1/2.,mu))*dt ;vy2

   ;print,"At t = ", t," kx2 = ",kx2," ky2 = ",ky2," kvx2 = ",kvx2," kvy2 = ",kvy2


;UPDATE x,y,vx,vy - THIRD pass

   kx3=(vx0+kvx2/2.)*dt ;x3

   ky3=(vy0+kvy2/2.)*dt ;y3

kvx3=xddot(vy0+kvy2/2.,x0+kx2/2.,mu,rad1(x0+kx2/2.,y0+ky2/2.,mu),rad2(x0+kx2/2.,y0+ky2/2.,mu))*dt ;vx3

kvy3=yddot(vx0+kvx2/2.,y0+ky2/2.,mu,rad1(x0+kx2/2.,y0+ky2/2.,mu),rad2(x0+kx2/2.,y0+ky2/2.,mu))*dt ;vy3

   ;print,"At t = ", t," kx3 = ",kx3," ky3 = ",ky3," kvx3 = ",kvx3," kvy3 = ",kvy3


;UPDATE x,y,vx,vy - FINAL pass

   kx4=(vx0+kvx3)*dt ;x4

   ky4=(vy0+kvy3)*dt ;y4

kvx4=xddot(vy0+kvy3,x0+kx3,mu,rad1(x0+kx3,y0+ky3,mu),rad2(x0+kx3,y0+ky3,mu))*dt ;vx4

kvy4=yddot(vx0+kvx3,y0+ky3,mu,rad1(x0+kx3,y0+ky3,mu),rad2(x0+kx3,y0+ky3,mu))*dt ;vy4
```

```
    ;print,"At t = ", t," kx4 = ",kx4," ky4 = ",ky4," kvx4 = ",kvx4," kvy4 = ",kvy4

;UPDATE x0,y0, vx0, vy0 to xn,yn,vxn,vyn
    xn = x0+(kx1+2.0*(kx2+kx3)+kx4)/6.
    yn = y0+(ky1+2.0*(ky2+ky3)+ky4)/6.
    vxn = vx0+(kvx1+2.0*(kvx2+kvx3)+kvx4)/6.
    vyn =vy0+(kvy1+2.0*(kvy2+kvy3)+kvy4)/6.

 ;print,"At t = ", t," xn = ",xn," yn = ",yn," vxn = ",vxn," vyn = ",vyn
    xarray=[[xarray],[xn]]
    yarray=[[yarray],[yn]]
    vxarray=[[vxarray],[vxn]]
    vyarray=[[vyarray],[vyn]]
    x0=xn
    y0=yn
    vx0=vxn
    vy0=vyn
    print, "T = ",t ,"   integration loop ", i
    i +=1
    t+=dt
    tarray=[[tarray],[t]]
    potential=-mu/rad1(x0,y0,mu)^2.-(1.0-mu)/rad2(x0,y0,mu)^2. -.5*(x0^2.+y0^2.)
    parray=[[parray],[potential]]
ENDWHILE
    energy=.5*vxarray^2.+.5*vyarray^2.
    jacobi=-2.0*(parray+energy)
```

```
   momentum= vxarray+vyarray

   angmomentum=vyarray*xarray-vxarray*yarray


  outarray = {ke:energy,pot:parray,jacobi:jacobi, p:momentum, rp:angmomentum,
x:xarray, y:yarray,vx:vxarray, vy:vyarray, t:tarray}


END


PRO lpoints


END


pro visorbit, inx,iny

nx=n_elements(inx)

window,xsize=1000,ysize=1000

plot,inx,iny,xrange=[.2,1],yrange=[-.5,.5],ystyle=1,xstyle=1,psym=3

FOR i=0,nx-1 DO BEGIN

oplot,inx[i],iny[i],psym=3,thick=4

wait=.1

ENDFOR

END


FUNCTION xddot, vy,x,u,r1,r2

term1=2.0*vy+x

term2=-1.0*u*(x-1.0+u)/r1^3.

term3=-1.0*(1.0-u)*(x+u)/r2^3.

return,term1+term2+term3
```

```
END

;Y acceleration
FUNCTION yddot,vx,y,u,r1,r2
term1=y-2.0*vx
term2=-1.0*u*y/r1^3.
term3=-1.0*(1.0-u)*y/r2^3.
return, term1+term2+term3
END



FUNCTION rad1,x,y,u
return, sqrt((x-1.0+u)^2.+y^2.)
END

FUNCTION rad2,x,y,u
return, sqrt((x+u)^2.+y^2.)
END

FUNCTION lpts,a,u
;calculate lagrange points as a function of mu where mu = m1 and m1 < m2
IF a eq 1 THEN BEGIN
alpha=(u/(3d0*(1.0-u)))^(1./3.)
r2=alpha-alpha^2/3d0-alpha^3./9d0-23d0*alpha^3./81d0
return, [(1d0-u)-r2,0d0]
ENDIF
```

```
IF a eq 2 THEN BEGIN
alpha=(u/(3d0*(1.0-u)))^(1./3.)
r2=alpha+alpha^2/3d0-alpha^3./9d0-31d0*alpha^3./81d0
return,[r2+(1d0-u),0d0]
ENDIF

IF a eq 3 THEN BEGIN
rat=u/(1d0-u)
b=-7d0*rat/12d0+7d0*rat^2./12d0-13223d0*rat^3./20736
r2=2d0+b
return,[(1d0-u)-r2,0]
ENDIF

IF a eq 4 THEN BEGIN
Return, [.5d0-u,sqrt(3d0)/2.]
ENDIF
IF a eq 5 THEN BEGIN
Return, [.5d0-u,-1.0*sqrt(3d0)/2.]
ENDIF

END
PRO lagrange,sep,msmall,mbig, values

 ;a is separation between m1 and m2 in m
 a=1.49598d11
 a=sep*a
 ms=1.9891d30
```

```
;m2>m1 and on negative x for this formulation to work
m1=msmall*ms
m2=mbig*ms
;radial separation from CM_system
m=m2/(m1+m2)
r1=m*a
r2=(m-1.0)*a
bounds=[r1,r2]
kmin=min(bounds)*1.5
kmax=max(bounds)*1.5
kmin=-2.0*a
kmax=2.0*a
;gravitational constant
G=-6.67428d-11
norm=G*(m1+m2)/a
;array size-keep this even so we kill zeros
n=1000
;make kx and ky and r array where r is distance from center of mass
k = dindgen(n)*(kmax-kmin)/(n-1.)+kmin
kx =k#make_array(n,value=1.0)
ky =make_array(n,value=1.0)#k
kmag2xy = kx*kx+ky*ky
r=sqrt(kmag2xy)
;need s1,s2 = |r-r1|, |r-r2|
;s2=sqrt((kx-r2)^2.+ky^2.)
;s1=sqrt((kx-r1)^2.+ky^2.)
s2=sqrt(r2^2.+r^2.0-2.0*r2*kx)
```

```
  s1=sqrt(r1^2.+r^2.0-2.0*r1*kx)

  pot1= G*m1/s1

  pot2= G*m2/s2

  pot3= G*.5*(m1+m2)*r^2./a^3.

  potential=pot1+pot2+pot3

 potnorm=potential/(-1.0*norm)

dx=abs(kx[2]-kx[1])

dy=dx

grad_x=(SHIFT(potential, -1, 0) - SHIFT(potential, 1, 0))/(2 * dx)

grad_y=(SHIFT(potential, 0, -1) - SHIFT(potential, 0, 1))/(2 * dy)

 values =
{r1:r1,r2:r2,kx:kx,ky:ky,kmag2xy:kmag2xy,pot:potential,potnorm:potnorm,m1:m1,
m2:m2, a:a, grad_x:grad_x, grad_y:grad_y}


END


PRO l4orbit, t,x,y

;returns x and y orbit of a particle librating about l4

x=(3.54d-4)*sin(.268*t)-(9.85d-5)*sin(.963*t)

y=(6.23d-5)*cos(.268*t)-(4.86d-5)*cos(.963*t)

END


PRO orbits,x,y,vx,vy,mu,dt,tstop,outarray

;initialize arrays

;l4/l5 y=+-sqrt(3.0)/2. x=.5-mu


  i=0L

  dt=double(dt)
```

```
  t=dt
  tarray=t
  mu=double(mu)
  x0=double(x)
  y0=double(y)
  vx0=double(vx)
  vy0=double(vy)



  xarray=x0
  yarray=y0
  vxarray=vx0
  vyarray=vy0
  parray=-mu/rad1(x0,y0,mu)^2.-(1.0-mu)/rad2(x0,y0,mu)^2. -.5*(x0^2.+y0^2.)

;m1<m2
;mu = M1
;1-mu = M2
;M2+M1=1
;r1=1-mu
;r2=-mu
;G=omega=1


;MAIN INTEGRATION LOOP
  WHILE t lt tstop  DO BEGIN
```

```
 ;print,"At t = ", t," x0 = ",x0," y0 = ",y0," vx0 = ",vx0," vy0 = ",vy0

;UPDATE x,y,vx,vy - FIRST pass - x1=vx0*dt, v1=a0*dt

    kx1=vx0*dt ;x1

    ky1=vy0*dt ;y1

    kvx1=xddot(vy0,x0,mu,rad1(x0,y0,mu),rad2(x0,y0,mu))*dt ;vx1

    kvy1=yddot(vx0,y0,mu,rad1(x0,y0,mu),rad2(x0,y0,mu))*dt ;vy1

    ;print,"At t = ", t," kx1 = ",kx1," ky1 = ",ky1," kvx1 = ",kvx1," kvy1 = ",kvy1


;UPDATE x,y,vx,vy - SECOND pass, use at x0+k1/2 so new x is x0+vx0dt, new v is
vx0+a*dt

    kx2=(vx0+kvx1/2.)*dt ;x2

    ky2=(vy0+kvy1/2.)*dt ;y2

kvx2=xddot(vy0+kvy1/2.,x0+kx1/2.,mu,rad1(x0+kx1/2.,y0+ky1/2.,mu),rad2(x0+kx
1/2.,y0+ky1/2.,mu))*dt ;vx2

kvy2=yddot(vx0+kvx1/2.,y0+ky1/2.,mu,rad1(x0+kx1/2.,y0+ky1/2.,mu),rad2(x0+kx
1/2.,y0+ky1/2.,mu))*dt ;vy2

    ;print,"At t = ", t," kx2 = ",kx2," ky2 = ",ky2," kvx2 = ",kvx2," kvy2 = ",kvy2


;UPDATE x,y,vx,vy - THIRD pass

    kx3=(vx0+kvx2/2.)*dt ;x3

    ky3=(vy0+kvy2/2.)*dt ;y3

kvx3=xddot(vy0+kvy2/2.,x0+kx2/2.,mu,rad1(x0+kx2/2.,y0+ky2/2.,mu),rad2(x0+kx
2/2.,y0+ky2/2.,mu))*dt ;vx3

kvy3=yddot(vx0+kvx2/2.,y0+ky2/2.,mu,rad1(x0+kx2/2.,y0+ky2/2.,mu),rad2(x0+kx
2/2.,y0+ky2/2.,mu))*dt ;vy3

    ;print,"At t = ", t," kx3 = ",kx3," ky3 = ",ky3," kvx3 = ",kvx3," kvy3 = ",kvy3
```

```
;UPDATE x,y,vx,vy - FINAL pass
    kx4=(vx0+kvx3)*dt ;x4
    ky4=(vy0+kvy3)*dt ;y4

kvx4=xddot(vy0+kvy3,x0+kx3,mu,rad1(x0+kx3,y0+ky3,mu),rad2(x0+kx3,y0+ky3,mu))*dt ;vx4

kvy4=yddot(vx0+kvx3,y0+ky3,mu,rad1(x0+kx3,y0+ky3,mu),rad2(x0+kx3,y0+ky3,mu))*dt ;vy4


    ;print,"At t = ", t," kx4 = ",kx4," ky4 = ",ky4," kvx4 = ",kvx4," kvy4 = ",kvy4


;UPDATE x0,y0, vx0, vy0 to xn,yn,vxn,vyn
    xn = x0+(kx1+2.0*(kx2+kx3)+kx4)/6.
    yn = y0+(ky1+2.0*(ky2+ky3)+ky4)/6.
    vxn = vx0+(kvx1+2.0*(kvx2+kvx3)+kvx4)/6.
    vyn =vy0+(kvy1+2.0*(kvy2+kvy3)+kvy4)/6.


 ;print,"At t = ", t," xn = ",xn," yn = ",yn," vxn = ",vxn," vyn = ",vyn
    xarray=[[xarray],[xn]]
    yarray=[[yarray],[yn]]
    vxarray=[[vxarray],[vxn]]
    vyarray=[[vyarray],[vyn]]
    x0=xn
    y0=yn
    vx0=vxn
    vy0=vyn
    print, "T = ",t ,"   integration loop ", i
```

```
   i +=1
   t+=dt
   tarray=[[tarray],[t]]
   potential=-mu/rad1(x0,y0,mu)^2.-(1.0-mu)/rad2(x0,y0,mu)^2. -.5*(x0^2.+y0^2.)
   parray=[[parray],[potential]]
ENDWHILE
   energy=.5*vxarray^2.+.5*vyarray^2.
   jacobi=-2.0*(parray+energy)
   momentum= vxarray+vyarray
   angmomentum=vyarray*xarray-vxarray*yarray

  outarray = {ke:energy,pot:parray,jacobi:jacobi, p:momentum, rp:angmomentum,
x:xarray, y:yarray,vx:vxarray, vy:vyarray, t:tarray}

END

PRO lpoints

END
```

PENDULUM WITH SPRING EXAMPLE

```
FUNCTION lddot, l, theta, thetadot, k, m, b
term1 = (l+b)*thetadot^2.
term2=9.8*cos(theta)
term3 = -k*l/m
```

```
return, term1 + term2 + term3
END

FUNCTION thetaddot,theta,thetadot,l,ldot,b
term1=-1.*(9.8*sin(theta)+2*ldot*thetadot)
term2=l+b
return, term1/term2
END



PRO ch7plot2,l,theta,ldot,thetadot,kin,min,bin,dt,tstop,outarray
;initialize arrays

 i=0L
 dt=double(dt)
 t=dt
 tarray=t
 k=double(kin)
 m=double(min)
 b=double(bin)
 x0=double(l)
 y0=double(theta)
 vx0=double(ldot)
 vy0=double(thetadot)
```

```
  xarray=x0
  yarray=y0
  vxarray=vx0
  vyarray=vy0

;MAIN INTEGRATION LOOP
  WHILE t lt tstop  DO BEGIN

 ;print,"At t = ", t," x0 = ",x0," y0 = ",y0," vx0 = ",vx0," vy0 = ",vy0
;UPDATE x,y,vx,vy - FIRST pass - x1=vx0*dt, v1=a0*dt
    kx1=vx0*dt ;x1
    ky1=vy0*dt ;y1
    kvx1=lddot(x0, y0, vy0, k, m,b)*dt ;vx1
    kvy1=thetaddot(y0,vy0,x0,vx0,b)*dt ;vy1
    ;print,"At t = ", t," kx1 = ",kx1," ky1 = ",ky1," kvx1 = ",kvx1," kvy1 = ",kvy1

;UPDATE x,y,vx,vy - SECOND pass, use at x0+k1/2 so new x is x0+vx0dt, new v is
vx0+a*dt
    kx2=(vx0+kvx1/2.)*dt ;x2
    ky2=(vy0+kvy1/2.)*dt ;y2
    kvx2=lddot(x0+kx1/2.,y0+ky1/2.,vy0+kvy1/2.,k,m,b)*dt ;vx2
    kvy2=thetaddot(y0+ky1/2.,vy0+kvy1/2.,x0+kx1/2.,vx0+kvx1/2.,b)*dt;vy2
    ;print,"At t = ", t," kx2 = ",kx2," ky2 = ",ky2," kvx2 = ",kvx2," kvy2 = ",kvy2

;UPDATE x,y,vx,vy - THIRD pass
    kx3=(vx0+kvx2/2.)*dt ;x3
    ky3=(vy0+kvy2/2.)*dt ;y3
```

```
    kvx3=lddot(x0+kx2/2.,y0+ky2/2.,vy0+kvy2/2.,k,m,b)*dt ;vx3
    kvy3=thetaddot(y0+ky2/2.,vy0+kvy2/2.,x0+kx2/2.,vx0+kvx2/2.,b)*dt;vy3
    ;print,"At t = ", t," kx3 = ",kx3," ky3 = ",ky3," kvx3 = ",kvx3," kvy3 = ",kvy3

;UPDATE x,y,vx,vy - FINAL pass
    kx4=(vx0+kvx3)*dt ;x4
    ky4=(vy0+kvy3)*dt ;y4
    kvx4=lddot(x0+kx3,y0+ky3,vy0+kvy3,k,m,b)*dt ;vx4
    kvy4=thetaddot(y0+ky3,vy0+kvy3,x0+kx3,vx0+kvx3,b)*dt;vy3
    ;print,"At t = ", t," kx4 = ",kx4," ky4 = ",ky4," kvx4 = ",kvx4," kvy4 = ",kvy4

;UPDATE x0,y0, vx0, vy0 to xn,yn,vxn,vyn
    xn = x0+(kx1+2.0*(kx2+kx3)+kx4)/6.
    yn = y0+(ky1+2.0*(ky2+ky3)+ky4)/6.
    vxn = vx0+(kvx1+2.0*(kvx2+kvx3)+kvx4)/6.
    vyn =vy0+(kvy1+2.0*(kvy2+kvy3)+kvy4)/6.

 ;print,"At t = ", t," xn = ",xn," yn = ",yn," vxn = ",vxn," vyn = ",vyn
    xarray=[[xarray],[xn]]
    yarray=[[yarray],[yn]]
    vxarray=[[vxarray],[vxn]]
    vyarray=[[vyarray],[vyn]]
    x0=xn
    y0=yn
    vx0=vxn
    vy0=vyn
    print, "T = ",t ,"   integration loop ", i
```

```
   i +=1
   t+=dt
   tarray=[[tarray],[t]]
ENDWHILE
   energy=.5*vxarray^2.+.5*vyarray^2.

  outarray = {ke:energy, x:xarray, y:yarray,vx:vxarray, vy:vyarray, t:tarray}

END
```

SPINNING CYLINDER WITH BEAD EXAMPLE

```
FUNCTION rddot, r,rdot,theta,omega,cons
term1=r*((theta+omega)^2.-2.*9.8*cons)
term2=rdot^2.*4*r*cons^2
term3=1.+4.*cons^2*r^2
return,(term1-term2)/term3
END

FUNCTION thetaddot,thetadot,r,rdot,omega
term1=-2.*rdot*(thetadot+omega)
return, term1/(r+.00001)
END
```

```
PRO ch7plot,r,theta,rdot,thetadot,c,omegaforce,dt,tstop,outarray
;initialize arrays

 i=0L
 dt=double(dt)
 t=dt
 tarray=t
 omega=double(omegaforce)
 x0=double(r)
 y0=double(theta)
 vx0=double(rdot)
 vy0=double(thetadot)
 cons=double(c)



 xarray=x0
 yarray=y0
 vxarray=vx0
 vyarray=vy0

;MAIN INTEGRATION LOOP
 WHILE t lt tstop  DO BEGIN

 ;print,"At t = ", t," x0 = ",x0," y0 = ",y0," vx0 = ",vx0," vy0 = ",vy0
```

```
;UPDATE x,y,vx,vy - FIRST pass - x1=vx0*dt, v1=a0*dt

   kx1=vx0*dt ;x1

   ky1=vy0*dt ;y1

   kvx1=rddot(x0,vx0,y0,omega,cons)*dt ;vx1

   kvy1=thetaddot(vy0,x0,vx0,omega)*dt ;vy1

   ;print,"At t = ", t," kx1 = ",kx1," ky1 = ",ky1," kvx1 = ",kvx1," kvy1 = ",kvy1


;UPDATE x,y,vx,vy - SECOND pass, use at x0+k1/2 so new x is x0+vx0dt, new v is
vx0+a*dt

   kx2=(vx0+kvx1/2.)*dt ;x2

   ky2=(vy0+kvy1/2.)*dt ;y2

   kvx2=rddot(x0+kx1/2.,vx0+kvx1/2.,y0+ky1/2.,omega,cons)*dt ;vx2

   kvy2=thetaddot(vy0+kvy1/2.,x0+kx1/2.,vx0+kvx1/2.,omega)*dt;vy2

   ;print,"At t = ", t," kx2 = ",kx2," ky2 = ",ky2," kvx2 = ",kvx2," kvy2 = ",kvy2


;UPDATE x,y,vx,vy - THIRD pass

   kx3=(vx0+kvx2/2.)*dt ;x3

   ky3=(vy0+kvy2/2.)*dt ;y3

   kvx3=rddot(x0+kx2/2.,vx0+kvx2/2.,y0+ky2/2.,omega,cons)*dt ;vx3

   kvy3=thetaddot(vy0+kvy2/2.,x0+kx2/2.,vx0+kvx2/2.,omega)*dt;vy3

   ;print,"At t = ", t," kx3 = ",kx3," ky3 = ",ky3," kvx3 = ",kvx3," kvy3 = ",kvy3


;UPDATE x,y,vx,vy - FINAL pass

   kx4=(vx0+kvx3)*dt ;x4

   ky4=(vy0+kvy3)*dt ;y4

   kvx4=rddot(x0+kx3,vx0+kvx3,y0+ky3,omega,cons)*dt ;vx4

   kvy4=thetaddot(vy0+kvy3,x0+kx3,vx0+kvx3,omega)*dt;vy4
```

```
   ;print,"At t = ", t," kx4 = ",kx4," ky4 = ",ky4," kvx4 = ",kvx4," kvy4 = ",kvy4


;UPDATE x0,y0, vx0, vy0 to xn,yn,vxn,vyn
   xn = x0+(kx1+2.0*(kx2+kx3)+kx4)/6.
   yn = y0+(ky1+2.0*(ky2+ky3)+ky4)/6.
   vxn = vx0+(kvx1+2.0*(kvx2+kvx3)+kvx4)/6.
   vyn =vy0+(kvy1+2.0*(kvy2+kvy3)+kvy4)/6.


 ;print,"At t = ", t," xn = ",xn," yn = ",yn," vxn = ",vxn," vyn = ",vyn
   xarray=[[xarray],[xn]]
   yarray=[[yarray],[yn]]
   vxarray=[[vxarray],[vxn]]
   vyarray=[[vyarray],[vyn]]
   x0=xn
   y0=yn
   vx0=vxn
   vy0=vyn
   print, "T = ",t ,"   integration loop ", i
   i +=1
   t+=dt
   tarray=[[tarray],[t]]
ENDWHILE
   energy=.5*vxarray^2.+.5*vyarray^2.


  outarray = {ke:energy, x:xarray, y:yarray,vx:vxarray, vy:vyarray, t:tarray}


END
```