

# **CELESTIAL MECHANICS (Fall 2012): COMPUTER EXERCISES III - SOLUTIONS**

(Heikki Salo 26.11.2012)

These example routines for the solution of the exercises can be copied from  
`/wrk/hsalo/TM2012_idl.dir/TM2012_DEMO3.dir`

## 8. NUMERICAL INTEGRATION OF TWO BODY MOTION

### 8.1 Simple example (`inte_simple.pro`)

- A basic main program type procedure for integrating the 2-body motion in Cartesian coordinates, using either I or II order Taylor expansion, or Runge-Kutta 4.
- Initial values are from the analytical orbit defined by the given mean anomaly  $M$  and by the orbital elements  $a, \epsilon$ . Units are set by  $\mu = 1$
- Integration related parameters are the step-size  $dt$ , given relative to orbital period, and the duration of integration  $t_{max}$
- The calculated orbit is plotted every  $ntul$  steps
- The energy and angular momentum (per unit mass) are calculated before and after the integration. Also printed is the relative change of  $h$  and  $|\vec{k}|$  per orbit.
- The function subroutine needed by RK4 procedure is **`inte_simple_func.pro`**. The extra variables (in this case just  $\mu$ ) between the main program and this function are passed via *common* statement

```
*****
; inte_simple.pro
; simple example of numerical integration of 2body orbit
; HS 13.11.2006
*****
; use common to transfer the value of myy into the function
; which evaluates the forces for RK4-procedure

      common any_name,myy

;choice of units
      myy=1.

;initial values
      M=0./!radeg           ;mean anomaly
      a=1.
      eks=0.5

;solve eccentric anomaly from Kepler's equation  (M and E in radians)
      kepler,M,eks,E

;use orbital coordinate system (x-axis points to pericenter)
      b=a*sqrt(1.-eks^2)
      x=a*(cos(E)-eks)
      y=b*sin(E)
      vx=-a*sin(E)*sqrt(myy/a^3)/(1-eks*cos(E))
      vy= b*cos(E)*sqrt(myy/a^3)/(1-eks*cos(E))

;initial energy and angular momentum
      r=sqrt(x^2+y^2)
      h0=0.5*(vx^2+vy^2)-myy/r
      k0=x*vy-y*vx
```

```

;integration parameters
period=2.*!dpi*sqrt(myy/a^3)
dt=0.01d0*period
tmax=10.*period
nsteps=tmax/dt
ntul=1          ;output every ntul steps

;choice of the integration method
choice=1        ;Taylor I
choice=2        ;Taylor2
; choice=0      ;RK4

;draw window and define plot region
nwin
plot,x*[1,1],y*[1,1],/iso,xrange=[-3,3],yrange=[-3,3]

;integration loop
time=0.
for i=0l,nsteps-1 do begin

    if(choice eq 1) then begin
        r=sqrt(x^2+y^2)
        ax=-myy/r^3*x
        ay=-myy/r^3*y
        x=x+vx*dt
        y=y+vy*dt
        vx=vx+ax*dt
        vy=vy+ay*dt
        time=time+dt
    endif

    if(choice eq 2) then begin
        r=sqrt(x^2+y^2)
        r2=r^2
        r3=r^3
        r_dot_v=x*vx+y*vy
        apu=3.*r_dot_v/r2
        dt2=0.5*dt^2
        ax=-myy/r3*x
        ay=-myy/r3*y
        aax=-myy/r3*(vx-apu*x)
        aay=-myy/r3*(vy-apu*y)
        x=x+vx*dt+ax*dt2
        y=y+vy*dt+ay*dt2
        vx=vx+ax*dt+aax*dt2
        vy=vy+ay*dt+aay*dt2
        time=time+dt
    endif

    if(choice eq 0) then begin
        yy=[x,y,vx,vy]
        dydx=inte_simple_func(time,yy)
        res=rk4(yy,dydx,time,dt,'inte_simple_func',/double)
        time=time+dt
        x=res(0) & y=res(1) & vx=res(2) & vy=res(3)
    endif

;plot?
    if(i mod ntul eq 0) then oplot,x*[1,1],y*[1,1],psym=3

endfor

```

```

;final energy and angular momentum
r=sqrt(x^2+y^2)
h1=0.5*(vx^2+vy^2)-myy/r
k1=x*vy-y*vx

if(choice eq 1) then method='Taylor I '
if(choice eq 2) then method='Taylor II'
if(choice eq 0) then method='RK4      '

xyouts,0.5,0.9,/normal,ali=0.5,'method='+string(method)+' dt/PER='+string(dt/period)
print,'method=',method,' dt/PER=',dt/period
print,'initial h and k:',h0,k0
print,'final   h and k:',h1,k1

nper=tmax/period
dh=(h1-h0)/abs(h0)/nper
dk=(k1-k0)/abs(k0)/nper
print,'dh/|h| and dk/|k| per orbit',dh,dk
end

```

### Function subroutine in **inte\_simple\_func.pro**

```

;*****
;function inte_simple_func,t,yy
;*****
;non-perturbed 2-body integration
;in cartesian coordinates
;called by inte_simple

common any_name,myy

x=yy(0)
y=yy(1)
vx=yy(2)
vy=yy(3)
dydx=yy*0.

r2=x^2+y^2
r3=r2^1.5
fx=-myy*x/r3
fy=-myy*y/r3

dydx(0)=vx
dydx(1)=vy
dydx(2)=fx
dydx(3)=fy
return,dydx
end

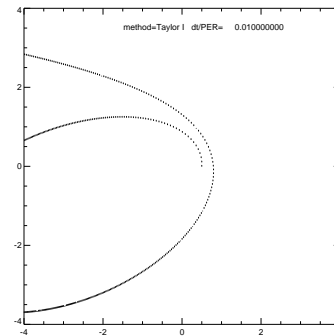
```

- There is also a subroutine version named **inte\_simple\_f.pro**. This uses as defaults: eks=0.5, RK4, dt=0.01 period. Each of these can be changed with a corresponding keyword

Example `inte_simple_demo.pro`, which calls `inte_simple.f.pro`

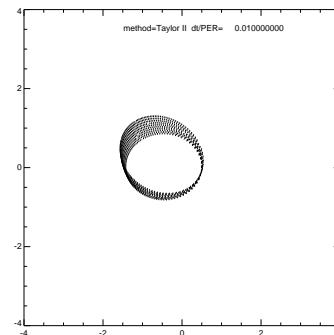
```
psdirect,program+'_1',ps,/color
inte_simple_f,eks=0.5,dt=0.01,choice=1
psdirect,program+'_1',ps,/color,/stop
```

```
method=Taylor I   dt/PER=    0.010000000
initial h and k:  -0.50000005    0.86602539
final  h and k:    0.067630238    1.3669949
dh/|h| and dk/|k| per orbit    0.11352605    0.057846982
```



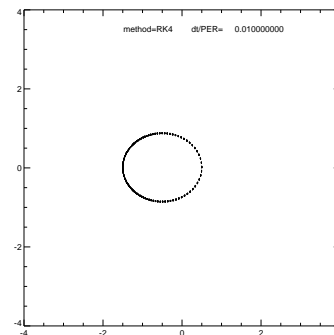
```
psdirect,program+'_2',ps,/color
inte_simple_f,eks=0.5,dt=0.01,choice=2
psdirect,program+'_2',ps,/color,/stop
```

```
method=Taylor II dt/PER=    0.010000000
initial h and k:  -0.50000005    0.86602539
final  h and k:   -0.45505289    0.87362013
dh/|h| and dk/|k| per orbit    0.0089894314    0.00087696563
IDL> psclose
```



```
psdirect,program+'_3',ps,/color
inte_simple_f,eks=0.5,dt=0.01
psdirect,program+'_3',ps,/color,/stop
```

```
method=RK4        dt/PER=    0.010000000
initial h and k:  -0.50000005    0.86602539
final  h and k:   -0.50014636    0.86598613
dh/|h| and dk/|k| per orbit -2.9261253e-05 -4.5326194e-06
```



## 8.2 More sophisticated example (rk\_intel.pro)

- subroutine type program, with a lot of options.
- Taylor series integration up to 6th degree, or RK4

```

;*****
; rk_intel.pro
;*****
;Integrate 2-body equations with Runge-Kutta 4 (RK4)
;check the accuracy of the orbital elements
;OCT 2002/2006 Celestial mechanics (HS) -->
;*****

pro rk_intel,elem,t1_0,t2_0,dt_0,output0=output0,myy0=myy0,example=example,$
    plot=plot,cplot=cplot,wid0=wid0,oplot=oplot,dl=dl,de=de,$
    t_out=t_out,l_out=l_out,e_out=e_out,taylor=taylor,title0=title0,$
    dtorb=dtorb,silent=silent,connect=connect,$
    x_out=x_out,y_out=y_out,z_out=z_out,$
    vx_out=vx_out,vy_out=vy_out,vz_out=vz_out

if(n_params() le 0) then begin
    print,'-----'
    print,'pro rk_intel,elem,t1,t2,dt,output=output,plot=plot'
    print,'-----'
    print,'Cartesian integration of non-perturbed 2-body orbit HS 20.11.02/12.02.06'
    print,' elem=[a,e,i,ome,w,tau] initial orbital elements'
    print,' t1,t2 integration time interval (in orbital periods)'
    print,' dt time step (orbital periods)'
    print,'KEYWORDS:'
    print,'-----'
    print,' myy G * (m1+m2) def=1.'
    print,' taylor=choice use Taylor series, with degree=taylor (def=rk4)'
    print,' choices= 1, 2 explicitly written'
    print,' -1,-2,-3,-4,-5,-6 using f,g-series'
    print,' dtorb =value -> dt=period/dtorb'
    print,'-----'
    print,'PLOTTING KEYWORDS:'
    print,' plot=istep plot every istep steps (def=no plot)'
    print,' oplot=color plot on top of previous orbit with'
    print,' color=oplot+2 (i.e 1->col=3=green)'
    print,' /connect -> connect orbit points in the plot (def=no)'
    print,' wid limit of the plot region (DEF=1.25 a)'
    print,' /cplot plot analytic solution (white squares)'
    print,' title -> plot title'
    print,' output=val output interval of ELEM,L,E in steps (def=nsteps/10)'
    print,' val=negative -> just store'
    print,'-----'
    print,'OUTPUT/STORE KEYWORDS:'
    print,' t_out,l_out,e_out dL/L and dE/E vs t_out (stored every |output| step)'
    print,' x_out,y_out,z_out positions'
    print,' vx_out,vy_out,vz_out velocities'
    print,' dl,de return averaged change in dL/L and dE/E /orbit period'
    print,' /silent -> do not print anything to terminal'
    print,'-----'
    print,'EXAMPLE INPUT VALUES:'
    print,' /example example of integration:'
    print,' a=1,ecc=0.5,i=10,ome=90.,w=0,tau=0, t1=0, t2=10*TORB, dt=0.01*TORB'
    print,' rk_intel,elem,t1,t2,dt,/example,/plot'
    print,'-----'
    return
endif

```

```

;to transfer the value of myy into the function
;which evaluates the forces for RK4-procedure
common rk_intel1_com,myy

myy=1.d0
if(keyword_set(myy0)) then myy=myy0*1.d0

if(keyword_set(example)) then begin
    t1_0=0.d0 & t2_0=10.d0 & dt_0=0.01d0
    elem=[1.d0,0.5d0,10.d0,90.d0,0.d0,0.d0]
endif

elem=elem*1.d0
torb=2.d0!*dpi*sqrt(elem(0)^3/myy)
per=torb
t1=t1_0*torb & t2=t2_0*torb & dt=dt_0*torb

if(keyword_set(dtorb)) then dt=torb/dtorb

dt2=dt^2 ;needed in taylor
tapu0=1.d0
tapu1=dt
tapu2=tapu1*dt/2.d0
tapu3=tapu2*dt/3.d0
tapu4=tapu3*dt/4.d0
tapu5=tapu4*dt/5.d0
tapu6=tapu5*dt/6.d0
tapu7=tapu6*dt/7.d0
tapu8=tapu7*dt/8.d0

wid=elem(0)*(1.+elem(1))*1.25
if(keyword_set(wid0)) then wid=wid0

;*****
; R,V at t=t1:
; use the procedure elem_to_rv to calculate
; R and V corresponding to the given orbital elements
;*****
elem_to_rv,elem,t1,rad,vel

;check + obtain constants of integration
rv_to_elem,t1,rad,vel,elem2,kvec=kvec,evec=evec,ene=ene0,imp=imp0

;number of steps, output interval
nsteps=long((t2-t1)/dt)
time=t1
output=long(nsteps/10.)
if(keyword_set(output0)) then output=output0

;arrays for storing constants of motion vs. time
noutput=long(nsteps/abs(output))+1
t_out=dblarr(abs(noutput))
l_out=t_out
e_out=t_out
iout=01

;some formats needed later
ff= '(f10.4,6f12.6,2e14.2)'
ff_ori='(f10.4,6f12.6,e12.2,2h=L,e12.2,2h=E)'

if(not keyword_set(silent)) then begin
    print,'-----',

```

```

print,'CARTESIAN INTEGRATION OF NON-PERTURBED 2-BODY MOTION'
print,'-----'
if(not keyword_set(taylor)) then $
  print,' USING RK4'
  if( keyword_set(taylor)) then $
    print,' USING TAYLOR-SERIES, with degree ',taylor
  print,'MY= G* (m1+m2) = ',myy
  print,'PERIOD = ',per
  print,'T1,T2,DT (in periods) = ',t1/per,t2/per,dt/per
  print,'NSTEPS, OUTPUT = ',nsteps,output
  print,' '
  otsi='      TIME      AMAJ      ECC      INC      OME      W'
  otsi=otsi+'      TAU      dL/L      dE/E'
  print,otsi
  print,time,elem,imp0,ene0,f=ff_ori
endif

istep=01
if(keyword_set(plot)) then begin
  istep=plot
  if(not keyword_set(oplot)) then begin
    title=''
    if(keyword_set(title0)) then title=title0
    nwin
    plot,lindgen(10),xr=[-1,1]*wid,yr=[-1,1]*wid,$
      xs=1,ys=1,/nod,/iso,title=title
    plots,rad(0),rad(1),psym=1
    plots,0,0,psym=1,syms=3
  endif
endif

col=2
if(keyword_set(oplot)) then col=oplot+2

xold=rad(0)
yold=rad(1)

;-----
;MAKE INTEGRATION
;-----

  for i=11,nsteps do begin

;-----
;RK4 INTEGRATION
;-----

    if(not keyword_set(taylor)) then begin
      yy=[rad,vel]
      dydx=rk_inte1_func(time,yy)
      res=rk4(yy,dydx,time,dt,'rk_inte1_func',/double)
      rad=res(0:2)
      vel=res(3:5)
      time=time+dt
    endif

;-----
;TAYLOR SERIES
;-----

    if(keyword_set(taylor)) then begin
      x=rad(0)
      y=rad(1)
      z=rad(2)

```



```

        vx=vel(0)
        vy=vel(1)
        vz=vel(2)
; I degree?
        if(taylor eq 1) then begin
            r2=(x^2+y^2+z^2)
            r3=r2^1.5
            dx=vx
            dy=vy
            dz=vz
            dvx=-myy/r3*x
            dvy=-myy/r3*y
            dvz=-myy/r3*z
            x=x+dx*dt
            y=y+dy*dt
            z=z+dz*dt
            vx=vx+dvx*dt
            vy=vy+dvy*dt
            vz=vz+dvz*dt
            time=time+dt
            rad=[x,y,z]
            vel=[vx,vy,vz]
        endif
; II degree?
        if(taylor eq 2) then begin
            r2=(x^2+y^2+z^2)
            r3=r2^1.5
            rpv=x*vx+y*vy+z*vz
            dvx=-myy/r3*x
            dvy=-myy/r3*y
            dvz=-myy/r3*z
            dddvx=-myy/r3*(vx-3.*rpv/r2*x)
            dddvy=-myy/r3*(vy-3.*rpv/r2*y)
            dddvz=-myy/r3*(vz-3.*rpv/r2*z)
            dx=vx
            dy=vy
            dz=vz
            ddx=dvx
            ddy=dvy
            ddz=dvz
            x=x+dx*dt+0.5*ddx*dt^2
            y=y+dy*dt+0.5*ddy*dt^2
            z=z+dz*dt+0.5*ddz*dt^2
            vx=vx+dvx*dt+0.5*ddvx*dt^2
            vy=vy+dvy*dt+0.5*ddvy*dt^2
            vz=vz+dvz*dt+0.5*ddvz*dt^2
            time=time+dt
            rad=[x,y,z]
            vel=[vx,vy,vz]
        endif

;taylor=negative -use f,g series with order=|taylor|
        if(taylor lt 0) then begin
            nt=-taylor
            r2=x^2+y^2+z^2
            r3=r2^1.5
            rpv=x*vx+y*vy+z*vz

            u=myy/r3
            p=1.d0/r2*rpv
            q=(vx^2+vy^2+vz^2)/r2-u
            up=u*p
            p2=p*p

```

```

q2=q*q
u2=u*u
p3=p2*p

f0=1.d0
f1=0.d0
f2=-u
f3=3.d0*up
f4=u*(-15.*p2+3.d0*q+u)
f5=15.d0*up*(7.d0*p2-3.d0*q-u)

;Bates has an error!
;modified u*p to u*q
;      f6=105.d0*u*p2*(-9.d0*p2+6.d0*q+2.d0*u)-u*(45.d0*q2+24.d0*u*q+u2)
;      f7=315.d0*u*p3*(33.d0*p2-30.d0*q+5.d0*u)+63.d0*up*(25.d0*q2+14.d0*u*q+u2)
;Sconzo

f6=-945.d0*p^4*u+p^2*(630.d0*q*u+210.d0*u^2)$
-24.d0*q*u^2-45.d0*q^2*u-u^3
f7=10395.d0*p^5*u-p^3*(9450.d0*q*u+3150.d0*u^2)+$
p*(882.d0*q*u^2+1575.d0*q^2*u+63.d0*u^3)

g0=0.d0
g1=1.d0
g2=0.d0
g3=-u
g4=6.d0*up
g5=u*(-45.d0*p2+9.d0*q+u)

;Bates has an error!
;      g6=30.d0*up*(14.d0*p2-6.d0*q-u)
;      g7=315.d0*u*p2*(-15.d0*p2+10.d0*q+2.d0*u)-u*(225.d0*q2+54.d0*u*q+u2)
;sconzo

g6=420.d0*p^3*u-p*(180.d0*q*u+30.d0*u^2)
g7=-4725.d0*p^4*u+p^2*(3150.d0*q*u+630.d0*u^2)-$
54.d0*q*u^2-225.d0*q^2*u-u^3

if(nt eq 1) then begin
  fr=f0*tapu0+f1*tapu1
  gr=g0*tapu0+g1*tapu1
  fv=f1*tapu0+f2*tapu1
  gv=g1*tapu0+g2*tapu1
endif
if(nt eq 2) then begin
  fr=f0*tapu0+f1*tapu1+f2*tapu2
  gr=g0*tapu0+g1*tapu1+g2*tapu2
  fv=f1*tapu0+f2*tapu1+f3*tapu2
  gv=g1*tapu0+g2*tapu1+g3*tapu2
endif
if(nt eq 3) then begin
  fr=f0*tapu0+f1*tapu1+f2*tapu2+f3*tapu3
  gr=g0*tapu0+g1*tapu1+g2*tapu2+g3*tapu3
  fv=f1*tapu0+f2*tapu1+f3*tapu2+f4*tapu3
  gv=g1*tapu0+g2*tapu1+g3*tapu2+g4*tapu3
endif
if(nt eq 4) then begin
  fr=f0*tapu0+f1*tapu1+f2*tapu2+f3*tapu3+f4*tapu4
  gr=g0*tapu0+g1*tapu1+g2*tapu2+g3*tapu3+g4*tapu4
  fv=f1*tapu0+f2*tapu1+f3*tapu2+f4*tapu3+f5*tapu4
  gv=g1*tapu0+g2*tapu1+g3*tapu2+g4*tapu3+g5*tapu4
endif
if(nt eq 5) then begin
  fr=f0*tapu0+f1*tapu1+f2*tapu2+f3*tapu3+f4*tapu4+f5*tapu5
  gr=g0*tapu0+g1*tapu1+g2*tapu2+g3*tapu3+g4*tapu4+g5*tapu5
  fv=f1*tapu0+f2*tapu1+f3*tapu2+f4*tapu3+f5*tapu4+f6*tapu5

```

```

        gv=g1*tapu0+g2*tapu1+g3*tapu2+g4*tapu3+g5*tapu4+g6*tapu5
    endif
    if(nt eq 6) then begin
        fr=f0*tapu0+f1*tapu1+f2*tapu2+f3*tapu3+f4*tapu4+f5*tapu5+f6*tapu6
        gr=g0*tapu0+g1*tapu1+g2*tapu2+g3*tapu3+g4*tapu4+g5*tapu5+g6*tapu6
        fv=f1*tapu0+f2*tapu1+f3*tapu2+f4*tapu3+f5*tapu4+f6*tapu5+f7*tapu6
        gv=g1*tapu0+g2*tapu1+g3*tapu2+g4*tapu3+g5*tapu4+g6*tapu5+g7*tapu6
    endif

    xa=fr*x+gr*vx
    ya=fr*y+gr*vy
    za=fr*z+gr*vz
    vx=fv*x+gv*vx
    vy=fv*y+gv*vy
    vz=fv*z+gv*vz
    x=xa
    y=ya
    z=za

    time=time+dt
    rad=[x,y,z]
    vel=[vx,vy,vz]
endif
endif

;-----
; PLOT?
;-----
    if(istep ne 0) then begin
        if(i mod istep eq 0) then begin
            plots,rad(0),rad(1),psym=3,col=col
;connect to previous point?
            if(keyword_set(connect)) then begin
                oplot,[xold,rad(0)],[yold,rad(1)],psym=0,col=col
                xold=rad(0)
                yold=rad(1)
            endif
;overplot analytical?
            if(keyword_set(cplot)) then begin
                elem_to_rv,elem,time,radc,velc
                plots,radc(0),radc(1),psym=4,col=1,syms=.5
            endif
        endif
    endif

;-----
; OUTPUT?
;rv_to_elem calculates orbital elements from R and V
;-----
    if(i mod abs(output) eq 0) then begin
        rv_to_elem,time,rad,vel,elem2,imp=imp,ene=ene
        dimp=(imp-imp0)/imp0
        dene=(ene-ene0)/ene0
        dl=dimp/(time/per) ;change/orbit
        de=dene/(time/per) ;change/orbit
        elem2(5)=elem2(5) mod per ;tau
        if(elem2(5) ge 0.5*per) then elem2(5)=elem2(5)-per
        if(not keyword_set(silent)) then begin
            if(output gt 0 or i eq nsteps) then print,time/per,elem2,dimp,dene,f=ff
        endif
        t_out(iout)=time/per
        e_out(iout)=dene
        l_out(iout)=dimp
    endif

```

```

        iout=iout+1
    endif
endfor
t_out=t_out(0:iout-1)
e_out=e_out(0:iout-1)
l_out=l_out(0:iout-1)

rv_to_elem,time,rad,vel,elem2,imp=imp,ene=ene
dimp=(imp-imp0)/imp0
dene=(ene-ene0)/ene0
dl=dimp/(time/per)      ;change/orbit
de=dene/(time/per)      ;change/orbit
if(not keyword_set(silent)) then begin
    print,'time,dL/L, dE/E (/per orbit)'
    print,time/per,dl,de,f='(f30.16,2g16.6)'
endif
end
end

```

## Function subroutine in **rk\_intel\_func.pro**

```

;*****
function rk_intel_func,t,yy
;*****
;non-perturbed 2-body integration
;in cartesian coordinates
;called by rk_intel

common rk_intel_com,myy

x=yy(0)
y=yy(1)
z=yy(2)
vx=yy(3)
vy=yy(4)
vz=yy(5)
dydx=yy*0.

r2=x^2+y^2+z^2
r3=r2^1.5
fx=-myy*x/r3
fy=-myy*y/r3
fz=-myy*z/r3

dydx(0)=vx
dydx(1)=vy
dydx(2)=vz
dydx(3)=fx
dydx(4)=fy
dydx(5)=fz

return,dydx
end

```

Examples:

- See the separate document **rk\_intel\_play.pdf** for a demonstration ( To use: open the document by pdf-viewer, drag with mouse the idl commands indicated by red color to the IDL window. Read the comments + compare to the produced plots.)

- Here - basic usage:

/example --> example of integration:

a=1,ecc=0.5,i=10,ome=90.,w=0,tau=0, t1=0, t2=10\*TORB, dt=0.01\*TORB

IDL> rk\_intel,elem,t1,t2,dt,/example,/plot,title='example'

-----  
CARTESIAN INTEGRATION OF NON-PERTURBED 2-BODY MOTION  
-----

USING RK4

MY= G\* (m1+m2) = 1.0000000

PERIOD = 6.2831853

T1,T2,DT (in periods) = 0.0000000 10.000000 0.01000000

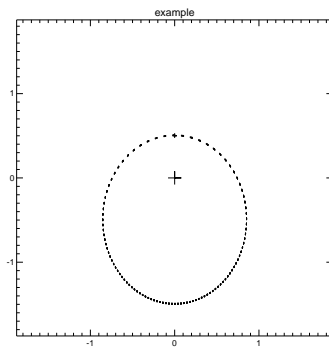
NSTEPS, OUTPUT = 999 99

TIME	AMAJ	ECC	INC	OME	W	TAU	dL/L	dE/E
0.0000	1.000000	0.500000	10.000000	90.000000	0.000000	0.000000	8.66e-01=L	-5.00e-01=E
0.9900	0.999973	0.499986	9.999999	89.999995	0.003638	-0.000298	-4.01e-06	2.73e-05
1.9800	0.999943	0.499969	9.999999	89.999995	0.007464	0.000208	-8.11e-06	5.73e-05
2.9700	0.999912	0.499952	9.999999	89.999995	0.011529	0.000774	-1.23e-05	8.83e-05
3.9600	0.999881	0.499936	9.999999	89.999995	0.015775	0.001642	-1.67e-05	1.19e-04
4.9500	0.999851	0.499920	9.999999	89.999995	0.020120	0.002791	-2.11e-05	1.49e-04
5.9400	0.999821	0.499904	9.999999	89.999995	0.024511	0.004213	-2.56e-05	1.79e-04
6.9300	0.999791	0.499888	9.999999	89.999995	0.028924	0.005909	-3.01e-05	2.09e-04
7.9200	0.999761	0.499873	9.999999	89.999995	0.033350	0.007880	-3.46e-05	2.39e-04
8.9100	0.999732	0.499857	9.999999	89.999995	0.037784	0.010129	-3.92e-05	2.68e-04
9.9000	0.999702	0.499842	9.999999	89.999995	0.042225	0.012654	-4.37e-05	2.98e-04
time,dL/L, dE/E (/per orbit)								
	9.9900000000002240		-4.53715e-06		2.92905e-05			

Try with different time steps (leave away /example)

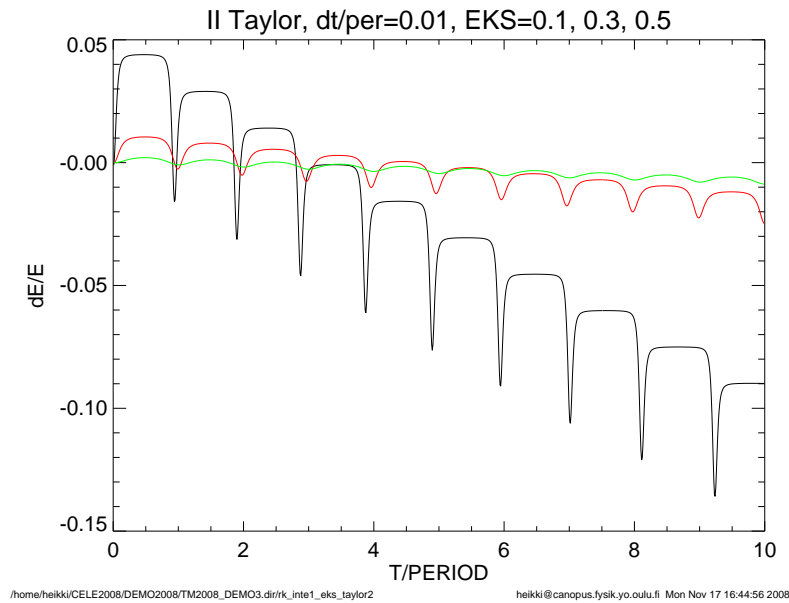
IDL> rk\_intel,elem,t1,t2,0.001,/plot,title='example dt=0.001'

IDL> rk\_intel,elem,t1,t2,0.0001,/plot,title='example dt=0.0001'

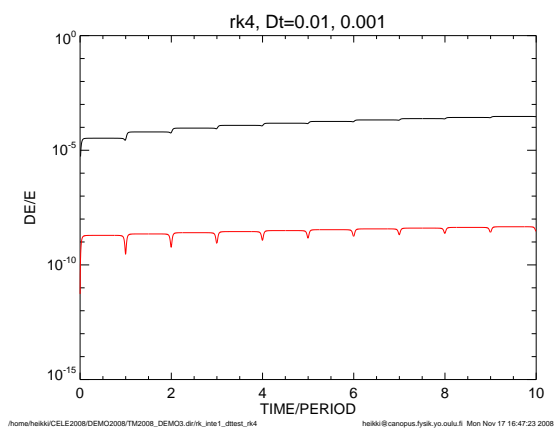


How does the error in energy depends on eccentricity, timestep, and the order of integration?

**rk\_inte1\_eks\_taylor2.pro** Comparison for eks=0.1,0.3,0.5, using II degree Taylor

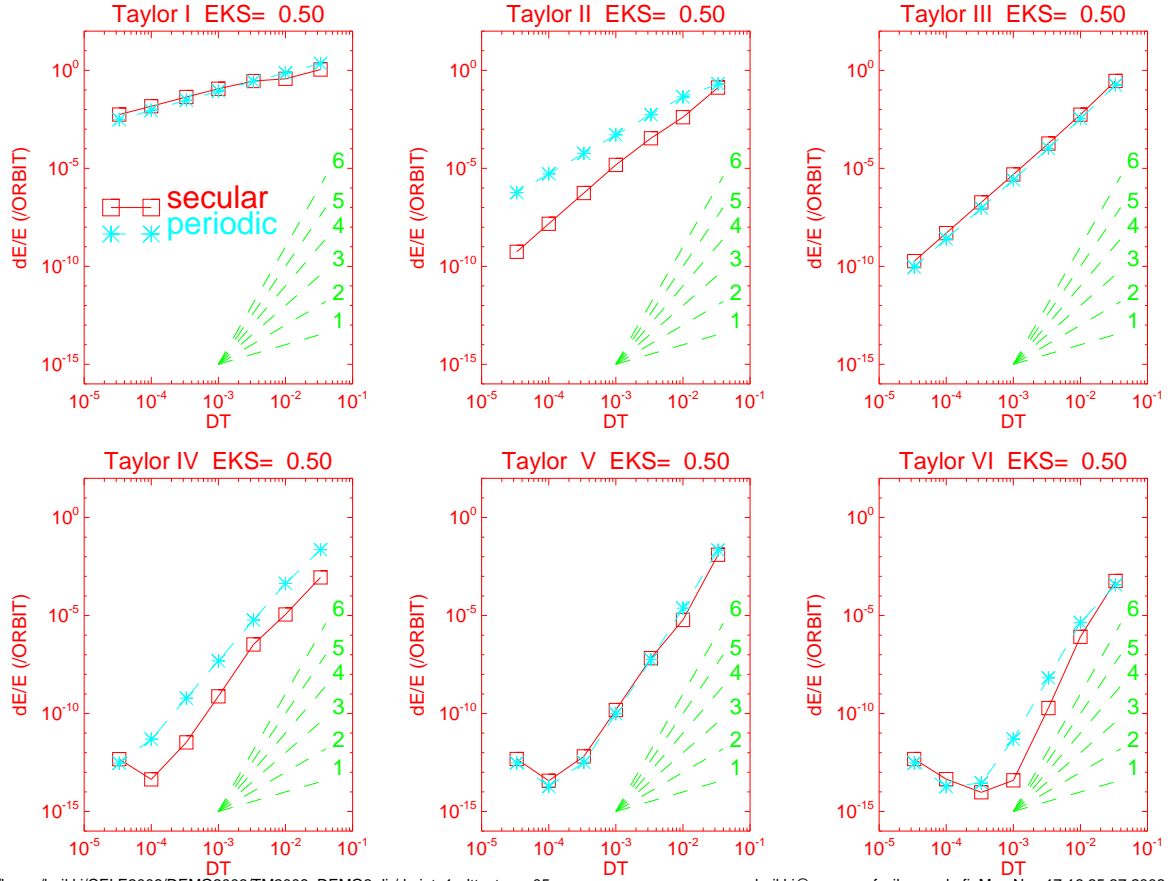


**rk\_inte1\_dttest\_rk4.pro** Check the energy error vs. time, with RK4 and two different time steps  $dt=0.01$  and  $0.001$  periods (eks=0.5)



# rk\_inte1\_dttest.pro

The secular and maximum periodic errors are shown separately for the case  $\epsilon = 0.5$ .  
DT gives the timestep relative to orbital period



### 8.3 PERTURBED TWO BODY MOTION (inte\_perp.pro)

• Add perturbation to the relative 2-body force. Perturbation has the form used in the lectures:

$$\vec{F} = f_r \vec{R}/r + f_v \vec{V}/v + f_n \vec{N}$$

$$f_r = \alpha \mu/a^2$$

$$f_v = \beta \mu/a^2$$

$$f_n = \gamma \mu/a^2$$

• The following subroutine uses rk4 to integrate the perturbed motion. Initial values are defined via unperturbed orbital elements. Perturbation defined via keywords **alpha**, **beta**, **gamma**. As a default they affect continuously: by the keywords **E1**, **E2** one can specify an interval during which the perturbations affect the orbit.

```

;*****
; inte_perp.pro
;*****
;Integrate 2-body equations with Runge-Kutta 4 method
;add perturbations F = alfa * R/r + beta * V/v + gamma * N
; at the directions of R/r, V/v, N
; at the specific E-intervals
;For the course of celestial mechanics, Fall 2002/2006 (HS)

pro inte_perp,elem,t1,t2,dt,output0=output0,myy0=myy0,example=example,$
    e10=e10,e20=e20,alfa0=alfa0,beta0=beta0,gamma0=gamma0,fimp=fimp,$
    save0=save0,outsave0=outsave0,fric=fric,final_elem=final_elem,$
    orb=orb,rtable=rtable,ttable=ttable,silent=silent

if(n_params() le 0) then begin
    print,'pro inte_perp,elem,t1,t2,dt,output=output,myy=myy,orb=orb'
    print,' '
    print,'Cartesian integration of perturbed 2-body orbit'
    print,'elem=[a,e,i,ome,w,tau]  initial orbital elements'
    print,'t1,t2  integration time interval'
    print,'dt  time step'
    print,'KEYWORDS:'
    print,'final_elem  returns final table of orbital elements'
    print,'ttable,rtable  returns,t(*),r(*,3)'
    print,'output  output interval in steps (def=nsteps/10)'
    print,'myy  G * (m1+m2)  def=1.'
    print,'/orb  -> t1,t2,dt given in orbital periods'
    print,'/example  example of integration:'
    print,'  a=1,ecc=0.5,i=10,ome=90,w=0,tau=0'
    print,'  t1=0, t2=10*PER, dt=0.01*PER,'
    print,'  (inserted to elem)'
    print,'inte_perp,elem,t1,t2,dt,/example'
    print,' '
    print,'save=filename ->  save ttable,rtable,vtable,elemtable'
    print,'  default='inte_perp.save''
    print,'outsave ->  store every outsave steps (def=10)'
    print,' '
    print,'PERTURBATION PARAMETERS'

```



```

        print,'alfa,beta,gamma: F = alfa * R/r + beta * V/v + gamma * N'
        print,'E1,E2: perturbation affects on the ecc anom interval:'
        print,'                                E1 - E2 (degrees)'
        print,'/fimp -> scale perturbation with 1/DE (impulsive perturbation)'
        print,' '
        print,'-----'
        return
    endif

;*****
; common inte_perp_com,myy,alfa,beta,gamma
; ra=180.d0/!dpi
; ar=!dpi/180.d0

;IDL save-file to store the integration results
; savefile='inte_perp.save'
; if(keyword_set(save0)) then savefile=save0

;storing interval
; outsave=10
; if(keyword_set(outsave0)) then outsave=outsave0

myy=1.d0
if(keyword_set(myy0)) then myy=myy0

if(keyword_set(example)) then begin
    t1=0.d0
    t2=10*2.d0*!dpi
    dt=0.01d0*2.*!dpi
    elem=[1.0, 0.5, 10., 90., 0., 0.]*1.d0
endif

t1=t1*1.d0
t2=t2*1.d0
dt=dt*1.d0
torb=2.d0*!dpi*sqrt(elem(0)^3/myy)
per=torb
eks=elem(1)
tau=elem(5)
if(keyword_set(orb)) then begin
    t1=t1*torb
    t2=t2*torb
    dt=dt*torb
endif

;*****
; perturbation: normalized to central force at r=a0 orbit
; E-interval from E1,E2: alfa=alfa1, otherwise zero
; (same with beta,gamma)
; e1,e2 in degrees
; alfa1,beta1,gamma1 = input values
; alfa ,beta ,gamma = current values
;*****

fac=myy/elem(0)^2
e1=-1.d6
e2= 1.d6
alfa1=0.d0
beta1=0.d0
gamma1=0.d0
if(keyword_set(alfa0)) then alfa1=alfa0*1.d0*fac
if(keyword_set(beta0)) then beta1=beta0*1.d0*fac
if(keyword_set(gamma0)) then gamma1=gamma0*1.d0*fac

```

```

    alfa=alfal
    beta=betal
    gamma=gammal
    if(keyword_set(e10)) then e1=e10*1.d0
    if(keyword_set(e20)) then e2=e20*1.d0
    if(keyword_set(fimp)) then fac=fac/(e2-e1)*360.

;eflag = 1 if eccentric anomaly needs to be calculated:
; this is the case if E-interval defined
; AND perturbation is added
    eflag=0
    if(keyword_set(e10) or keyword_set(e20)) then begin
        if(alfal ne 0) then eflag=1
        if(betal ne 0) then eflag=1
        if(gammal ne 0) then eflag=1
    endif

;*****
; original R,V
;*****
    elem_to_rv,elem,t1,rad,vel
    rv_to_elem,t1,rad,vel,elem2,kvec=kvec,evec=evec,ene=ene0,imp=imp0

    nsteps=long((t2-t1)/dt)
    time=t1
    output=long(nsteps/10.)
    if(keyword_set(output0)) then output=output0

    if(not keyword_set(silent)) then begin
        print,'-----'
        print,'CARTESIAN RK4 INTEGRATION OF PERTURBED 2-BODY MOTION'
        print,'-----'
        print,'MY= G* (m1+m2) = ',myy
        print,'PERIOD = ',per
        print,'T1,T2,DT (in periods) = ',t1/per,t2/per,dt/per
        print,'NSTEPS, OUTPUT = ',nsteps,output
        print,'ALFA, BETA, GAMMA = ',alfal,betal,gammal
        print,'E1,E2 (degrees)',e1,e2
        print,' '
        ff='(f10.4,6f12.6,2e12.2)'
        otsi='      T/PER      AMAJ      ECC      INC      OME      W'
        otsi=otsi+'      TAU      dL/L      dE/E'
        print,otsi
        print,time,elem,imp0,ene0,f=ff
    endif
    elem0=elem

;*****
; for saving results
    nsave=nsteps/outsave+2
    ttable=dblarr(nsave)
    rtable=dblarr(nsave,3)
    vtable=dblarr(nsave,3)
    elemtable=dblarr(nsave,6)
    enetable=dblarr(nsave,2)
    ttable(0)=t1
    rtable(0,0:2)=rad
    vtable(0,0:2)=vel
    elemtable(0,0:5)=elem
    isave=0

;*****
;integration

```

```

;*****
;
; for i=01,nsteps do begin
; check if perturbation is in effect
; if(eflag eq 1) then begin
;     elem_to_rv,elem,time,rad0,vel0,eano=e
;     alfa=0.d0
;     beta=0.d0
;     gamma=0.d0
;     if(e ge e1 and e le e2) then begin
;         alfa=alfa1*fac
;         beta=beta1*fac
;         gamma=gamma1*fac
;     endif
; endif
; yy=[rad,vel]
; dydx=inte_perp_func(time,yy)
; res=rk4(yy,dydx,time,dt,'inte_perp_func',/double)
; rad=res(0:2)
; vel=res(3:5)
; time=time+dt
;
; output?
; if(not keyword_set(silent)) then begin
;     if(i+1 eq ((i+1)/output)*output) then begin
;         rv_to_elem,time,rad,vel,elem2,imp=imp,ene=ene
;         dimp=(imp-imp0)/imp0
;         dene=(ene-ene0)/ene0
;         print,time/per,elem2,dimp,dene,f=ff
;     endif
; endif
;
; save?
; if(i+1 eq ((i+1)/outsave)*outsave) then begin
;     rv_to_elem,time,rad,vel,elem2,imp=imp,ene=ene
;     isave=isave+1
;     ttable(isave)=time
;     rtable(isave,0:2)=rad
;     vtable(isave,0:2)=vel
;     elemtable(isave,0:5)=elem2
;     enetable(isave,0:1)=[imp,ene]
; endif
; endfor
;
; rv_to_elem,time,rad,vel,final_elem,imp=imp,ene=ene
;
; ttable=ttable(0:isave)
; elemtable=elemtable(0:isave,*)
; rtable=rtable(0:isave,*)
; vtable=vtable(0:isave,*)
; fiitable=atan(rtable(*,1),rtable(*,0))*ra
; radtable=sqrt(rtable(*,0)^2+rtable(*,1)^2)
;
; save,file=savefile,t1,t2,dt,myy,per,nsteps,output,$
;     alfa1,beta1,gamma1,e1,e2,elem,final_elem,$
;     ttable,rtable,elemtable,vtable,enetable,fiitable,radtable
;
; if(not keyword_set(silent)) then begin
;     print,'to restore saved data:'
;     print,'restore,'"'+savefile+'"'
; endif
; end

```

```

;*****
;
;   function inte_perp_func,t,yy
;*****
; perturbed 2-body integration using cartesian coordinates
; perturbing force:  $F = \alpha \cdot R/r + \beta \cdot V/v + \gamma \cdot N$ 
; called by inte_perp.pro (HS 07.10.02/12.11.06)
;*****
;   common inte_perp_com,myy,alfa,beta,gamma

;   x=yy(0) & y=yy(1) & z=yy(2) & vx=yy(3) & vy=yy(4) & vz=yy(5)
;   dydx=yy*0.
;   r2=x^2+y^2+z^2
;   r3=r2^1.5d0
;   r=sqrt(r2)

;central force
;   fx=-myy*x/r3
;   fy=-myy*y/r3
;   fz=-myy*z/r3

;perturbation ?
;   if(alfa ne 0.) then begin
;       fx=fx+alfa*x/r
;       fy=fy+alfa*y/r
;       fz=fz+alfa*z/r
;   endif
;   if(beta ne 0.) then begin
;       v=sqrt(vx^2+vy^2+vz^2)
;       fx=fx+beta*vx/v
;       fy=fy+beta*vy/v
;       fz=fz+beta*vz/v
;   endif
;   if(gamma ne 0.) then begin
;       cross_product,[x,y,z],[vx,vy,vz],n
;       nn=sqrt(n(0)^2+n(1)^2+n(2)^2)
;       nx=n(0)/nn & ny=n(1)/nn & nz=n(2)/nn
;       fx=fx+gamma*nx
;       fy=fy+gamma*ny
;       fz=fz+gamma*nz
;   endif

;   dydx(0)=vx & dydx(1)=vy & dydx(2)=vz & dydx(3)=fx & dydx(4)=fy & dydx(5)=fz
;   return,dydx
end

```

### Example: **inte\_perp\_demo\_rad.pro**

- The effect of instantaneous radial perturbation, affecting at different parts of the orbit (the same plot was attached to the lectures):

```
;*****
;inte_perp_demo_rad.pro
;apply radial perturbation at different parts of the orbit
;Heikki Salo 2002/05.11.2006
;*****

;elem=[a,e,i,ome,w,tau]
      elem=[1.,.5,1.,0.,0.,0.]
      myy=1.
      cforce=myy/elem(0)^2
      eks=elem(1)
;no perturbation
      inte_perp,elem,0.,1.,.0005,/orb,t=t0,r=r0,/sil,fin=efin0,outsave=1
;solve eccentric anomaly
      per=2.*!pi*sqrt(elem(0)^3/myy)
      m=2.*!pi/per*(t0-elem(5))
      kepler_array,m,elem(1),e
      e=e*!radeq

      xylim,2
      !p.multi=[0,2,2]
      charsize,1.5
      nwin,xs=600,ys=600

;use large perturbation to illustrate the
;effect of perturbation on the orbit (fac=1.*cforce)
      fac=1.*cforce

;different perturbation in each case
      for icase=1,4 do begin
          if(icase eq 1) then begin
              e1=80.    & e2=e1+20. & alfamui=0.25*fac
              comment='!7D!3a,!7De!3 > 0'
          endif
          if(icase eq 2) then begin
              e1=260.  & e2=e1+20. & alfamui=0.25*fac
              comment='!7D!3a,!7De!3 < 0'
          endif
          if(icase eq 3) then begin
              e1=0.1    & e2=e1+20. & alfamui=0.5*fac
              comment='!7Dx<0!3'
          endif
          if(icase eq 4) then begin
              e1=170.   & e2=e1+20. & alfamui=0.25*fac
              comment='!7Dx>0!3'
          endif

          st=string(alfamui,'(e8.2)')+string(e1,'(f6.0)')+string(e2,'(f6.0)')
          title='alfa,e1,e2= '+st

;for plotting: interval where perturbation is used
          index=where(e gt e1 and e lt e2)
          plot,r0(*,0),r0(*,1),thick=2,title=title,/iso,lines=2,chars=1.
          oplot,r0(index,0),r0(index,1),thick=2,col=2,lines=0,psym=6,syms=.3
          oplot,[-2,2],[0,0],lines=1
          oplot,[0,0],[-2,2],lines=1
```

```

xyouts,-1.8,1.6,comment,chars=1.5

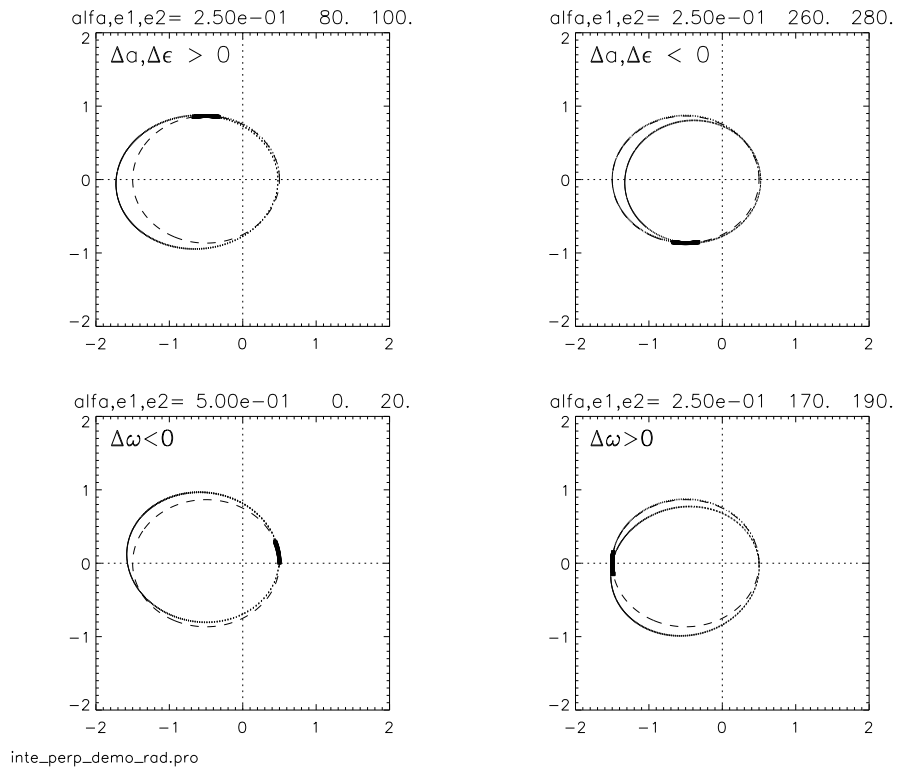
inte_perp,elem,0.,2.5,.0005,/orb,t=t1,r=r1,alfa=alfamui,$
e1=e1,e2=e2,final_elem=final_elem,sil=1
oplot,r1(*,0),r1(*,1),psym=3,col=2

endfor

close,1
xyouts,0.01,0.01,'inte_perp_demo_rad.pro',chars=1,/normal

print,' to plot same to psfile: '
print," psoopen,'inte_perp_demo_rad.ps',/vfont"
print," .run inte_perp_demo_rad"
print," psclose "
print," $lpr inte_perp_demo_rad.ps"
end

```



### Example: **inte\_perp\_demo\_vel.pro**

- The effect of instantaneous perturbation in the direction of velocity vector, affecting at different parts of the orbit (the same plot was attached to the lectures):

```
;*****
;inte_perp_demo_vel.pro
;apply perturbation at direction of velocity
;different parts of the orbit
;TM2006 HS
;*****

;elem=[a,e,i,ome,w,tau]
;elem=[1.,.5,1.,0.,0.,0.]
;myy=1.
;cforce=myy/elem(0)^2
;eks=elem(1)

;no perturbation
;inte_perp,elem,0.,1.,.0005,/orb,t=t0,r=r0,/sil,fin=efin0,outsave=1

;solve eccentric anomaly
;per=2.*pi*sqrt(elem(0)^3/myy)
;m=2*pi/per*(t0-elem(5))
;kepler_array,m,elem(1),e
;e=e*!radeg

xylim,2
!p.multi=[0,2,2]
charsize,1.5
nwin,xs=600,ys=600
fac=1.*cforce

;different perturbation in each case
for icase=1,4 do begin
    if(icase eq 1) then begin
        e1=80. & e2=e1+20. & betamui=0.2*fac
        comment='!7Dx>0!3'
    endif
    if(icase eq 2) then begin
        e1=260. & e2=e1+20. & betamui=0.2*fac
        comment='!7Dx<0!3'
    endif
    if(icase eq 3) then begin
        e1=0.1 & e2=e1+20. & betamui=0.5*fac
        comment='!7De!3 > 0'
    endif
    if(icase eq 4) then begin
        e1=180. & e2=e1+20. & betamui=0.25*fac
        comment='!7De!3 < 0'
    endif

    cose1=cos(e1/!radeg)
    cose2=cos(e2/!radeg)
    sine1=sin(e1/!radeg)
    sine2=sin(e2/!radeg)

    st=string(betamui,'(e8.2)')+string(e1,'(f5.0)')+string(e2,'(f5.0)')
    title='beta,e1,e2= '+st

;for plotting: interval where perturbation is used
```

```

index=where(e gt e1 and e lt e2)

plot,r0(*,0),r0(*,1),thick=2,title=title,/iso,lines=2,chars=1.
oplot,r0(index,0),r0(index,1),thick=2,col=2,lines=0,psym=6,syms=.3
oplot,[-2,2],[0,0],lines=1
oplot,[0,0],[-2,2],lines=1
xyouts,-1.8,1.6,comment,chars=1.5

inte_perp,elem,.0,2.5,.0005,/orb,t=t1,r=r1,beta=betamui,$
    e1=e1,e2=e2,final_elem=final_elem,sil=1
oplot,r1(*,0),r1(*,1),psym=3,col=2

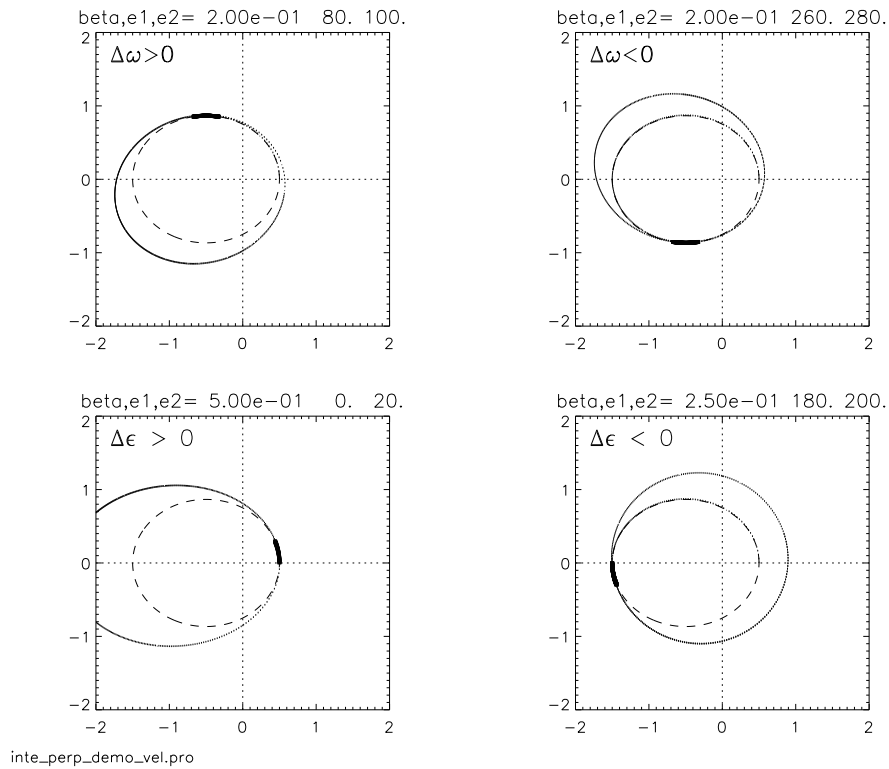
;-----

endfor

xyouts,0.01,0.01,'inte_perp_demo_vel.pro',chars=1,/normal

print,' to plot same to psfile: '
print," psopen,'inte_perp_demo_vel.ps',/vfont"
print," .run inte_perp_demo_vel
print," psclose "
print," $lpr inte_perp_demo_vel.ps"
end

```





### Example: **inte\_perp\_demo\_rad2.pro**

- The effect of continuous radial perturbation. Comparison to analytical formulas derived in the lectures (the same plot was attached to the lectures):

```
;*****
;inte_perp_demo_rad2.pro
;apply continuous radial perturbation
;compare numerical integration and analytical results
;*****

defplot
!p.multi=[0,3,3]
charsize,2.5
nwin

;elem=[a,e,i,ome,w,tau]
elem=[1.,.5,1.,0.,0.,0.]
myy=1.
cforce=myy/elem(0)^2
eks=elem(1)

alfamui=0.001*cforce
alfamui=0.005*cforce

t0=0.d0
t1=5.d0
dt=0.0005d0

;*****
;frame 1: non-perturbed orbit
;*****

inte_perp,elem,t0,t1,dt,/orb,/sil,r=r0,t=t0

xylim,2
plot,r0(*,0),r0(*,1),psym=3,xtitle='x',ytitle='y',title='non-perturbed',/iso
plots,0,0,psym=1

;*****
;frame 2: perturbed
;*****

t0=0.d0
t1=5.d0
dt=0.0005d0

inte_perp,elem,t0,t1,dt,/orb,alfa=alfamui,$
      save='inte_perp_demo_rad2.save',r=r,t=t

plot,r(*,0),r(*,1),psym=3,xtitle='x',ytitle='y',title='perturbed',/iso
plots,0,0,psym=1
xylim,0

;*****
;frame 3: plot info of elements
;*****

plot,lindgen(11),/nod,xs=15,ys=15 ; dummy plot defined for printing
ff='(g10.6)'
```

```

xyouts,1,9,'a   =' +string(elem(0),ff),chars=1.
xyouts,1,8,'eks =' +string(elem(1),ff),chars=1.
xyouts,1,7,'ink =' +string(elem(2),ff),chars=1.
xyouts,1,6,'ome =' +string(elem(3),ff),chars=1.
xyouts,1,5,'w   =' +string(elem(4),ff),chars=1.
xyouts,1,4,'tau =' +string(elem(5),ff),chars=1.
xyouts,1,3,'dt  =' +string(dt,ff),chars=1

; *****
; frames 4-9
; *****
;restore results and compare to analytical
;ttable(*) and elemtable(*,6) contain T and ELEM

restore,"inte_perp_demo_rad2.save"

;analytical formula:
;solve eccentric anomaly
m=2*pi/per*ttable
kepler_array,m,elem(1),e
e=e*!radeg

e1=e(0)
cose1=cos(e1/!radeg)
cose2=cos(e/!radeg)
sine1=sin(e1/!radeg)
sine2=sin(e/!radeg)

da=-alfamui/cforce*2.*eks*(cose2-cose1)
de=-alfamui/cforce*(1.-eks^2)*(cose2-cose1)
dw= alfamui/cforce*sqrt(1.-eks^2)*$
    ((e-e1)/!radeg -1./eks*(sine2-sine1))*!radeg

di=da*0.
dome=da*0.

index=lindgen(n_elements(ttable)/5)*5 ; plot every 5

plot,ttable/per,elemtable(*,0)-elem(0),xtitle='T/orb',ytitle='da'
oplot,ttable(index)/per,da(index),col=2,psym=6,syms=.3

plot,ttable/per,elemtable(*,1)-elem(1),xtitle='T/orb',ytitle='de',$
    title='alfa= ' +string(alfamui)
oplot,ttable(index)/per,de(index),col=2,psym=6,syms=.3

plot,ttable/per,elemtable(*,2)-elem(2),yr=[-10,10],xtitle='T/orb',ytitle='di'
oplot,ttable(index)/per,di(index),col=2,psym=6,syms=.3

plot,ttable/per,elemtable(*,3)-elem(3),yr=[-10,10],xtitle='T/orb',$
    ytitle='dome'
oplot,ttable(index)/per,dome(index),col=2,psym=6,syms=.3

index=lindgen(n_elements(ttable)/10)*10 ; plot every 10
plot,ttable/per,elemtable(*,4)-elem(4),xtitle='T/orb',ytitle='dw'
oplot,ttable(index)/per,dw(index),col=2,psym=6,syms=.5

plot,ttable/per,elemtable(*,5)-elem(5),xtitle='T/orb',ytitle='dtau'

; *****
charsize,1
xyouts,0.01,0.01,'inte_perp_demo_rad2.pro',chars=.75,/normal

```

```

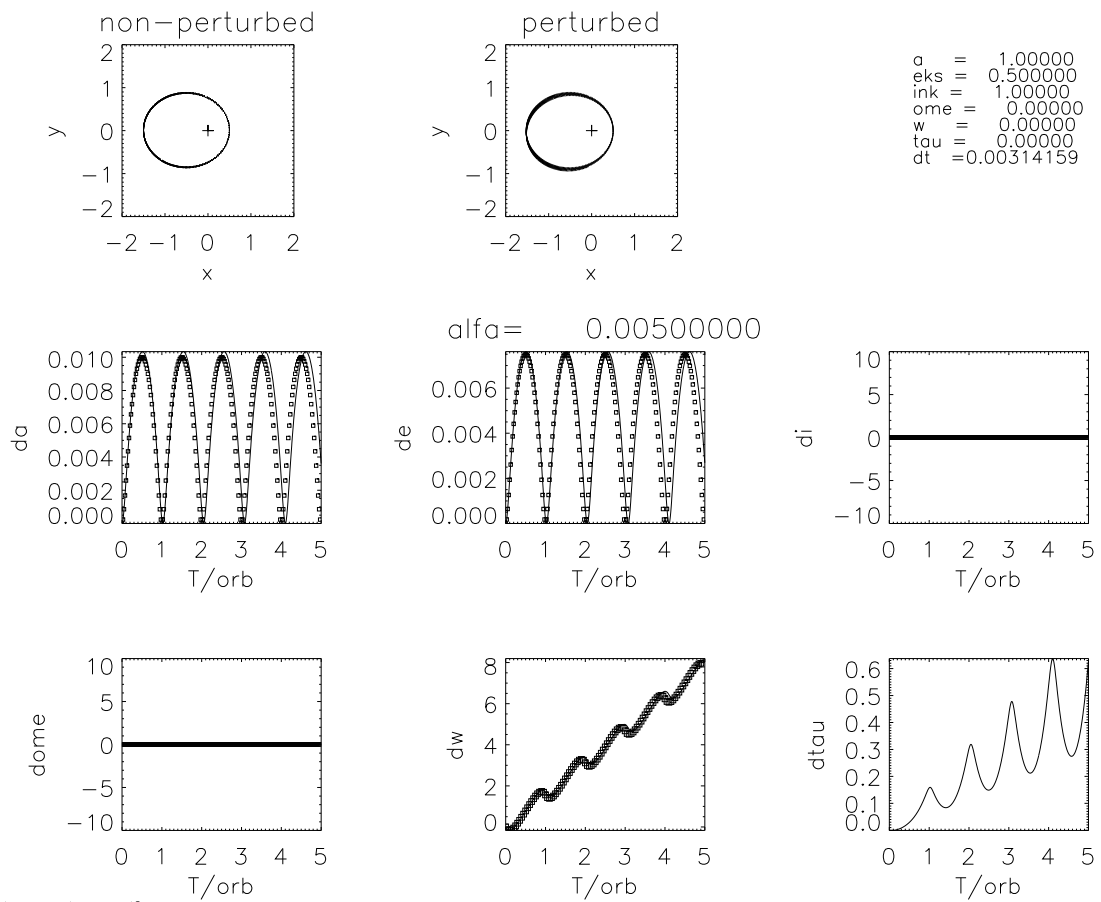
;*****

```

```

print,' to plot same to psfile: '
print," psopen,'inte_perp_demo_rad2.ps',/vfont"
print," .run inte_perp_demo_rad2
print," psclose "
print," $lpr inte_perp_demo_rad2.ps"
end

```



inte\_perp\_demo\_rad2.pro

### Example: **inte\_perp\_demo\_vel2.pro**

- The effect of continuous perturbation in the direction of velocity vector. Comparison to analytical formulas derived in the lectures (the same plot was attached to the lectures):

```
;*****
;inte_perp_demo_vel2.pro
;apply continuous perturbation in the direction of velocity vector
;compare numerical integration and analytical results
;hs 13.11.2006
;*****
  defplot
  !p.multi=[0,3,3]
  charsize,2.5
  nwin
;elem=[a,e,i,ome,w,tau]
  elem=[1.,.5,1.,0.,0.,0.]
  myy=1.
  cforce=myy/elem(0)^2
  eks=elem(1)
  betamui=0.0005*cforce

;*****
;frame 1: non-perturbed orbit
;*****
  t0=0.d0
  t1=5.d0
  dt=0.0005d0
  inte_perp,elem,t0,t1,dt,/orb,/sil,r=r0,t=t0
  xyim,2
  plot,r0(*,0),r0(*,1),psym=3,xtitle='x',ytitle='y',title='non-perturbed',/iso
  plots,0,0,psym=1

;*****
;frame 2: perturbed
;*****
  t0=0.d0
  t1=5.d0
  dt=0.0005d0
  inte_perp,elem,t0,t1,dt,/orb,beta=betamui,r=r,t=t,$
    save='inte_perp_demo_vel2.save'
  plot,r(*,0),r(*,1),psym=3,xtitle='x',ytitle='y',title='perturbed',/iso
  plots,0,0,psym=1
  xyim,0

;*****
;frame 3: plot info of elements
;*****
  plot,lindgen(11),/nod,xs=15,ys=15 ; dummy plot defined for printing
  ff='(g10.6)'
  xyouts,1,9,'a  =' +string(elem(0),ff),chars=1.
  xyouts,1,8,'eks =' +string(elem(1),ff),chars=1.
  xyouts,1,7,'ink =' +string(elem(2),ff),chars=1.
  xyouts,1,6,'ome =' +string(elem(3),ff),chars=1.
  xyouts,1,5,'w  =' +string(elem(4),ff),chars=1.
  xyouts,1,4,'tau =' +string(elem(5),ff),chars=1.
  xyouts,1,3,'dt  =' +string(dt,ff),chars=1

;*****
; frames 4-9
;*****
```

```

;restore results and compare to analytical
;ttable(*) and elemtable(*,6) contain T and ELEM
  restore,"inte_perp_demo_vel2.save"

;*****
;analytical formulas:
;1-order integration of p, eks, w with respect to E
;p,eks,w treated constants in calculation of derivatives
;*****
  etable=dindgen(10000)/1000.*max(ttable)/per*2.d0*!dpi
  atable=etable*0.
  ekstable=etable*0.
  wtable=etable*0.

  a0=elem(0)
  eks0=elem(1)
  w0=elem(4)
  p0=a0*(1.d0-eks0^2)
  delta_e=etable(1)-etable(0)
  eks_num=eks0
  p_num=p0
  w_num=w0
  e_num=etable(0)
  cose=cos(etable)
  sine=sin(etable)
  v=sqrt((1.d0/a0)*(1.d0+eks0*cose)/(1.d0-eks_num*cose))
  b=a0*sqrt(1.d0-eks0^2)
  dtde=sqrt(a0^3/myy)*(1.d0-eks0*cose)

  for i=0,9999 do begin
    deks=2.*betamui/v(i)*sqrt(p0/myy)*b*cose(i)*delta_e
    dp=2.*betamui*p0/v(i)*dtde(i)*delta_e
    dw=2.*betamui/v(i)*a0/eks0*sqrt(p0/myy)*sine(i)*delta_e
    eks_num=eks_num+deks
    p_num=p_num+dp
    w_num=w_num+dw*!radeg
    ekstable(i)=eks_num
    wtable(i)=w_num
    atable(i)=p_num/(1.-eks_num^2)
  endfor

  da=atable-a0
  de=ekstable-eks0
  dw=(wtable-w0) mod 360.
  di=da*0.
  dome=da*0.
  M=etable-eks0*sin(etable)
  pertable=m/2./!pi
  index=lindgen(n_elements(pertable)/50)*50 ; plot every 5

;*****
  plot,ttable/per,elemtable(*,0)-elem(0),xtitle='T/orb',ytitle='da'
  oplot,pertable(index),da(index),col=2,psym=6,syms=.3
  index=lindgen(n_elements(pertable)/10)*10 ; plot every 10
  plot,ttable/per,elemtable(*,1)-elem(1),xtitle='T/orb',ytitle='de',$
    title='beta= '+string(betamui)
  oplot,pertable(index),de(index),col=2,psym=6,syms=.3

;small eks approximation
  oplot,pertable,-betamui*eks*pertable*2*!pi,col=3

  plot,ttable/per,elemtable(*,2)-elem(2),yr=[-10,10],xtitle='T/orb',ytitle='di'
  oplot,pertable(index),di(index),col=2,psym=6,syms=.3

```

```

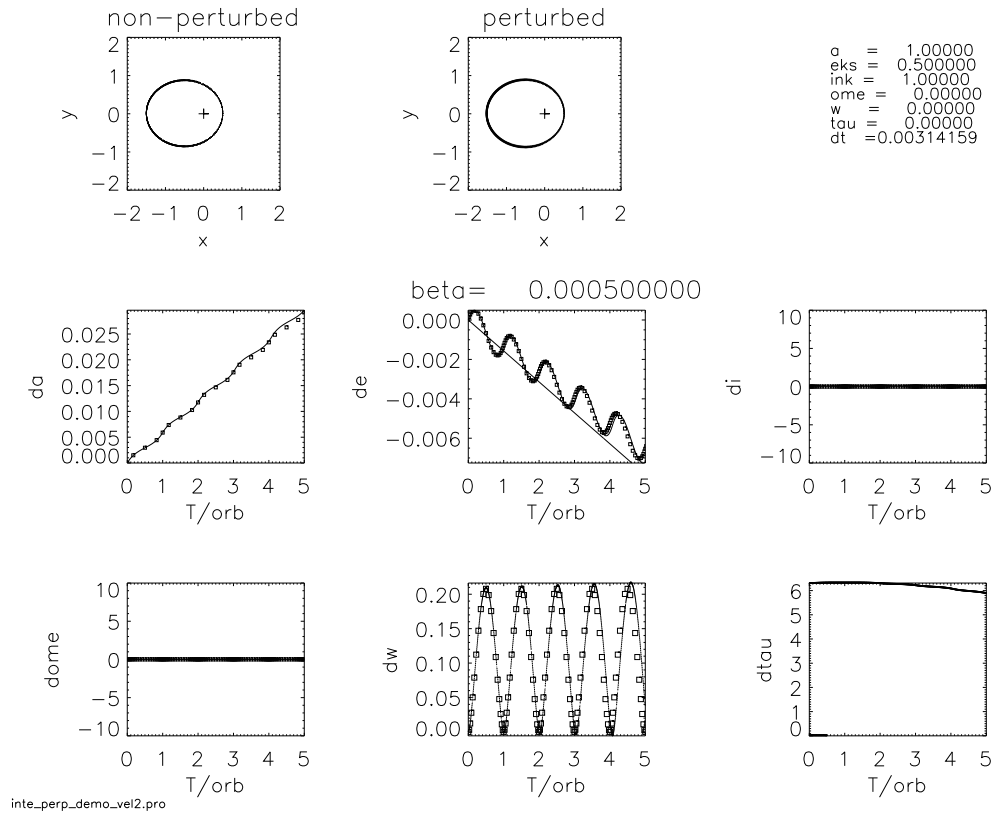
plot,ttable/per,elemtable(*,3)-elem(3),yr=[-10,10],$
  xtitle='T/orb',ytitle='dome'
oplot,pertable(index),dome(index),col=2,psym=6,syms=.3

plot,ttable/per,elemtable(*,4)-elem(4),xtitle='T/orb',ytitle='dw',psym=3
oplot,pertable(index),dw(index),col=2,psym=6,syms=.5

plot,ttable/per,elemtable(*,5)-elem(5),xtitle='T/orb',ytitle='dtau',psym=3
oplot,ttable/per,elemtable(*,5)-elem(5)+2.*!pi*sqrt(elemtable(*,0)^3/myy),$
  psym=3

;*****
charsize,1
xyouts,0.01,0.01,'inte_perp_demo_vel2.pro',chars=.75,/normal
;*****
print,' to plot same to pfile: '
print," psopen,'inte_perp_demo_vel2.ps',/vfont"
print," .run inte_perp_demo_vel2"
print," psclose "
print," $lpr inte_perp_demo_vel2.ps"
end

```



### Example: **inte\_perp\_demo\_nor2.pro**

- The effect of continuous perturbation in the direction normal to the orbit. Comparison to analytical formulas derived in the lectures (the same plot was attached to the lectures):

```
,*****
;
;inte_perp_demo_nor2.pro
;apply continuous radial perturbation
;compare numerical integration and analytical results
;hs 13.11.2006
;*****
  defplot
  !p.multi=[0,3,3]
  charsize,2.5
  nwin
;elem=[a,e,i,ome,w,tau]
  elem=[1.,.05,20.,30.,40.,0.]
  myy=1.
  cforce=myy/elem(0)^2
  eks=elem(1)
  gammamui=0.001*cforce*.01

;*****
;frame 1: non-perturbed orbit
;*****
  t0=0.d0
  t1=5.d0
  dt=0.0005d0
  inte_perp,elem,t0,t1,dt,/orb,/sil,r=r0,t=t0
  xylin,2
  plot,r0(*,0),r0(*,1),psym=3,xtitle='x',ytitle='y',title='non-perturbed',/iso
  plots,0,0,psym=1

;*****
;frame 2: perturbed
;*****
  t0=0.d0
  t1=5.d0
  dt=0.0005d0
  inte_perp,elem,t0,t1,dt,/orb,gamma=gammamui,$
    save='inte_perp_demo_nor2.save',t=t,r=r
  plot,r(*,0),r(*,1),psym=3,xtitle='x',ytitle='y',title='perturbed',/iso
  plots,0,0,psym=1
  xylin,0

;*****
;frame 3: plot info of elements
;*****
  plot,lindgen(11),/nod,xs=15,ys=15 ; dummy plot defined for printing
  ff='(g10.6)'
  xyouts,1,9,'a  =' +string(elem(0),ff),chars=1.
  xyouts,1,8,'eks =' +string(elem(1),ff),chars=1.
  xyouts,1,7,'ink =' +string(elem(2),ff),chars=1.
  xyouts,1,6,'ome =' +string(elem(3),ff),chars=1.
  xyouts,1,5,'w   =' +string(elem(4),ff),chars=1.
  xyouts,1,4,'tau =' +string(elem(5),ff),chars=1.
  xyouts,1,3,'dt  =' +string(dt,ff),chars=1

;*****
; frames 4-9
```

```

;*****
;restore results and compare to analytically
;ttable(*) and elemtable(*,6) contain T and ELEM
  restore,"inte_perp_demo_nor2.save"

;*****
;analytical formula:
;solve eccentric anomaly
  m=2*pi/per*ttable
  eks=elem(1)
  kepler_array,m,eks,e
  e=e*!radeg
  e1=e(0)
  cose1=cos(e1/!radeg)
  cose2=cos(e/!radeg)
  sine1=sin(e1/!radeg)
  sine2=sin(e/!radeg)
  sini=sin(elem(2)/!radeg)
  cosi=cos(elem(2)/!radeg)
  sinw=sin(elem(4)/!radeg)
  cosw=cos(elem(4)/!radeg)

  apu1=(1.+eks^2)*(sine2-sine1)-eks*(e-e1)/!radeg-$
    eks*(0.25*(sin(2.*e/!radeg)-sin(2.*e1/!radeg))+0.5*(e-e1)/!radeg)
  apu1=(1.+eks^2)*(sine2-sine1)-eks*(0.25*(sin(2.*e/!radeg)-sin(2.*e1/!radeg))+1.5*(e-e1)/!radeg)

  apu2=-(cose2-cose1)-eks*(sin(e/!radeg)^2-sin(e1/!radeg)^2)
  apu2=-(cose2-cose1)+.25*eks*(cos(2*e/!radeg)-cos(2*e1/!radeg))

  dome=gammamui/sqrt(1.-eks^2)/sini*(sinw*apu1+cosw*sqrt(1.-eks^2)*apu2)*!radeg
  di=gammamui/sqrt(1.-eks^2)*(cosw*apu1-sinw*sqrt(1.-eks^2)*apu2)*!radeg
  da=di*0.
  dw=-cosi*dome
  de=di*0.

  di_sec=gammamui/sqrt(1.-eks^2)*cosw*(-1.5)*eks*e
  dome_sec=gammamui/sqrt(1.-eks^2)/sini*sinw*(-1.5)*eks*e
  dw_sec=-cosi*dome_sec

;*****

  index=lindgen(n_elements(ttable)/10)*10 ; plot every 10

  plot,ttable/per,elemtable(*,0)-elem(0),xtitle='T/orb',ytitle='da',$
    yr=[-1,1]*1d-8
  oplot,ttable(index)/per,da(index),col=2,psym=6,syms=.3

  plot,ttable/per,elemtable(*,1)-elem(1),xtitle='T/orb',ytitle='de',$
    title='gamma= '+string(gammamui),yr=[-1,1]*1d-8
  oplot,ttable(index)/per,de(index),col=2,psym=6,syms=.3

  plot,ttable/per,elemtable(*,2)-elem(2),xtitle='T/orb',ytitle='di',psym=3
  oplot,ttable(index)/per,di(index),col=2,psym=6,syms=.3
  oplot,ttable(index)/per,di_sec(index),col=3,lines=2

  plot,ttable/per,elemtable(*,3)-elem(3),xtitle='T/orb',ytitle='dome',psym=3
  oplot,ttable(index)/per,dome(index),col=2,psym=6,syms=.3
  oplot,ttable(index)/per,dome_sec(index),col=3,lines=2

  plot,ttable/per,elemtable(*,4)-elem(4),xtitle='T/orb',ytitle='dw',psym=3
  oplot,ttable(index)/per,dw(index),col=2,psym=6,syms=.5

```



```

oplot,ttable(index)/per,dw_sec(index),col=3,lines=2

plot,ttable/per,elemtable(*,5)-elem(5),xtitle='T/orb',ytitle='dtau',psym=3
oplot,ttable/per,elemtable(*,5)-elem(5)+per,psym=3

;*****
charsize,1
xyouts,0.01,0.01,'inte_perp_demo_nor2.pro',chars=.75,/normal
;*****
print,' to plot same to psfile: '
print," psopen,'inte_perp_demo_nor2.ps',/vfont"
print," .run inte_perp_demo_nor2"
print," psclose "
print," $lpr inte_perp_demo_nor2.ps"
end

```

