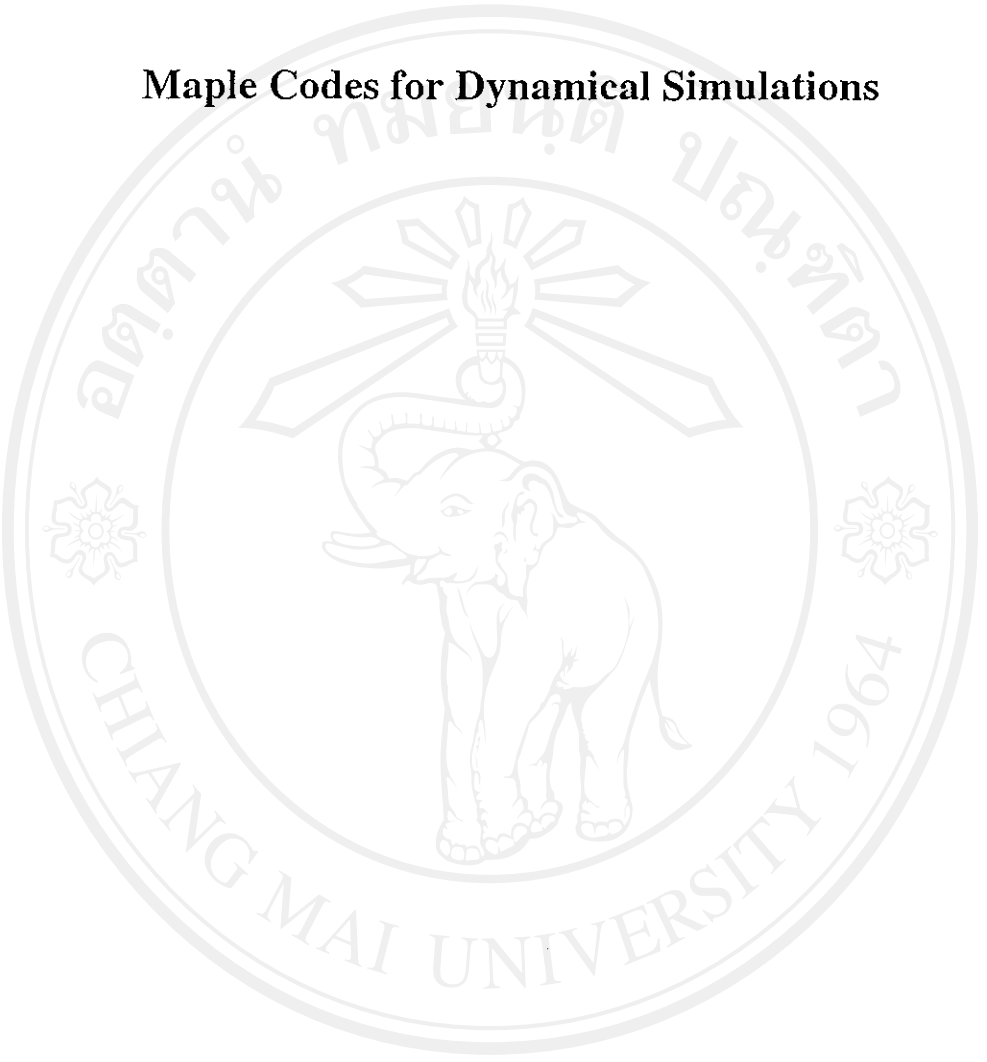
APPENDIX

Maple Codes for Dynamical Simulations



New Orbit of Particles

Initializations

```
> with(plots):
Warning, the name changecoords has been redefined

> G:=1.184e-4; py:=evalf(Pi);
```

$$G := 0.0001184$$

$$py := 3.141592654$$

```
> r:=(alpha,epsilon,phi,theta)->alpha/(1-epsilon*cos(theta-phi*py/180));
```

$$r := (\alpha, \varepsilon, \phi, \theta) \rightarrow \frac{\alpha}{1 - \varepsilon \cos\left(\theta - \frac{1}{180}\,\phi\, py\right)}$$

```
> M:=1/G;
```

$$M := 8445.945946$$

New Orbit Procedures

```
> Neworb:=proc(m1,m2,L1,L2,ec1,ec2,k,phd)

  # la->(latus rectum)/2
  # ec->eccentricity
  # phd->phase difference of orbits
  # or_orb->oiginal orbit of particle
  # p->radial momentum
  # L->angular momentum
  # E->total energy of each particle
  # col_ang->angle of collision point
  # k->collision points, select whether 1 or 2
  # Sign->sign of radial momentum
  # _or->of original orbit
  # _re->of recoil particle
  # re_orb->new or bit of recoil particle
  # col_point->collision point

  local la_or1,la_or2,or_orb1,or_orb2,r_max1,r_max2,E_or1,E_or2,col_ang,K,
        r_col,Sign1,Sign2,p_or1,p_or2,L_re1,L_re2,p_re1,p_re2,E_re1,E_re2,
        la_re1,la_re2,ec_re1,ec_re2,ph_re1,ph_re2,re_orb1,re_orb2,col_point;

  la_or1:=L1^2/(G*M*m1^2);
  la_or2:=L2^2/(G*M*m2^2);

  or_orb1:=plot(r(la_or1,ec1, 0 ,t),t=0..2*Pi,coords=polar,color=black,thickness=3);
  or_orb2:=plot(r(la_or2,ec2,phd,t),t=0..2*Pi,coords=polar,color=black,thickness=3);

  r_max1:=r(la_or1,ec1, 0 , 0 );
  r_max2:=r(la_or2,ec2,phd,phd*Pi/180);

  E_or1:=evalf(L1^2/(2*m1*r_max1^2)-G*M*m1/r_max1);
  E_or2:=evalf(L2^2/(2*m2*r_max2^2)-G*M*m2/r_max2);

  col_ang[1]:=max(solve(r(la_or1,ec1,0,t)=r(la_or2,ec2,phd,t))*180/py);
  col_ang[2]:=min(solve(r(la_or1,ec1,0,t)=r(la_or2,ec2,phd,t))*180/py);

  r_col[1]:=evalf(r(la_or1,ec1,0,col_ang[1]*Pi/180));
```

```
r_col[2]:=evalf(r(la_or1,ec1,0,col_ang[2]*Pi/180));

Sign1:=T->sign(evalf(subs(t=T,diff(r(la_or1,ec1, 0 ,t),t))));
Sign2:=T->sign(evalf(subs(t=T,diff(r(la_or2,ec2,phd,t),t))));

p_or1:=(sqrt((E_or1+G*M*m1/r_col[k]-1/2*L1^2/m1/r_col[k]^2)*2*m1))*Sign1(col_ang[k]
*py/180);
p_or2:=(sqrt((E_or2+G*M*m2/r_col[k]-1/2*L2^2/m2/r_col[k]^2)*2*m2))*Sign2(col_ang[k]
*py/180);

L_re1:=((m1-m2)/(m1+m2))*L1+(2*m1/(m1+m2))*L2;
L_re2:=(2*m2/(m1+m2))*L1-((m1-m2)/(m1+m2))*L2;

p_re1:=((m1-m2)/(m1+m2))*p_or1+(2*m1/(m1+m2))*p_or2;
p_re2:=(2*m2/(m1+m2))*p_or1-((m1-m2)/(m1+m2))*p_or2;

E_re1:=p_re1^2/(2*m1)+L_re1^2/(2*m1*r_col[k]^2)-G*M*m1/r_col[k];
E_re2:=p_re2^2/(2*m2)+L_re2^2/(2*m2*r_col[k]^2)-G*M*m2/r_col[k];

la_re1:=L_re1^2/(G*M*m1^2);
la_re2:=L_re2^2/(G*M*m2^2);

ec_re1:=sqrt(1+2*E_re1*L_re1^2/(G^2*M^2*m1^3));
ec_re2:=sqrt(1+2*E_re2*L_re2^2/(G^2*M^2*m2^3));

K:=k->piecewise(k=1,1,k=2,-1,0);

ph_re1:=evalf(col_ang[k]-K(k)*arccos((1-la_re1/r_col[k])/ec_re1)*180/Pi);
ph_re2:=evalf(col_ang[k]-K(k)*arccos((1-la_re2/r_col[k])/ec_re2)*180/Pi);

re_orb1:=plot(r(la_re1,ec_re1,ph_re1,t),t=0..2*Pi,coords=polar,color=black,linestyl
e=DASHDOT);
re_orb2:=plot(r(la_re2,ec_re2,ph_re2,t),t=0..2*Pi,coords=polar,color=black);

col_point:=pointplot([r_col[k],col_ang[k]*py/180],coords=polar,symbol=circle,symbol
size=20);

print(l_ori=L1/L2);
print(l_rec=evalf(L_re1/L_re2));
print(Delta*phi=phd);
display(or_orb1,or_orb2,re_orb1,re_orb2,col_point,scaling=constrained);

end proc:
```

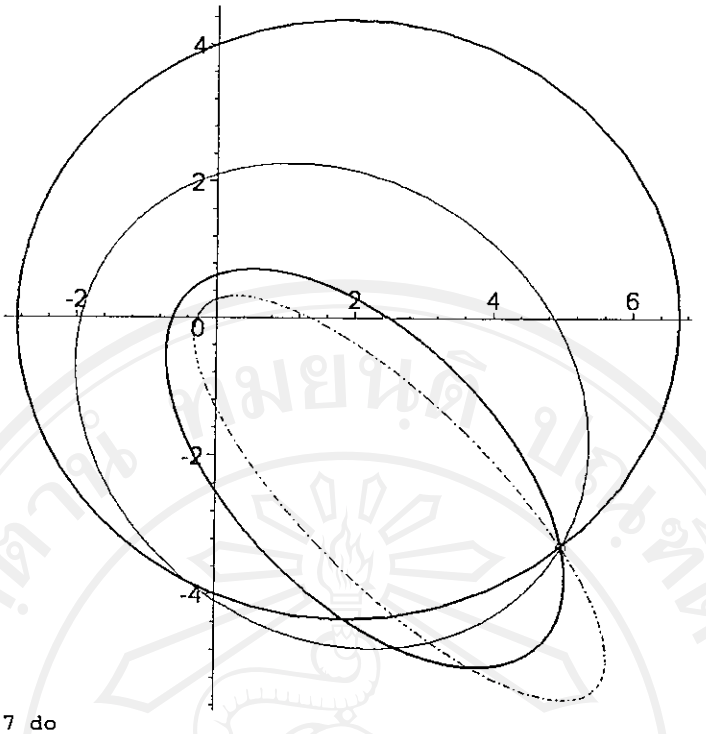
Evaluation of New Orbits

```
> Neworb(0.5,1,1,1,0.4,0.85,1,315);
```

$$l\_ori = 1$$

$$l\_rec = 0.2000000001$$

$$\Delta\phi = 315$$




```
> for i from 0 to 7 do
      Neworb(0.5,1,1,1,0.4,0.85,1,45*i);
  end do;
>
```

Two-Real-Body Collision

Initializations

```
> restart;
> with(plots):
Warning, the name changecoords has been redefined

> with(plottools):
Warning, the name arrow has been redefined

> G:=1.184e-4; py:=evalf(Pi);
```

$$G := 0.0001184$$

$$py := 3.141592654$$

```
> r:=(alpha,epsilon,phi,theta)->alpha/(1-epsilon*cos(theta-phi*py/180));
```

$$r := (\alpha, \varepsilon, \phi, \theta) \rightarrow \frac{\alpha}{1 - \varepsilon \cos\left(\theta - \frac{t}{180}\,\phi\, py\right)}$$

```
> M:=1/G; #central mass
  m1:=0.5;
  m2:=1.0;
```

$$M := 8445.945946$$

$$m1 := 0.5$$

$$m2 := 1.0$$

```
> a:=0.1; #radius of object
```

$$a := 0.1$$

Switching Function

```
> Rigidity:=1e8;
```

$$Rigidity := 0.1\ 10^{9}$$

```
> Sw:=(Status,r)->piecewise(Status=1 and r<=2*a,Rigidity,0);
```

$$Sw := (Status, r) \rightarrow \text{piecewise}(Status = 1 \text{ and } r \le 2\ a, Rigidity, 0)$$

Solution of New Orbit

```
> Solution:=proc(r1,r2,u1,u2,v1,v2,ang1,ang2,St)
    local System;

  System:={
  diff(x1(t),t,t)=-G*M*x1(t)/(x1(t)^2+y1(t)^2)^(3/2)

  +G*m2*(x1(t)-x2(t))*Sw(St,sqrt((x1(t)-x2(t))^2+(y1(t)-y2(t))^2))/a^3,

  diff(y1(t),t,t)=-G*M*y1(t)/(x1(t)^2+y1(t)^2)^(3/2)

  +G*m2*(y1(t)-y2(t))*Sw(St,sqrt((x1(t)-x2(t))^2+(y1(t)-y2(t))^2))/a^3,

  diff(x2(t),t,t)=-G*M*x2(t)/(x2(t)^2+y2(t)^2)^(3/2)

  +G*m1*(x2(t)-x1(t))*Sw(St,sqrt((x1(t)-x2(t))^2+(y1(t)-y2(t))^2))/a^3,
```

```
diff(y2(t),t,t)=-G*M*y2(t)/(x2(t)^2+y2(t)^2)^(3/2)

+G*m2*(y2(t)-y1(t))*Sw(St,sqrt((x1(t)-x2(t))^2+(y1(t)-y2(t))^2))/a^3,

x1(0)= r1*cos(ang1*py/180), y1(0)= r1*sin(ang1*py/180),
x2(0)= r2*cos(ang2*py/180), y2(0)= r2*sin(ang2*py/180),

D(x1)(0)= u1*cos(ang1*py/180)-v1*sin(ang1*py/180),
D(y1)(0)= u1*sin(ang1*py/180)+v1*cos(ang1*py/180),
D(x2)(0)= u2*cos(ang2*py/180)-v2*sin(ang2*py/180),
D(y2)(0)= u2*sin(ang2*py/180)+v2*cos(ang2*py/180)
};
     dsolve(System,numeric,output=listprocedure,maxfun=5000000);
end proc:
```

Determination of Initial Conditions Before Collision

```
> L1:=1;
  L2:=1.1;
```

$$L1 := 1$$

$$L2 := 1.1$$

```
> la1:=L1^2/(G*M*m1^2); la2:=L2^2/(G*M*m2^2);
  ec1:=0.4; ec2:=0.85;
  ph1:=0;  ph2:=90; # ph1 must be 0
```

$$la1 := 4.000000000$$

$$la2 := 1.209999999$$

$$ec1 := 0.4$$

$$ec2 := 0.85$$

$$ph1 := 0$$

$$ph2 := 90$$

```
> r_max1:=r(la1,ec1,ph1,ph1*py/180);
  r_max2:=r(la2,ec2,ph2,ph2*py/180);
  r_min1:=r(la1,ec1,ph1,(ph1+180)*py/180);
  r_min2:=r(la2,ec2,ph2,(ph2+180)*py/180);
```

$$r\_max1 := 6.666666667$$

$$r\_max2 := 8.066666660$$

$$r\_min1 := 2.857142857$$

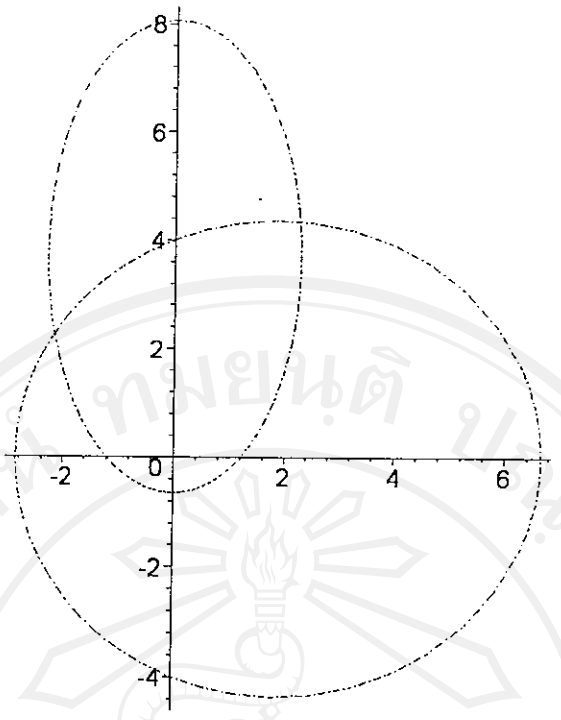
$$r\_min2 := 0.6540540535$$

```
> E1:=evalf(L1^2/(2*m1*r_max1^2)-G*M*m1/r_max1);
  E2:=evalf(L2^2/(2*m2*r_max2^2)-G*M*m2/r_max2);
```

$$E1 := -0.05250000000$$

$$E2 := -0.1146694216$$

```
> Ori_orb1:=plot(r(la1,ec1,ph1,t),t=0..2*Pi,coords=polar,linestyle=DASHDOT,color=blac
  k):
  Ori_orb2:=plot(r(la2,ec2,ph2,t),t=0..2*Pi,coords=polar,linestyle=DASHDOT,color=blac
  k):
  display(Ori_orb1,Ori_orb2,scaling=constrained);
```




```
> solve(r(la1,ec1,ph1,t)=r(la2,ec2,ph2,t))*180/py; #collision angles
```

$$133.7708857,\ 62.43269328$$

```
> angP[1]:=max(solve(r(la1,ec1,ph1,t)=r(la2,ec2,ph2,t))*180/py);
  angP[2]:=min(solve(r(la1,ec1,ph1,t)=r(la2,ec2,ph2,t))*180/py);
```

$$angP_1 := 133.7708857$$

$$angP_2 := 62.43269328$$

```
> r_inP[1]:=r(la1,ec1,ph1,angP[1]*py/180);
  r_inP[2]:=r(la2,ec2,ph2,angP[2]*py/180);
```

$$r\_inP_1 := 3.133051620$$

$$r\_inP_2 := 4.908674813$$

```
> P:=2; #collision point, select whether 1 or 2
```

$$P := 2$$

```
> Sign1:=theta->sign(evalf(subs(t=theta,diff(r(la1,ec1,ph1,t),t))));
  Sign2:=theta->sign(evalf(subs(t=theta,diff(r(la2,ec2,ph2,t),t))));
```

$$Sign1 := \theta \rightarrow \mathrm{sign}\left(\mathrm{evalf}\left(\mathrm{subs}\left(t=\theta, \frac{\partial}{\partial t}\mathrm{r}(la1, ec1, ph1, t)\right)\right)\right)$$

$$Sign2 := \theta \rightarrow \mathrm{sign}\left(\mathrm{evalf}\left(\mathrm{subs}\left(t=\theta, \frac{\partial}{\partial t}\mathrm{r}(la2, ec2, ph2, t)\right)\right)\right)$$

Determine **radial velocity "u"** and **azimuthal velocity "v"** of particles.

```
> u1:=evalf((sqrt((E1+G*M*m1/(r_inP[P])-1/2*L1^2/m1/(r_inP[P])^2)*2/m1)))*Sign1(angP[
  P]*py/180);
  u2:=evalf((sqrt((E2+G*M*m2/(r_inP[P])-1/2*L2^2/m2/(r_inP[P])^2)*2/m2)))*Sign2(angP[
  P]*py/180);
```

$$u1 := -0.1772935589$$

$$u2 := 0.3576106766$$

```
> v1:=L1/(m1*r_inP[P]);
  v2:=L2/(m2*r_inP[P]);
```

$$v1 := 0.4074419423$$

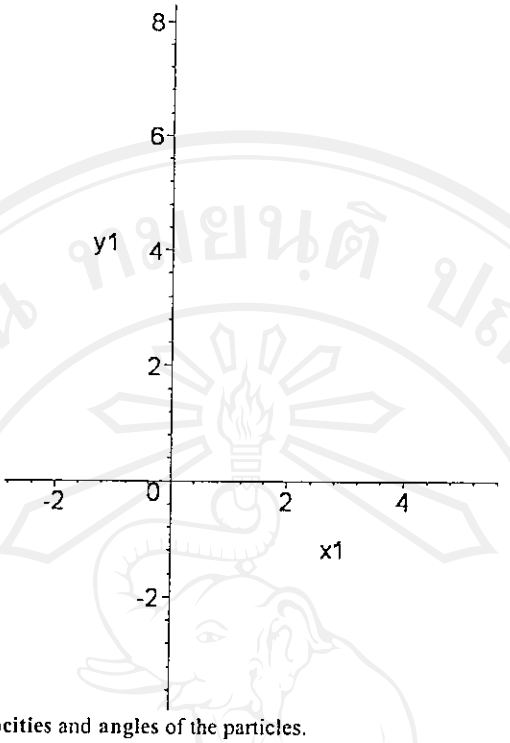
$$v2 := 0.2240930683$$

Substitute the initial conditions at collision point P. Run simulation **without** collision by put 0 in the last argument of Solution().

```
> K:=Solution(r_inP[P],r_inP[P],u1,u2,v1,v2,angP[P],angP[P],0):
```

Display the tracks of both particles without collision.

```
> Track1:=odeplot(K,[x1(t),y1(t)],-10..20,frames=21,numpoints=300,color=black):
  Track2:=odeplot(K,[x2(t),y2(t)],-10..20,frames=21,numpoints=300,color=black):
> display(Track1,Track2,scaling=constrained);
```



Construct the *new* coordinates, velocities and angles of the particles.

```
> R1:=eval(sqrt(x1(t)^2+y1(t)^2),K);
  U1:=eval(diff(sqrt(x1(t)^2+y1(t)^2),t),K);
  V1:=t->L1/(m1*R1(t));
  Ango1:=eval(x1(t)/sqrt(x1(t)^2+y1(t)^2),K);
  Ang1:=t->arccos((Ango1(t)))*180/py;


  R2:=eval(sqrt(x2(t)^2+y2(t)^2),K);
  U2:=eval(diff(sqrt(x2(t)^2+y2(t)^2),t),K);
  V2:=t->L2/(m2*R2(t));
  Ango2:=eval(x2(t)/sqrt(x2(t)^2+y2(t)^2),K);
  Ang2:=t->arccos(Ango2(t))*180/py;
```

$$R1 := \sqrt{(\text{proc}(t)\ \ldots\ \text{end proc})^2 + (\text{proc}(t)\ \ldots\ \text{end proc})^2}$$

$$U1 := \frac{1}{2}\,\frac{2\,(\text{proc}(t)\ \ldots\ \text{end proc})\,(\text{proc}(t)\ \ldots\ \text{end proc}) + 2\,(\text{proc}(t)\ \ldots\ \text{end proc})\,(\text{proc}(t)\ \ldots\ \text{end proc})}{\sqrt{(\text{proc}(t)\ \ldots\ \text{end proc})^2 + (\text{proc}(t)\ \ldots\ \text{end proc})^2}}$$

$$V1 := t \to \frac{L1}{m1\ \mathrm{R1}(t)}$$

$$Ango1 := \frac{\text{proc}(t)\ \ldots\ \text{end proc}}{\sqrt{(\text{proc}(t)\ \ldots\ \text{end proc})^2 + (\text{proc}(t)\ \ldots\ \text{end proc})^2}}$$

$$Ang1 := t \to \frac{180\ \arccos(\mathrm{Ango1}(t))}{py}$$

$$R2 := \sqrt{(\text{proc}(t)\ \ldots\ \text{end proc})^2 + (\text{proc}(t)\ \ldots\ \text{end proc})^2}$$

$$U2 := \frac{1}{2}\,\frac{2\,(\text{proc}(t)\ \ldots\ \text{end proc})\,(\text{proc}(t)\ \ldots\ \text{end proc}) + 2\,(\text{proc}(t)\ \ldots\ \text{end proc})\,(\text{proc}(t)\ \ldots\ \text{end proc})}{\sqrt{(\text{proc}(t)\ \ldots\ \text{end proc})^2 + (\text{proc}(t)\ \ldots\ \text{end proc})^2}}$$

$$V2 := t \to \frac{L2}{m2\ \mathrm{R2}(t)}$$

$$Ango2 := \frac{\text{proc}(t) \ \ldots \ \text{end proc}}{\sqrt{(\text{proc}(t) \ \ldots \ \text{end proc})^2 + (\text{proc}(t) \ \ldots \ \text{end proc})^2}}$$

$$Ang2 := t \rightarrow \frac{180 \arccos(\text{Ango2}(t))}{\pi}$$

Points Evaluation

Evaluate the new initial conditions by run simulation back to the past. For more precision, | T | should be small.

```
> T:=-0.5;R1(T);R2(T);U1(T);U2(T);V1(T);V2(T);Ang1(T);Ang2(T);
```

$$T := -0.5$$

4.996347988

4.725878393

-0.1733735511

0.3737451344

0.4002923745

0.2327609618

60.09667011

61.07463738

Run simulation **with** collision by put 1 in the last argument of Solution().

```
> S:=Solution(R1(T),R2(T),U1(T),U2(T),V1(T),V2(T),Ang1(T),Ang2(T),1):
```

Evaluate position of both objects.

```
> Pm1:=[
  eval(x1(t),S),
  eval(y1(t),S)]:

  Pm2:=[
  eval(x2(t),S),
  eval(y2(t),S)]:
```

Simulation Display

Choose the time and smoothness of simulation.

```
> Start:=-10; Finish:=50; Smooth:=1; Step:=(Finish-Start)*Smooth;
```

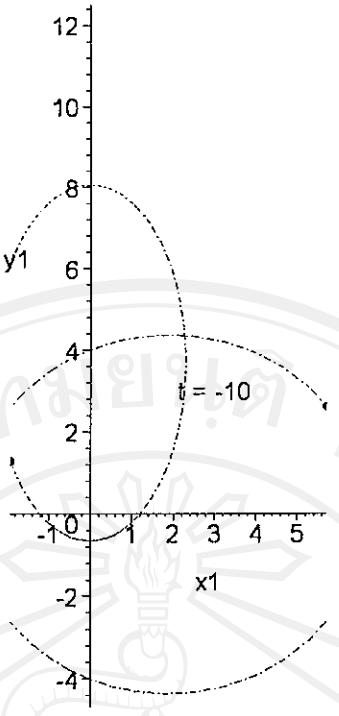
$$Start := -10$$

$$Finish := 50$$

$$Smooth := 1$$

$$Step := 60$$

```
> Obj1:=display(seq(disk(Pm1(Start+j/Smooth),a,color=red),j=0..Step),insequence=true)
  :

  Obj2:=display(seq(disk(Pm2(Start+j/Smooth),a,color=blue),j=0..Step),insequence=true
  ):

  TRACK1:=odeplot(S,[x1(t),y1(t)],Start..Finish,frames=Step+1,numpoints=300):

  TRACK2:=odeplot(S,[x2(t),y2(t)],Start..Finish,frames=Step+1,numpoints=300):

  TIME:=display(seq(textplot([3,3,t=round(Start+j/Smooth)],color=blue),j=0..Step),ins
  equence=true):

> display(Ori_orb1,Ori_orb2,Obj1,Obj2,TRACK1,TRACK2,TIME,scaling=constrained);
```



>

If the new tracks disappear, please increase the radius a of particles.

If the particles do not collide, please shorten the time |T|.

If the new orbit are not related to the original orbits, please consider the solution of collision angles provided by "solve", sometime it may gives us four values but two of them are trivial. Select them again by copy and paste on angP[1] or angP[2] to assign the precise collision angle.

Orbital Motion in the Resisting Medium

Initializations

```
> with(plots):
  with(plottools):
Warning, the name changecoords has been redefined

Warning, the name arrow has been redefined
```

```
> G:=1.184e-4;
```

$$G := 0.0001184$$

```
> M:=1/G;
```

$$M := 8445.945946$$

Trigger

```
> Thickness:=0.1; #thickness of disk cloud
```

$$Thickness := 0.1$$

```
> ON:=0; #starting time of applying interaction
```

$$ON := 0$$

```
> kappa:=0.3; #retarding factor
```

$$\kappa := 0.3$$

```
> K:=(z,t)->piecewise(t>=ON and abs(z)<=Thickness,kappa,0);
```

$$K := (z, t) \rightarrow \text{piecewise}(ON \le t \text{ and } |z| \le Thickness, \kappa, 0)$$

```
> fx:=1; fy:=1; fz:=1;
  #component that retarding force does not exist can be defined by 0
```

$$fx := 1$$

$$fy := 1$$

$$fz := 1$$

Solution

```
> Solution:=proc(r,v,inc)
      local Equaton;

  Equaton:={
  diff(x(t),t,t)=-G*M*x(t)/(x(t)^2+y(t)^2+z(t)^2)^(3/2)-fx*K(z(t),t)*abs(diff(x(t),t)
  +sqrt(G*M)*y(t)/(x(t)^2+y(t)^2)^(3/4))*(diff(x(t),t)+sqrt(G*M)*y(t)/(x(t)^2+y(t)^2)
  ^(3/4)),

  diff(y(t),t,t)=-G*M*y(t)/(x(t)^2+y(t)^2+z(t)^2)^(3/2)-fy*K(z(t),t)*abs(diff(y(t),t)
  -sqrt(G*M)*x(t)/(x(t)^2+y(t)^2)^(3/4))*(diff(y(t),t)-sqrt(G*M)*x(t)/(x(t)^2+y(t)^2)
  ^(3/4)),

  diff(z(t),t,t)=-G*M*z(t)/(x(t)^2+y(t)^2+z(t)^2)^(3/2)-fz*K(z(t),t)*abs(diff(z(t),t)
  )*diff(z(t),t),

       x(0)=r*cos(inc*Pi/180)  , D(x)(0)=0,
       y(0)=0                  , D(y)(0)=v,
       z(0)=r*sin(inc*Pi/180)  , D(z)(0)=0
```

```
};

dsolve(Equaton,numeric,output=listprocedure,maxfun=500000);
end proc:
```

Object Evaluation

```
> f:=0.8; #f>0,for f:=1 the orbit is circle
```

$$f := 0.8$$

```
> r:=1; v:=f*sqrt(G*M/r); inc:=30; #initial conditions
```

$$r := 1$$

$$v := 0.8000000000$$

$$inc := 30$$

```
> Obj:=[eval(x(t),Solution(r,v,inc)),eval(y(t),Solution(r,v,inc)),eval(z(t),Solution(
  r,v,inc))];
```

$$Obj := [\text{proc}(t) \ \ldots \ \text{end proc}, \text{proc}(t) \ \ldots \ \text{end proc}, \text{proc}(t) \ \ldots \ \text{end proc}]$$

Simulation Display

```
> Start:=0; Finish:=50; Smooth:=2; Step:=(Finish-Start)*Smooth;
```

$$Start := 0$$

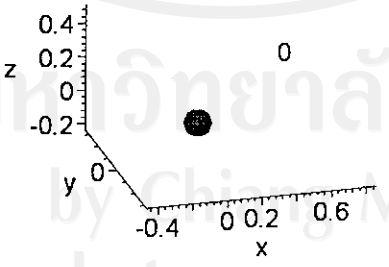
$$Finish := 50$$

$$Smooth := 2$$

$$Step := 100$$

```
> Center:=display(sphere([0,0,0],0.08,color=red),style=patchcontour):

  Object:=display(seq(pointplot3d(Obj(Start+j/Smooth),symbol=circle,color=blue),j=0..
  Step),insequence=true):

  Time:=display(seq(textplot3d([r/2,0,r/3,round(Start+j/Smooth)],color=black),j=0..St
  ep),insequence=true):

  Track:=odeplot(Solution(r,v,inc),[x(t),y(t),z(t)],Start..Finish,frames=Step+1,numpo
  ints=500,color=green):
```

```
> display(Center,Object,Time,Track,scaling=constrained,axes=frame,orientation=[-105,7
  0]);
```



Tidal Induction
with
Mass Distribution Function

Initializations

```
> restart;
> with(plots):
Warning, the name changecoords has been redefined
> with(plottools):
Warning, the name arrow has been redefined
> G:=1.184e-4;
```

$$G := 0.0001184$$

```
> M:=0.1/G; #M1=M2=M
```

$$M := 844.5945946$$

Mass Distribution Function

```
> r_max:=0.7; #maximum radius of cluster
```

$$r\_max := 0.7$$

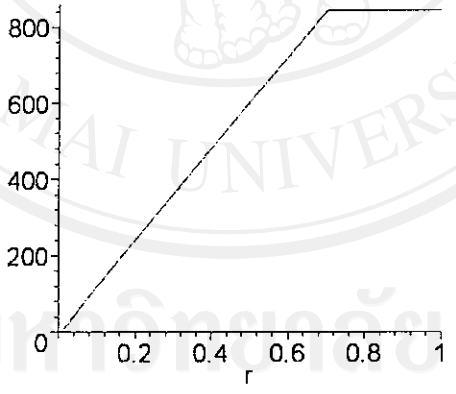
```
> a:=50; #distribution factor
```

$$a := 50$$

```
> mdf:=r->M*piecewise(r<=r_max,(1-exp(-a*r))*r/r_max,r>r_max,1);
```

$$mdf := r \to M \,\text{piecewise}\left(r \le r\_max, \frac{(1 - e^{(-a r)})\, r}{r\_max}, r\_max < r, 1\right)$$

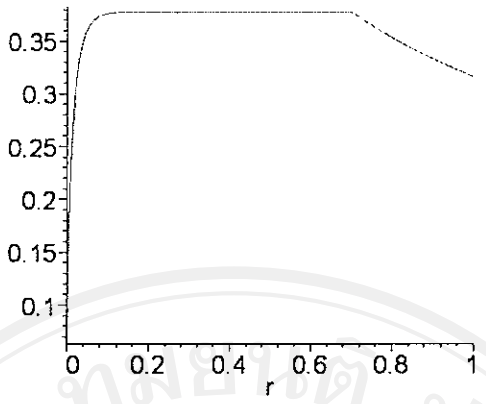
```
> plot(mdf(r),r=0..1); #mass distribution
```




```
> v:=r->sqrt(G*mdf(r)/r); #circular velocity of particle
```

$$v := r \to \sqrt{\frac{G\,\text{mdf}(r)}{r}}$$

```
> plot(v(r),r=0..1); #velocity distribution
```



Cricular Positions

```
> rx:=(r,angle)->evalf(r*cos(angle*Pi/180)):
> ry:=(r,angle)->evalf(r*sin(angle*Pi/180)):
```

Circular Velocities

```
> vx:=(r,angle)->evalf(-v(r)*sin(angle*Pi/180)):
> vy:=(r,angle)->evalf(v(r)*cos(angle*Pi/180)):
```

Solution

```
> Solutione:=proc(R,ecc,x0,y0,u0,v0)
      local System,V;
      V:=sqrt(G*M*(1-Ecc)/(2*R)); #V->launching azimuthal velocity
  System:={

  diff(X1(t),t,t)=G*M*(X2(t)-X1(t))/((X1(t)-X2(t))^2+(Y1(t)-Y2(t))^2)^(3/2),
  diff(Y1(t),t,t)=G*M*(Y2(t)-Y1(t))/((X1(t)-X2(t))^2+(Y1(t)-Y2(t))^2)^(3/2),

  diff(X2(t),t,t)=G*M*(X1(t)-X2(t))/((X1(t)-X2(t))^2+(Y1(t)-Y2(t))^2)^(3/2),
  diff(Y2(t),t,t)=G*M*(Y1(t)-Y2(t))/((X1(t)-X2(t))^2+(Y1(t)-Y2(t))^2)^(3/2),

  diff(x(t),t,t)=G*M*(X2(t)-x(t))/((X2(t)-x(t))^2+(Y2(t)-y(t))^2)^(3/2)
  +G*mdf(sqrt((X1(t)-x(t))^2+(Y1(t)-y(t))^2))*(X1(t)-x(t))/((X1(t)-x(t))^2+(Y1(t)-y(t
  ))^2)^(3/2),

  diff(y(t),t,t)=G*M*(Y2(t)-y(t))/((X2(t)-x(t))^2+(Y2(t)-y(t))^2)^(3/2)
  +G*mdf(sqrt((X1(t)-x(t))^2+(Y1(t)-y(t))^2))*(Y1(t)-y(t))/((X1(t)-x(t))^2+(Y1(t)-y(t
  ))^2)^(3/2),


  X1(0)= R/2, Y1(0)= 0, D(X1)(0)= 0, D(Y1)(0)= V,
  X2(0)=-R/2, Y2(0)= 0, D(X2)(0)= 0, D(Y2)(0)=-V,

  x(0)=x0+R/2, y(0)=y0,
  D(x)(0)=u0-V*y0/(R/2), D(y)(0)=v0+V*(R/2+x0)/(R/2)

  };
      dsolve(System,numeric,output=listprocedure,maxfun=500000);
  end proc:
```

Ring Construction

```
> rings:=3;
```

$$rings := 3$$

```
> in_ring_radius:=0.44; #innermost ring radius
```

$$in\_ring\_radius := 0.44$$

```
> for k from 1 to rings do r[k]:=in_ring_radius+(k-1)/100; end do;
```

$$r_1 := 0.4400000000$$

$$r_2 := 0.4500000000$$

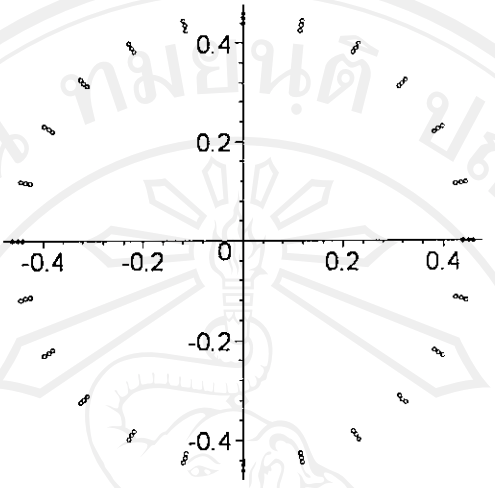
$$r_3 := 0.4600000000$$

```
> phase:=15; N:=24; #phase and total number of particle on the ring
                    #N*phase=360
```

$$phase := 15$$

$$N := 24$$

```
> display(seq(seq(pointplot([r[k],i*phase*Pi/180],symbol=circle,coords=polar),i=0..N)
  ,k=1..rings),scaling=constrained);
```



Initial Conditions

```
> R_max:=10; #distance between clusters
```

$$R\_max := 10$$

```
> Ecc:=0.75; #eccentricity of both clusters
```

$$Ecc := 0.75$$

```
> R_min:=R_max*(1-Ecc)/(1+Ecc);
```

$$R\_min := 1.428571429$$

```
> R_from_CM:=R_min/2;
  radius_of_outermost_ring:=r[rings];
```

$$R\_from\_CM := 0.7142857145$$

$$radius\_of\_outermost\_ring := 0.4600000000$$

```
> # HDR Induction occurs when R_from_CM >> radius_of_outermost_ring
  # Mass transfer occurs when R_from_CM ~ radius_of_outermost_ring
> a:=R_max/(1+Ecc); #semi-major axis
```

$$a := 5.714285714$$

```
> tau:=evalf(sqrt(4*Pi^2*a^3/G/(M+M))); #period of both orbits
```

$$\tau := 191.9145038$$

```
> x0:=(i,k)->rx(r[k],i*phase):
  y0:=(i,k)->ry(r[k],i*phase):
> u0:=(i,k)->vx(r[k],i*phase):
  v0:=(i,k)->vy(r[k],i*phase):
```

Points Evaluation

```
> S:=(i,k)->Solutione(R_max,Ecc,x0(i,k),y0(i,k),u0(i,k),v0(i,k));
```

$$S := (i,k) \rightarrow \text{Solutione}(R\_max, Ecc, \text{x0}(i,k), \text{y0}(i,k), \text{u0}(i,k), \text{v0}(i,k))$$

```
> MMRest:=
  [eval(X1(t),S(1,1)),eval(Y1(t),S(1,1))],
  [eval(X2(t),S(1,1)),eval(Y2(t),S(1,1))]:

  RingRest:=seq(seq([eval(x(t),S(i,k)),eval(y(t),S(i,k))],i=0..N),k=1..3):
```

```
MarkRest:=seq([eval(x(t),S(N,k)),eval(y(t),S(N,k))],k=1..3):

MMRot:=eval(sqrt(X1(t)^2+Y1(t)^2),S(1,1)):

RingRot:=seq(seq([
eval(x(t)*X1(t)/sqrt(X1(t)^2+Y1(t)^2)+y(t)*Y1(t)/sqrt(X1(t)^2+Y1(t)^2),S(i,k)),
eval(-x(t)*Y1(t)/sqrt(X1(t)^2+Y1(t)^2)+y(t)*X1(t)/sqrt(X1(t)^2+Y1(t)^2),S(i,k))]
,i=0..N),k=1..3):
```

Simulation Display

```
> tau/2;
```

$$95.95725190$$

```
> Start:=85; Finish:=105; Smooth:=5; Step:=(Finish-Start)*Smooth;
```

$$Start := 85$$

$$Finish := 105$$

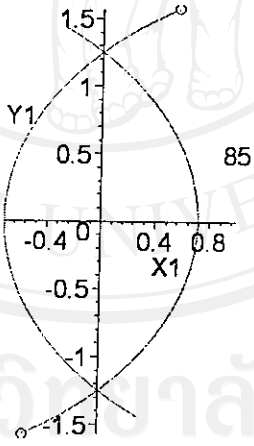
$$Smooth := 5$$

$$Step := 100$$

Tracks

```
> CENTER_REST:=display(seq(pointplot({MMRest(Start+j/Smooth)},symbol=circle,symbol
  size=20),j=0..Step),insequence=true):

  TRACK1:=odeplot(S(N,1),[X1(t),Y1(t)],Start..Finish,numpoints=300):

  TRACK2:=odeplot(S(N,1),[X2(t),Y2(t)],Start..Finish,numpoints=300):

  T_REST:=display(seq(textplot([0.2*R_max/2,0.1*R_max/2,round(Start+j/Smooth)]),j=
  0..Step),insequence=true):
```

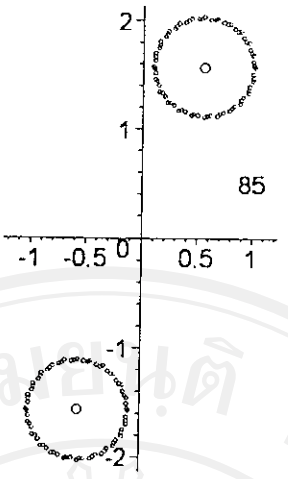
```
> display(CENTER_REST,TRACK1,TRACK2,T_REST, scaling=constrained );
```



Rest View

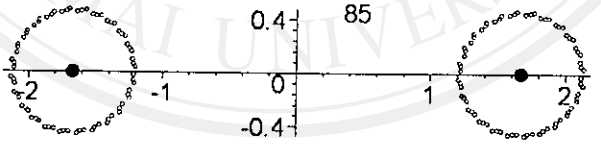
```
> RING_REST1:=display(seq(pointplot({RingRest(Start+j/Smooth)},symbol=circle,symbo
  lsize=8,color=blue),j=0..Step),insequence=true):

  RING_REST2:=display(seq(pointplot({-RingRest(Start+j/Smooth)},symbol=circle,symb
  olsize=8,color=blue),j=0..Step),insequence=true):
```

```
> display(CENTER_REST,RING_REST1,RING_REST2,T_REST,scaling=constrained);
```



Rot View

```
> CENTER_ROT1:=display(seq(disk([MMRot(Start+j/Smooth),0],0.05,color=black),j=0..S
  tep),insequence=true):

  CENTER_ROT2:=display(seq(disk([-MMRot(Start+j/Smooth),0],0.05,color=black),j=0..
  Step),insequence=true):

  RING_ROT1:=display(seq(pointplot({RingRot(Start+j/Smooth)},symbol=circle,symbols
  ize=8,color=blue),j=0..Step),insequence=true):

  RING_ROT2:=display(seq(pointplot({-RingRot(Start+j/Smooth)},symbol=circle,symbol
  size=8,color=blue),j=0..Step),insequence=true):

  T_ROT:=display(seq(textplot([r[3],r[3],round(Start+j/Smooth)]),j=0..Step),insequ
  ence=true):
> display(CENTER_ROT1,CENTER_ROT2,RING_ROT1,RING_ROT2,T_ROT,scaling=constrained);
```




```
>
```

CURRICULUM VITAE

Name Krisada Rawiruswattana

Date of Birth 16 December 1980

Education B.S. (Physics), Chiang Mai University (2002)