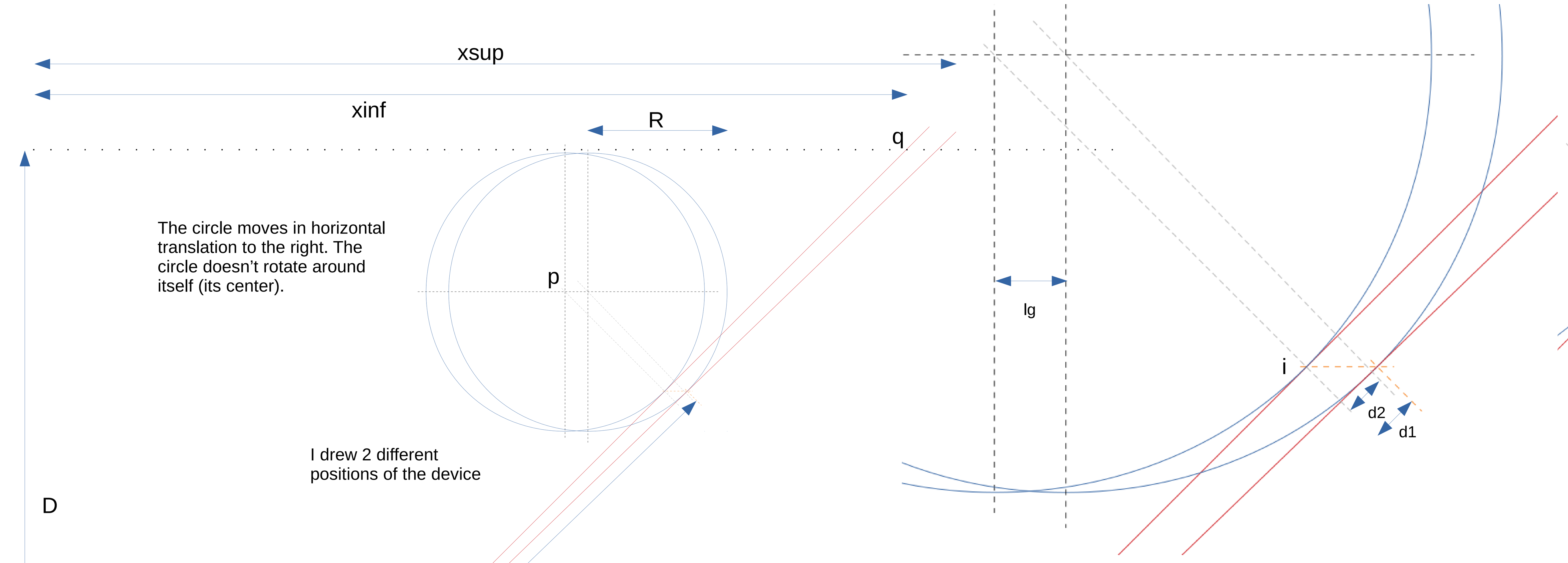
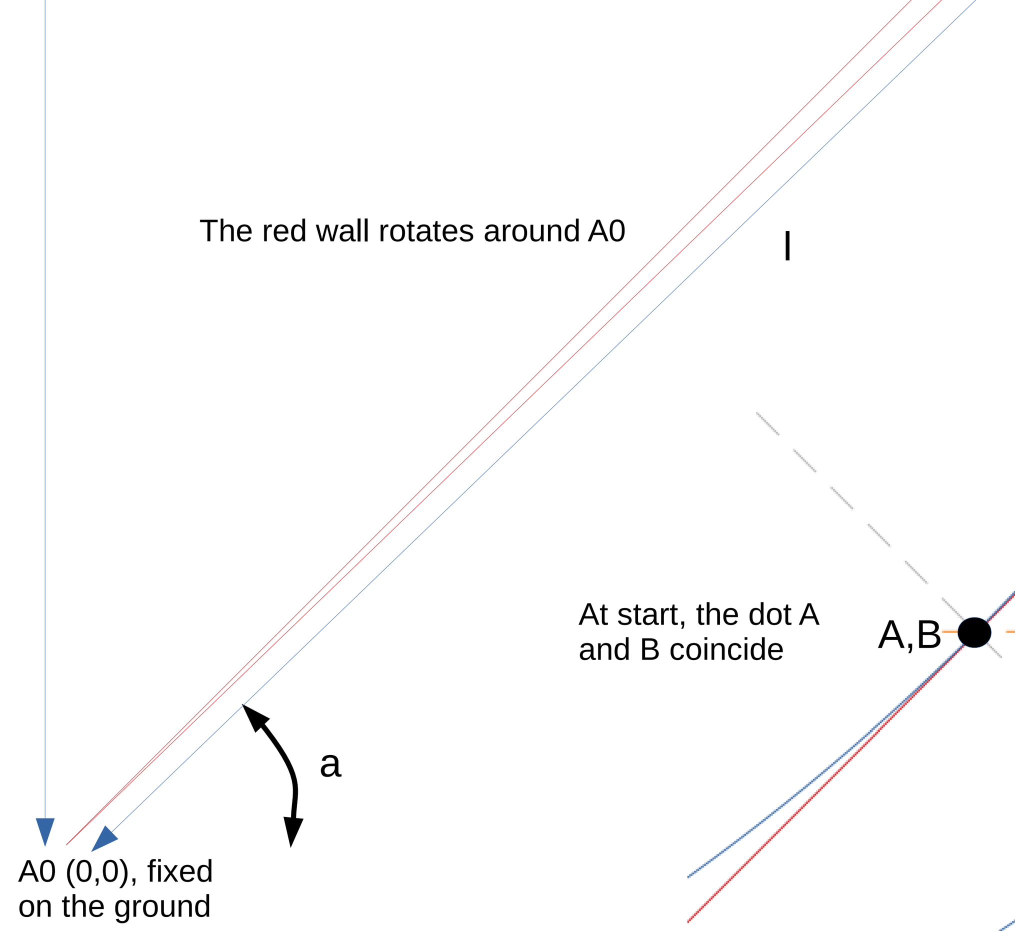
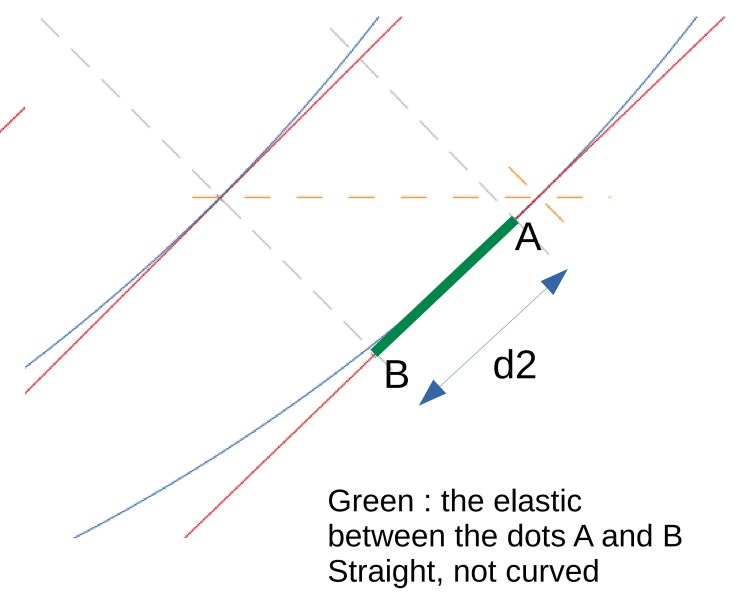
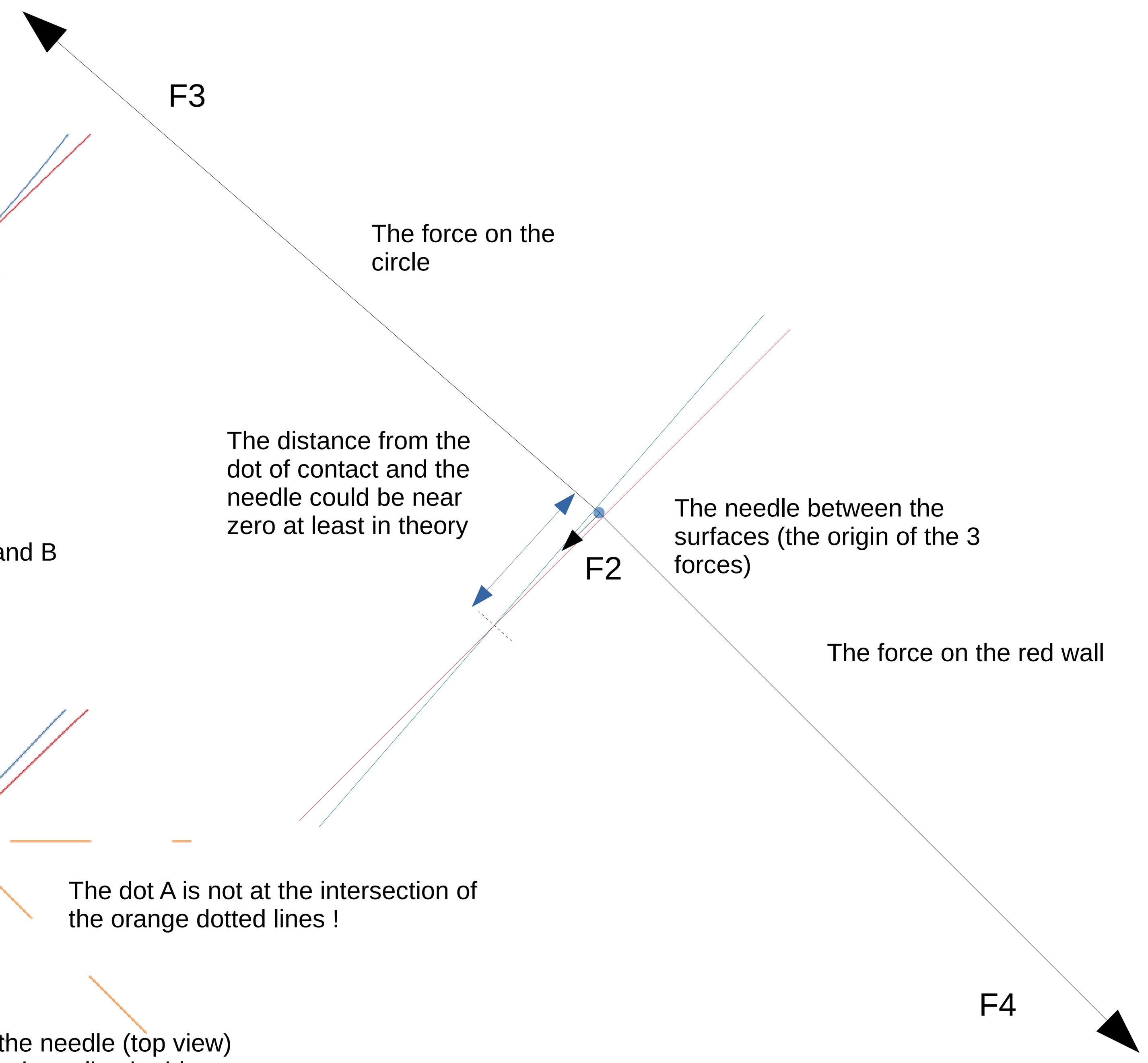
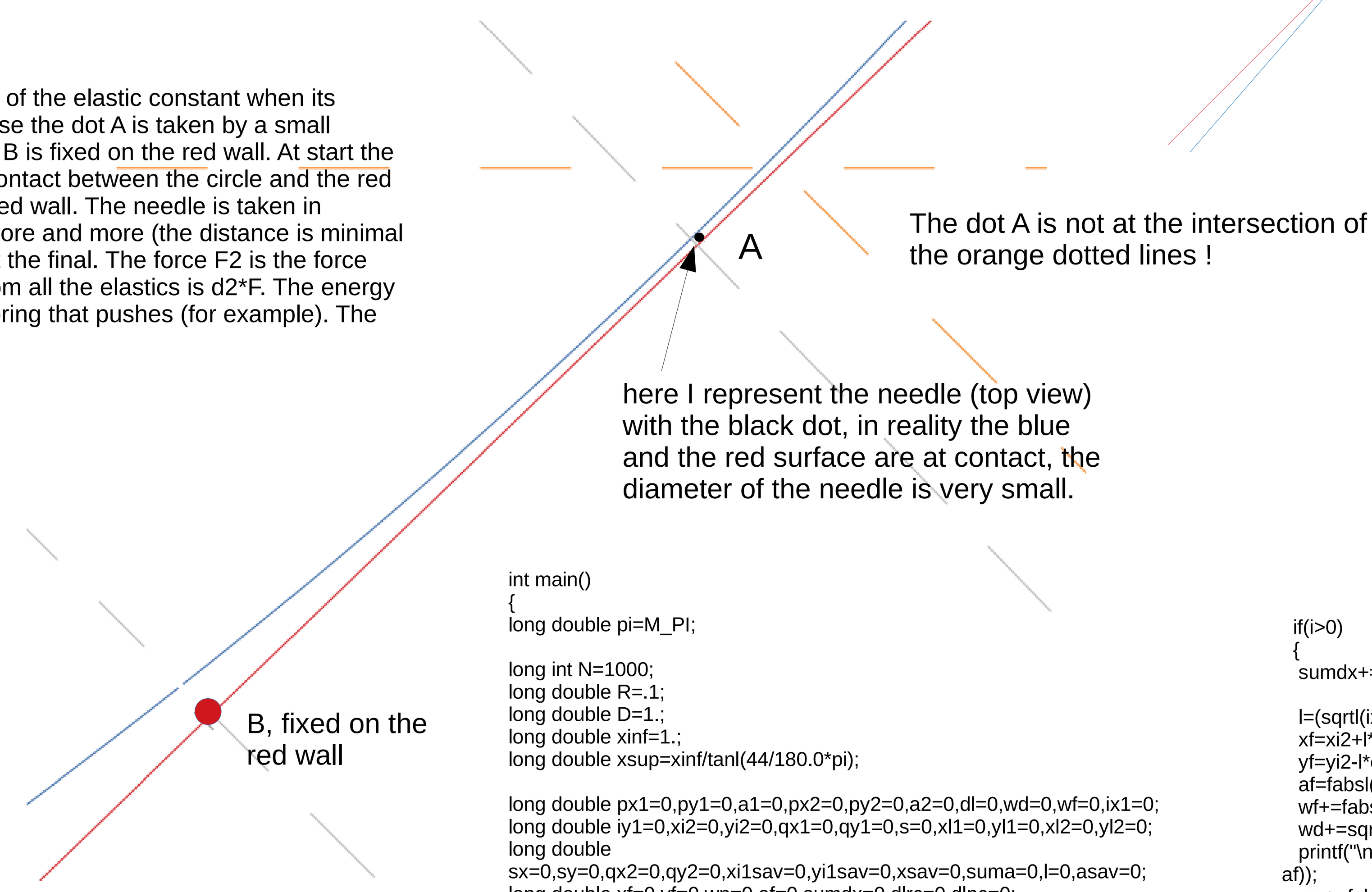


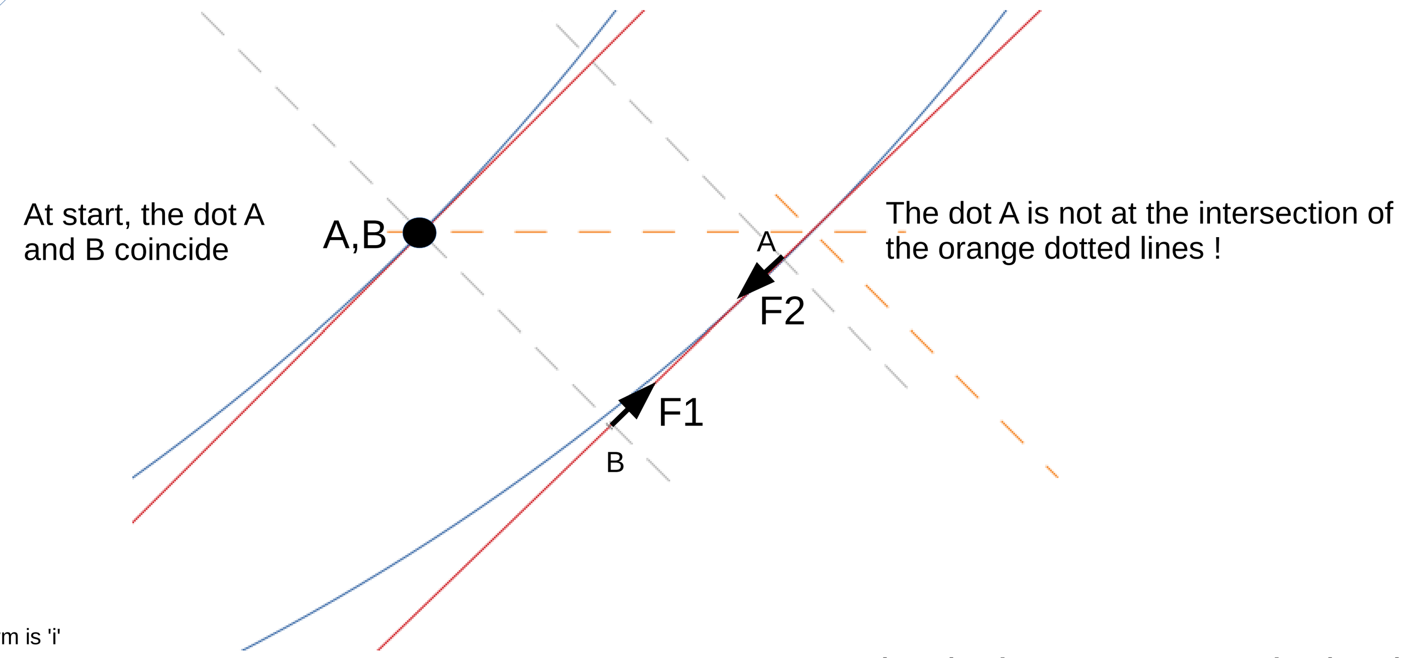






An elastic is attached between the dot A and the dot B, the elastic is straight (part of a line not curved). I suppose the force F of the elastic constant when its length changes, at least for the difference of length of that example. The dot A is between the circle and the red wall, I suppose the dot A is taken by a small needle (in theory the diameter of the needle is 0), it is to be at the dot of contact between the circle and the red wall. The dot B is fixed on the red wall. At start the length of the elastic is 0 and at final it is d2 : at start the dots A and B coincide. The two ends of the elastic are at the dot of contact between the circle and the red wall, but one end is fixed on the needle (and the needle is between the circle and the red wall) the other end is fixed on the red wall. The needle is taken in sandwich between the two surfaces and cannot move closer the dot B because the distance between the walls decreases more and more (the distance is minimal at the dot of contact). The elastic increases its length of d2 not d1. The force F1 is the force from the elastic to the red wall at the final. The force F2 is the force from the elastic to the needle. The needle gives the forces F3 and F4 to the circle and the red wall. The energy recovered from all the elastics is d2*F. The energy needed to move the circle is d1*F = lg*cos(a)*F. Destroy or create the energy depends if I use a pulling force (elastic) or a spring that pushes (for example). The difference of energy is (d2-d1)*F, it is the energy destroyed or created.





The (0,0) is on A0
The dot of contact between the circle and the red arm is 'i'
The center of the circle is the dot 'p'
The intersection between the upper horizontal line and the red arm is the dot 'q'
The height from A0 to the top of the circle is 'D'
The radius of the circle is 'R'
The length of the distance from A0 to 'i' is 'l' (small L)
I integrate from 'xinf' to 'xsup'

For the dots i,p,q, I added a letter for the 'x' or the 'y' and I added an information for take in account the step: the number '1' is the step in calculation and the '2' is the last calculation.

'wf' is the energy to move the circle in translation
'wd' is the energy recovered from the friction
'wp' is to verify the vector of the force of friction is like the orientation of the red wall, must be at 0

Pure mechanical counterexample that breaks the law of conservation of the energy
I don't need the mass nor an external gravity, it is a pure geometric example
In that example, I don't need the friction

```
int main()
{
long double pi=M_PI;

long int N=1000;
long double R=.1;
long double D=1.;
long double xinf=1.;
long double xsup=xinf/tanl(44/180.0*pi);

long double px1=0,py1=0,a1=0,px2=0,py2=0,a2=0,dl=0,wd=0,wf=0,ix1=0;
long double iy1=0,xi2=0,yi2=0,qx1=0,qy1=0,s=0,xl1=0,yl1=0,xl2=0,yl2=0;
long double
sx=0,sy=0,qx2=0,qy2=0,xi1sav=0,yi1sav=0,xsav=0,suma=0,l=0,asav=0;
long double xf=0,yf=0,wp=0,af=0,sumdx=0,dlrc=0,dlpc=0;

long int i;
qy1=D;

for(i=0;i<N;i++)
{
  //a1=(long double)(asup/180.0*pi-ainf/180.0*pi)/N*i;
  qx1=xinf+(xsup-xinf)/N*i;
  a1=atanl(qy1/qx1); if(i==0) a2=a1;
  s=R/tanl(a1/2.);
  px1=qx1-s;
  py1=qy1-R;
  ix1=px1+R*sinl(a1);
  iy1=py1-R*cosl(a1);
  if(i<1) {xi1sav=ix1;yi1sav=iy1;xsav=px1;asav=a1;}
  printf("\npx1=%Lf , py1=%Lf , a=%Lf , ix1=%Lf , iy1=%Lf ,
a1=%Lf",px1,py1,180/pi*atanl((fabsl(iy1-yi2)-fabsl(py1-py2))/(fabsl(ix1-xi2)-
fabsl(px1-px2))),ix1,iy1,180/pi*a1);
  suma+=a1;

  if(i>0)
  {
   sumdx+=(fabsl(px1-px2)-fabsl(ix1-xi2))*cosl(a1);

   l=(sqrtl(ix1*ix1+iy1*iy1)+sqrtl(xi2*xi2+yi2*yi2))/2.;
   xf=xi2+l*sinl(a1)*fabsl(a2-a1);
   yf=yi2-l*cosl(a1)*fabsl(a2-a1);
   af=fabsl(atanl((iy1-yf)/(ix1-xf)));
   wf+=fabsl(px2-px1)*cosl(af);
   wd+=sqrtl((ix1-xf)*(ix1-xf)+(iy1-yf)*(iy1-yf))*cosl(a1-af);
   printf("\nl=%Lf , xf=%Lf , yf=%Lf , af=%Lf , diffa=%Lf",l,xf,yf,180/pi*af,180/pi*(a1-
af));
   wp+=fabsl(sinl(a1-af))*l*fabsl(a2-a1);
   dlpc+=(px2-px1)*sinl(a1);
   dlrc+=l*(a2-a1);

  }
  px2=px1;  py2=py1;  qx2=qx1;  qy2=qy1;  xi2=ix1;  yi2=iy1;  xl2=xl1;  yl2=yl1;
a2=a1;

}
printf("\ndlpc=%Lf , dlrc=%Lf",dlpc,dlrc);
printf("\nN=%ld , R=%Lf , D=%Lf , xinf=%Lf , xsup=%Lf \ndlp=%Lf , dli=%Lf , suma=
%Lf , l=%Lf , cos=%Lf , wf=%Lf , wd=%Lf , wp=%Lf , diff=%Lf , eff=%Lf , ddx=%Lf ,
sumdx=%Lf\n",N,R,D,xinf,xsup,px1-xsav,ix1-xi1sav,(asav-a1)*180/pi,l,cosl(suma/
N),wf,wd,wp,wf-wd-wp,(wd+wp)/wf, fabsl(px1-xsav)-fabsl(ix1-xi1sav),sumdx);
return 0;
}
```