```
from math import *
import matplotlib.pyplot as plt
##########TARTAGLIA#########
m=0.050
g=9.81
V0=20
N=100

def tartaglia(theta,beta,tmax,N):
    h = tmax/N
    t = [0+k*h for k in range (N+1)]
    x = [0]
    y = [0]
    Vx = [V0*cos(theta)]
    Vy = [V0*sin(theta)]
    i=0
    while i<=N and y[i]>=0:
        x.append(x[-1]+h*Vx[-1])
        y.append(y[-1]+h*Vy[-1])
        Vx.append(Vx[-1] + h*(-beta/m)*sqrt(Vx[-1]**2 +
Vy[-1]**2)*Vx[-1])
        Vy.append(Vy[-1] + h*(-g-(beta/m)*sqrt(Vx[-2]**2 +
Vy[-1]**2)*Vy[-1]))
        i+=1
    return x,y
x,y = tartaglia(pi/4,0.1,10,1000)
#plt.plot(x,y)
#plt.show()

##Calcul de la portée du tir
def portee(theta,beta):
        x ,y = tartaglia(theta,beta,10,1000)
        return x[-1]
###Tracé de la portée du tir
#alpha=[k*pi/2/1000 for k in range(1,1001)]
#port=[portee(a,0.1) for a in alpha]
#plt.clf()
#plt.plot(alpha,port)
#plt.show()
#pm=max(port)
#ind=port.index(pm)
#print(alpha[ind]*180/pi) ###en degrés

##Tracer de la courbe theta_max en fonction de beta
beta=[k*0.2/100 for k in range(101)]
alpha=[k*pi/2/100 for k in range(1,101)]
theta_m=[]
for val in beta:
  portee2=[]
  for el in alpha:
    portee2.append(portee(el,val))
  pm=max(portee2)
  ind=portee2.index(pm)
  theta_m.append(alpha[ind]*180/pi)
```

```
###Lissage de la courbe####
def lissage(Lx,Ly,p):
    '''Fonction qui débruite une courbe par une moyenne glissante
    sur 2P+1 points'''
    Lxout=[]
    Lyout=[]
    for i in range(p,len(Lx)-p):
        Lxout.append(Lx[i])
    for i in range(p,len(Ly)-p):
        val=0
        for k in range(2*p):
            val+=Ly[i-p+k]
        Lyout.append(val/2/p)
    return Lxout,Lyout

xx,yy=lissage(beta,theta_m,15)
plt.plot(xx,yy)
plt.show()

#########DICHOTOMIE########
M=20
l0=0.8
d=1.7
k=2000

def f1(z):
        return M*g-2*k*z*(1-l0/sqrt(d**2/4+z**2))

def dicho(f,a,b,N):
        gg=a
        dd=b
        for i in range(N):
                m=(gg+dd)/2
                if f(m)*f(dd)<=0:
                        gg=m
                else:
                        dd=m
        return m

###Au maximum (quand d=0) la masse descend jusque à z=l0+Mg/2k
zm=l0+M*g/(2*k)
zlist=[zm/1000*k for k in range(1001)]
f1list=[f1(z) for z in zlist]
#plt.plot(zlist,f1list)
#plt.xlim(0.3,0.4)
#plt.grid()
#plt.show()

print(dicho(f1,0,zm,10000))
l=sqrt(d**2/4+zm**2)
print(asin(zm/l)*180/pi)
```