```python
import numpy as np
import matplotlib.pyplot as plt



def modelo(dr,n):

    l_P = []
    l_rho = []
    l_m = []
    l_r = []


    rho = 1
    P = rho**((float(n)+1)/n)

    m=0
    dm=0
    dP=0
    r=0


    while P> 0:
        cons=3*rho*(r**2) #conservacion de masa
        dm= cons*dr
        m=m + dm
        l_m.append(m)


        if r>=0.001:
            hidro = -(rho*m)/(r**2)

        if r < 0.001:
            hidro = -r


        dP = dr*hidro
        P = P + dP


        if P < 0:

            l_P.append(0)
            l_rho.append(0)
            l_r.append(r)


        if P>=0:

            l_P.append(P)
            rho = (P)**(1/((float(n)+1)/n))
            l_rho.append(rho)
            r = r + dr
            l_r.append(r)


        for i in range (0,len(l_r)): #encontrar el maximo
            a=l_r[i]

            if a>l_r[i-1]:
                maxr=l_r[i]
```

```
    return   np.array(l_r)/maxr , np.array(l_rho)



r_32,rho_32 = modelo(0.001,3/2)


r_s, rho_s = np.loadtxt("sol.dat",usecols=(0,3),unpack=True)


for i in range (0,len(rho_s)): #encontrar el maximo
    a=rho_s[i]
    if a>rho_s[i-1]:
        maxrho=rho_s[i]



plt.plot(r_32,rho_32,"r",label="n=3/2")

plt.plot(r_s,rho_s/maxrho,"y",label="sol")

plt.legend()
plt.ylabel(r"$\rho / \rho_c$")
plt.xlabel(r"$r / r_o$")

plt.show()
```