

```

import numpy as np
import matplotlib.pyplot as plt

def modelo(dr,n): #las variables de entrada serán el diferencial de radio
                  #y el n que se quiere usar

    l_P = [] #presion
    l_rho = [] #densidad
    l_m = [] #masa
    l_r = [] #radio

    rho = 1 # ya que partimos desde el centro y normalizamos,
    #rho=1 quiere decir que iniciamos en el centro (0,0) de la estrella
    nn=float(n)
    P = rho**((nn+1)/nn) #ecuacion de estado politropa

    m=0 #en el (0,0) la masa es 0
    dm=0
    dP=0
    r=0 #en el (0,0) r=0

    while P> 0: #p>0 indica que aun estamos dentro de la estrella
    #ya que si recordamos de clase, cuando llegamos a la superficie estelar
    #densidad=0

        cons=3*rho*(r**2) #conservacion de masa
        dm= cons*dr #razon de diferencial de masa
        m=m + dm#incremento de la masa
        l_m.append(m)

        if r>=0.001:
            hidro = -(rho*m)/(r**2)
            #0.001 será nuestro punto borde, ya que en el centro estelar
            #el radio es muy pequeño luego la ecuacion hidrostática
            #se indefinir
        if r < 0.001:
            hidro = -r #aproximación lineal (no afectara mucho los resultados)

        dP = dr*hidro #razon del diferencial de presion(<0)
        P = P + dP #caida de la presion a medida que aumenta el radio

        if P>=0:#seguimos dentro de la estrella

            l_P.append(P) #luego hay presion
            rho = (P)**(1/((nn+1)/nn)) #hay densidad, dada por politropo
            l_rho.append(rho)
            r = r + dr #radio incrementa
            l_r.append(r)

        if P < 0:#si la presion es menor a 0 estamos fuera de la estrella

            l_P.append(0) #fuera de la estrella no habra presion
            l_r.append(r) #en este punto se definira el radio maximo
            l_rho.append(0) #consideramos vacio fuera de la estrella

    for i in range (0,len(l_r)): #encontrar el maximo del radio para
    #normalizar nuevamente
        a=l_r[i]

        if a>l_r[i-1]:

```

```

maxr=l_r[i]

return np.array(l_r)/maxr , np.array(l_rho) #solo nos interesa rho y r

r_32,rho_32 = modelo(0.001,1.5) #intervalo,n
r_43,rho_43 = modelo(0.001,1.33333)
r_53,rho_53 = modelo(0.001,1.66667)
r_340,rho_340 = modelo(0.001,3.4)

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

for i in range (0,len(rho_s)): #normalizar densidad solar(encontrar max)
    a=rho_s[i]
    if a>rho_s[i-1]:
        maxrho=rho_s[i]

plt.plot(r_s,rho_s/maxrho,"y",label="sol")
plt.plot(r_32,rho_32,"r",label="n=3/2")
plt.plot(r_43,rho_43,"b",label="n=4/3")
plt.plot(r_53,rho_53,"g",label="n=5/3")
plt.plot(r_340,rho_340,"m",label="n=3.4")

plt.legend()
plt.ylabel(r"$\rho / \rho_c$")
plt.xlabel(r"$r / r_o$")
plt.title("$\Delta=0.001$")

plt.show()

```