```
import numpy as np
import math
import matplotlib.pyplot as plt
import matplotlib as mpl
from scipy import stats
import pylab as pl
import pylab
from pylab import *
from matplotlib import rc


def stellar_structure(n,dr):
        p = 1                   #Se definen aquí los valores iniciales.
        y = (float(n)+1)/n      #considerando que se parte desde el centro estelar.
        P = (p)**(y)            #definimos ecuacion de la polítropa.
        r=0
        dr=dr
        m=0
        dm=0
        dP=0


        lista_P = [] #presion
        lista_p = [] #densidad
        lista_m = [] #masa
        lista_r = [] #radio


        while P > 0:                #mientras la presión sea mayor a cero(sigamos en la estrella)
                f = 3*p*(r**2) # f=dm_n / dr_n (conservacion de masa)
                dm = f*dr #crece el diferencial de masa
                m =m + dm # aumenta la masa total contenida
                lista_m.append(m) #guardamos la masa total en cada intervalo



                if r < 0.004: #condición en borde (para que no explote)
                        g = -r #g es la ecuación de equilibrio hidrostático
```

```
            else:
                g = -(p*m)/(r**2) # ecuacion hidrostatica
                dP = g*dr #decrece la presion (g<0)
                P += dP #presion acumulada
            if P < 0: #si la presion se vuelve menor a 0 (el programa terminará
                                                        #en la proxima iteración)
                lista_P.append(0)        #la presion exrerior es 0 fuera de la estrella
                lista_p.append(0)        #la densidad exterior es 0 //
                lista_r.append(r)        #el radio exterior es el radio exterior


            else: #si la presion es >=0 (el programa continúa en la proxima iteración)
                lista_P.append(P) #agregamos presion a lista
                p = (P)**(1/y) #rho cambia por nuevo rho (politropa)
                lista_p.append(p) #agregamos este rho a la lista
                r += dr #crece radio
                lista_r.append(r) #agregamos nuevo radio


        return np.array(lista_P), np.array(lista_p), np.array(lista_m)/np.max(np.array(lista_m)),
        np.array(lista_r)/np.max(np.array(lista_r))

inc=0.3
P1,p1,m1,r1 = stellar_structure(1.5,inc)
P2,p2,m2,r2 = stellar_structure(1.33334,inc)
P3,p3,m3,r3 = stellar_structure(1.666667,inc)
P4,p4,m4,r4 = stellar_structure(3,inc)
r_sol, p_sol = np.loadtxt("sol.dat",usecols=(0,3),unpack=True)
plt.figure()
plt.plot(r4,p4,linestyle='-',color="r",label=r"$n = 3$")
plt.plot(r1,p1,linestyle='-',color="k",label=r"$n = 3/2$")
plt.plot(r2,p2,linestyle='-',color="b",label=r"$n = 4/3$")
plt.plot(r3,p3,linestyle='-',color="g",label=r"$n = 5/3$")
plt.plot(r_sol,p_sol/np.max(p_sol),color="y",linestyle='--',linewidth = 1.2,label=r"Sol")
plt.xlabel(r"$\frac{r}{R_{o}}$",fontsize=24)
plt.ylabel(r"$\frac{\rho}{\rho_{c}}$",fontsize=24)
plt.title("Intervalos $\Delta r$ = 0.3")
plt.legend()
plt.show()
```