AST 231: Problem Set 5

Gil Garcia

April 9, 2020

Problem 1. Low mass white dwarfs have an equation of state that is independent of temperature, since they are entirely supported by electron degeneracy pressure. The fact that they are low mass means that relativistic effects do not need to be taken into account because the momenta of the supporting particles (electrons) are small enough. In the case of non-relativistic (complete) degeneracy, the equation of state can be written as:

$$P = K\rho^{\frac{5}{3}}$$

The value of K in this case is given in the Lecture Notes for Lecture 13 (non-relativistic case).

a) Integrate the equations of stellar structure (hydrostatic equilibrium and mass conservation) to determine the mass and radius of a white dwarf that has a central density of 10^5 gm cm^{-3}.

b) Find the mean density of this star.

c) Give its mass in terms of solar masses and its radius in terms of both solar radii and Earth radii.

[Note: For the numerical integration it is fine to use a simple "Newton's method". You can vary the integration step size to see how it affects results. If any of you wish to do a more sophisticated integration, say using the Runge-Kutta or other method, you are most welcome to do so, but it is not necessary.]

Answer 1. (a) We want to integrate the equation of hydrostatic equilibrium:

$$\frac{\mathrm{dP}}{\mathrm{dr}} = \frac{-\mathrm{GM}_r\rho}{\mathrm{r}^2} \tag{1}$$

and the equation of mass conservation:

$$\frac{\mathrm{dM}}{\mathrm{dr}} = 4\pi\mathrm{r}^2\rho \tag{2}$$

To do this, we need the boundary conditions of pressure at r=0 and at r=R. We know that P(r=R) = 0 and because of the equation state, P(r=0) = $\mathrm{P}_c = K\rho_c^{5/3}$ where

$$K = \frac{h^2}{5m_e}\left(\frac{3}{8\pi}\right)^{\frac{2}{3}}\left(\frac{1}{2m_H}\right)^{\frac{5}{3}}$$

for the non-degenerate case. We integrate the two equations using the Newton Integration method. This starts by defining our initial conditions: r=0, m=0, $\rho_c = 10^8$ kg m^{-3}, and P = K$\rho_c^{5/3}$. We take a small step in radius (dr) and then recalculate and update our r,m,ρ, and P values. The new r value is simply $r_{\text{new}} = r + \text{dr}$. To calculate the new density, we use $\rho_{\text{new}} = (\frac{P}{K})^{3/5}$. We then use the two updated values to calculate the change in mass. That is,

dM $= 4\pi r_{\text{new}}^2 \rho_{\text{new}}$dr.

We update the mass value $m_{\text{new}} = m + \text{dM}$ and then we calculate the change in pressure:

dP $= \dfrac{Gm_{\text{new}}}{r_{\text{new}}^2}\rho_{\text{new}} dr$.

Finally, we update our pressure value $P_{\text{new}} = p - \text{dP}$. We repeat this process until P becomes 0. At this point, we will have reached the surface of the star. All this is done in Python (code attached at the end).

We use this to find the mass and radius of a white dwarf with central density of 10^8 kg m^{-3}:

$$\boxed{\text{m} = 3.07 \times 10^{29} \text{ kg and r} = 1.64 \times 10^7 \text{meters}}$$

.

(b)

We take collect the density value at each step of the Newton integration, and we find that the mean density is

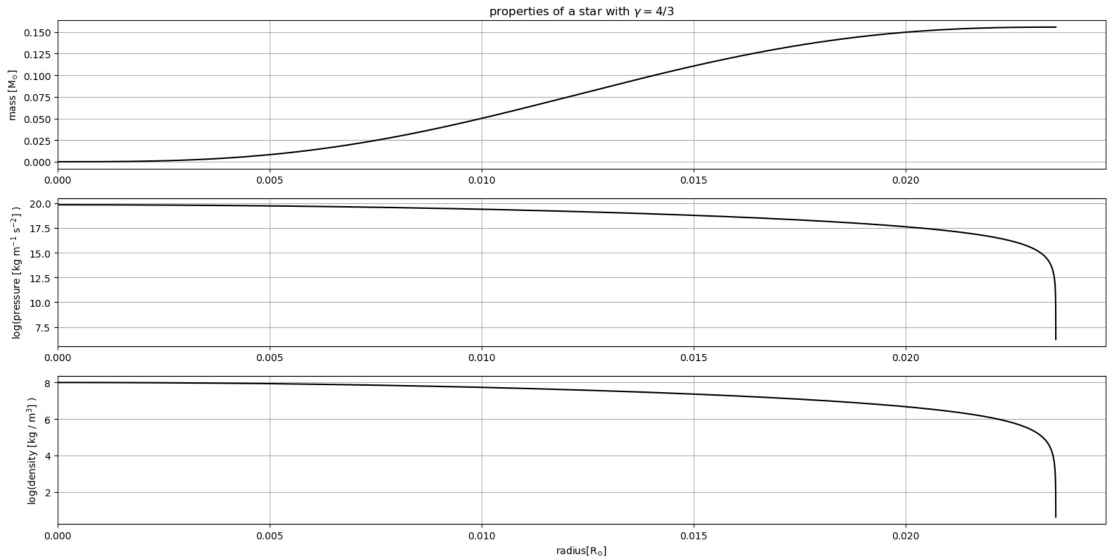
$$\boxed{\text{mean density } = \overline{\rho} = 4.62 \times 10^7 \text{ kg m}^{-3}}$$

(c)

Finally, using $M_\odot = 1.98 \times 10^{30}$ kg and $R_\odot = 6.963 \times 10^8$ m, we calculate the mass and radius in solar units. This gives us:

$$\boxed{\text{m} = 0.155 \text{ M}_\odot \text{ and r} = 0.0235\text{R}_\odot}$$

The properties found in parts (a) - (c) of the white dwarf are nicely summed up in the following plot:



Here, the mass (in terms of solar masses), the pressure, and the density are shown as a function of distance from the center (radius).

Problem 2. Now consider the case of a very high mass white dwarf, in which relativistic effects result in a somewhat different equation of state, namely the one applicable to complete ultra-relativistic degeneracy. From the lecture notes, we have in this case:

$$P = K\rho^{\frac{4}{3}}$$

Again, see the lecture notes for Lecture 13 to get the value of K (ultra-relativistic case), which is different from the value for non-relativistic degeneracy.

a) As before, integrate the equations of hydrostatic equilibrium and the mass equation to determine how the mass and radius of a star depend on central density over the range of 10^9 gm cm^{-3} to 10^{15} gm cm^{-3}. Note that those higher densities are about equal to the density of a neutron!

b) What is the maximum possible mass of a white dwarf star, according to your calculations? Note that you have calculated the Chandrasekhar mass limit for a white dwarf.

Answer 2. (a) We use a very similar Newton integrator as we did for 1 to solve the questions in 2. This time, instead, as our formula for pressure is

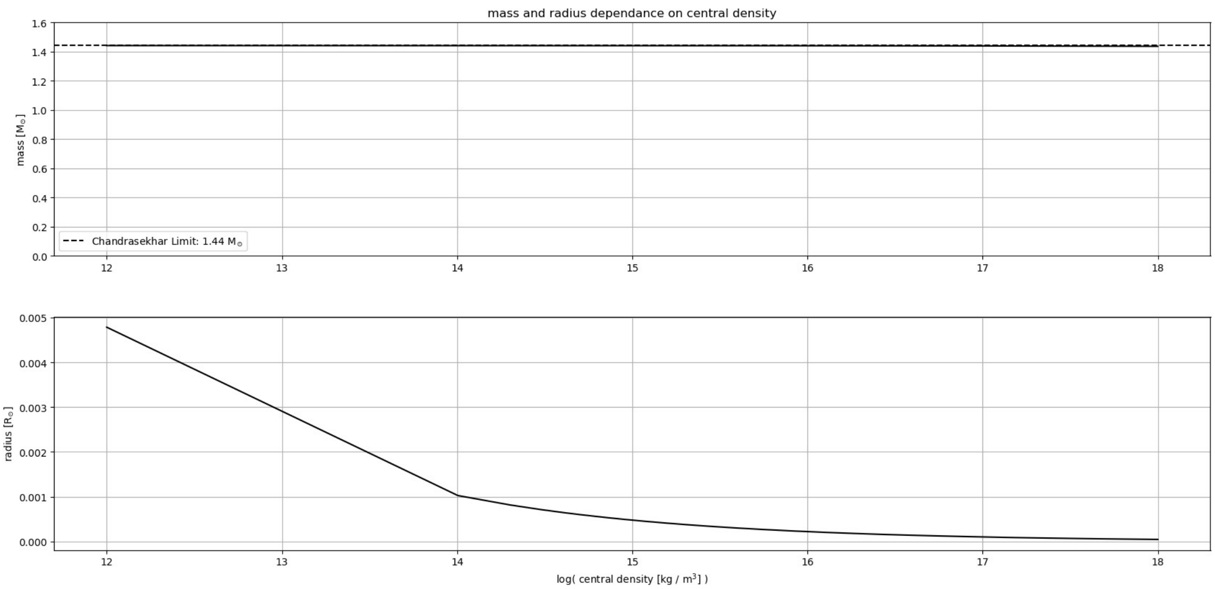
$$P = K\rho^{4/3} \tag{3}$$

where K is now in ultra-relativistic form. That is,

$$K = \frac{hc}{4}\left(\frac{3}{8\pi}\right)^{\frac{1}{3}}\left(\frac{1}{2m_H}\right)^{\frac{4}{3}}$$

.

Rather than doing just one Newton integration, we will be doing a lot of them varying only the central density from 10^{12} kg m^{-3} to 10^{18}. Doing so results in the following relationships between mass versus central density and radius versus central density:

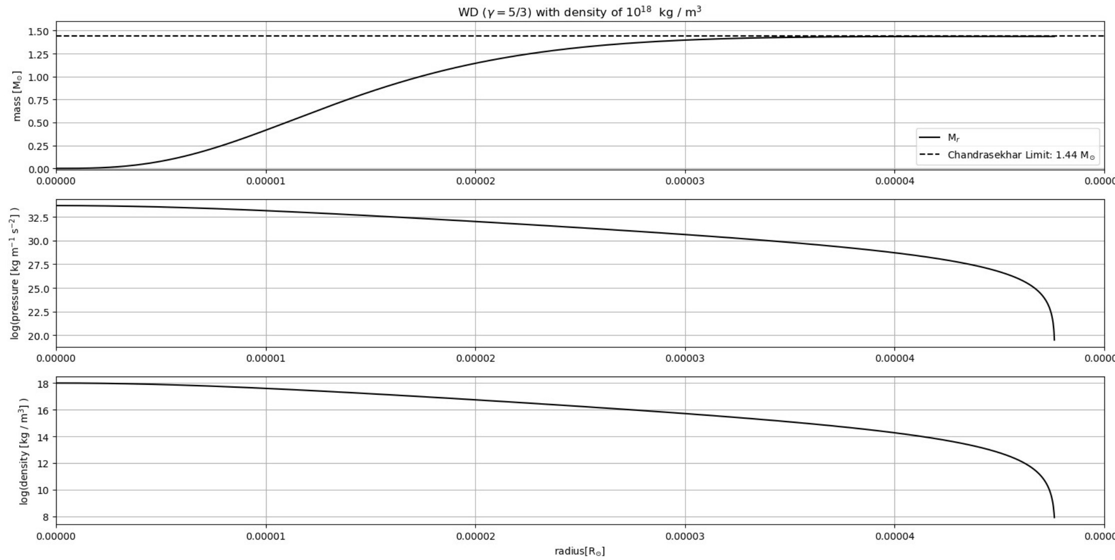


As we can see from the plots, as central density increases, there is no change in the maximum mass of a white dwarf with ultra relativistic effects. In 2b, we see that the maximum mass attainable is a special value. From the plots, we also see that as central density increases, the radius of the white dwarf decreases. There is a limit on how much degeneracy pressure a white dwarf (this is due to the speed of light). Once it is reached, as the mass keeps increasing and therefore, gravity's inwards force increase, there is no outward force from pressure to keep the star from decreasing in size. Therefore, adding more mass (or central density) means the white dwarf will have a smaller radius, as depicted in the plot.

(b) As we can see from the plot as well, the maximum possible mass of a white dwarf is 2.843×10^{30} kg or

$$\boxed{\mathrm{m} = 1.44\mathrm{M}_{\odot}}$$

We can take a closer look at a white dwarf governed by ultra-relativistic effects by looking at the following plot:



In the plot, the mass is asymptotically approaching the Chandrasekhar limit. We also see the changes in density and pressure becomes smaller and smaller as we move farther from the center.

Code for 1

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



DR = 100 # our step size in the integrations


####################
####################          #1
####################


#some constants that will be needed
g = 6.67e-11 #m^3 kg^-1 s^-2
solar_mass = 1.98e30 #kg
solar_radius = 6.963e8 #meters

h = 6.626e-34 #J s (planck constant)
m_e = 9.109e-31 #kg (mass of electron)
m_H = 1.67e-27 #kg (mass of hydrogen)
c = 2.99e8 # m/s (speed of light)



####################
####################          #1a
####################

#we use Newton's method to solve the coupled diff eqns

#first, we define the routines that will be needed in our integrations:

k_non_rel = (h**2 / (5*m_e)) * (3/(8*np.pi))**(2/3) * (1/(2*m_H))**(5/3) #K constant in non-relativistic case

def density(pressure): #density for non-rel case
    return (pressure / k_non_rel)**(3/5)

def mass_step(density,dr): #using mass conservation diff eq to calculate one step in mass
    return 4*np.pi*r**2*density*dr

def pressure_step(mass,radius,density,dr): #using hydrostatic diff eq to calculate one step in pressure
```

```
    return ((g*mass) / radius**2 ) * density * dr


# now we initialize our variables and set them equal to the initial conditions

r = 0 #start at the core, radius is 0
m = 0 #at the core, w a radius of 0, there is no mass enclosed
d = 10**8 #initial density in kg / m^3
p = k_non_rel * (d)**(5/3) #initial pressure at the core
#p = k_ultra_rel * (d)**(5/3)

#intializing lists to store all values at each intigration step

r_lst = []
m_lst =[]
d_lst = []
p_lst=[]


#now we write the integrator using the fact that at the outer boundar, pressure will be 0:
iter=0
while p > 0:
    #keep track of num of steps we take
    iter+=1

    #updating our lsts at each step
    r_lst += [r]
    m_lst += [m]
    d_lst += [d]
    p_lst += [p]

    #now we update the values:
    r = r + DR #updating our radius value by our radius step size
    d = density(p) #calculating the new pressure value
    dm = mass_step(d,DR) #calculating the change in mass
    m = dm + m #updating our mass value
    dp = pressure_step(m,r,d,DR) #calculating the change in pressure
    p = p - dp #updating our pressure value. pressure decreases as we increase radius.



#    print(r,m,d,p)
#    print()


```

```
#results of the integration:
print('# of steps taken:',iter)

print('final radius in meters:\t', format(r,"E") )
print('final mass in kg:\t', format(m,'E') )


print('final radius in r_sun:\t', r / solar_radius )
print('final mass in m_sun\t', m / solar_mass)



#plotting our results:
r_arr = np.array(r_lst) / solar_radius
m_arr = np.array(m_lst) / solar_mass
d_arr = np.array(d_lst)
p_arr = np.array(p_lst)

print('average density in kg per m^3:\t' , format(np.average(d_arr),'E') )

plt.subplot(311)
plt.title(r'properties of a star with $\gamma = 4 / 3$')
plt.plot(r_arr,m_arr,color='k',label=r'M$_r$')
plt.ylabel(r'mass [M$_{\odot}$]')
#plt.axhline(1.44,ls='--',color='k',label=r'Chandrasekhar Limit: 1.44 M$_{\odot}$')
#plt.ylim(-0.01,1.6)
left,right = plt.xlim()
plt.xlim(0,right)
plt.grid()
#plt.legend()


plt.subplot(312)
plt.plot(r_arr,np.log10(p_arr),color='k',label='pressure')
plt.ylabel(r'log(pressure [kg m$^{-1}$ s$^{-2}$] )')
left,right = plt.xlim()
plt.xlim(0,right)
plt.grid()



plt.subplot(313)
plt.plot(r_arr,np.log10(d_arr),color='k',label='density')
plt.ylabel(r'log(density [kg / m$^{3}$] )')
left,right = plt.xlim()
```

```
plt.xlim(0,right)
plt.grid()



plt.xlabel(r'radius[R$_{\odot}$]')
#plt.tight_layout()
plt.show()
print(k_non_rel)
```

Code for 2

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



DR = 10 # our step size in the integrations


####################
####################          #2 an b
####################


#some constants that will be needed
g = 6.67e-11 #m^3 kg^-1 s^-2
solar_mass = 1.98e30 #kg
solar_radius = 6.963e8 #meters

h = 6.626e-34 #J s (planck constant)
m_e = 9.109e31 #kg (mass of electron)
m_H = 1.67e-27 #kg (mass of hydrogen)
c = 2.99e8 # m/s (speed of light)



####################
####################          #2a
####################

#we use Newton's method to solve the coupled diff eqns

#first, we define the routines that will be needed in our integrations:

```

```
k_ultra_rel = ((h*c)/4) * (3/ (8*np.pi))**(1/3) * (1/ (2*m_H))**(4/3) #K constant in ultra-relativistic case
#print(format(k_ultra_rel,'E'))


def density(pressure): #density for ultra-rel case
    return (pressure / k_ultra_rel)**(3/4)

def mass_step(density,r,dr): #using mass conservation diff eq to calculate one step in mass
    return 4*np.pi*r**2*density*dr

def pressure_step(mass,radius,density,dr): #using hydrostatic diff eq to calculate one step in pressure
    return ((g*mass) / radius**2 ) * density * dr



#we define a fxn that will be doing the intergrating

def newton_intergration(initial_density):
    # now we initialize our variables and set them equal to the initial conditions

    r = 0 #start at the core, radius is 0
    m = 0 #at the core, w a radius of 0, there is no mass enclosed
    d = initial_density #initial density in kg / m^3. this is the variable we will pass the newton_intergration fcn.
    p = k_ultra_rel * (d)**(4/3) #initial pressure at the core

        #now we write the integrator using the fact that at the outer boundar, pressure will be 0:
    iter=0
    while p > 0:
        #keep track of num of steps we take
        iter+=1

        #now we update the values:
        r = r + DR #updating our radius value by our radius step size
        d = density(p) #calculating the new pressure value
        dm = mass_step(d,r,DR) #calculating the change in mass
        m = dm + m #updating our mass value
        dp = pressure_step(m,r,d,DR) #calculating the change in pressure
        p = p - dp #updating our pressure value. pressure decreases as we increase radius.
    return m,r


```

```
#the range of density values that we will use in the intergration
d_range = np.linspace(10**12,10**18,10000)


#initialize empty lists where we will add the radius and mass for each density from each iteration
mass_lst=[]
radius_lst=[]

#running through our newton_intergration for each value in our d_range
for d_value in d_range:
    mass,radius =newton_intergration(d_value)
    mass_lst+=[mass]
    radius_lst +=[radius]
    print(d_value,mass,radius)







#plotting our results:
r_arr = np.array(radius_lst) / solar_radius
m_arr = np.array(mass_lst) / solar_mass
d_arr = np.log10(np.array(d_range))



plt.subplot(211)
plt.title('mass and radius dependance on central density')
plt.plot(d_arr,m_arr,color='k')
plt.ylabel(r'mass [M$_{\odot}$]')
plt.axhline(1.4416,ls='--',color='k',label=r'Chandrasekhar Limit: 1.44 M$_{\odot}$')
#plt.axhline(1.44,ls='--',color='k',label=r'Chandrasekhar Limit: 1.44 M$_{\odot}$')
plt.ylim(0,1.6)
#left,right = plt.xlim()
#plt.xlim(0,right)
plt.grid()
plt.legend()



plt.subplot(212)
plt.plot(d_arr,r_arr,color='k')
plt.ylabel(r'radius [R$_{\odot}$]')
```

```
#plt.ylabel('log(pressure)')
#plt.yscale('log')
plt.grid()

#left,right = plt.xlim()
#plt.xlim(0,right)

plt.xlabel(r'log( central density [kg / m$^{3}$] )')
plt.tight_layout()
plt.show()
```