PHYS:50733

Spring 2019

Texas Christian University

Modelling White Dwarf Structures

Sachi Weerasooriya

May 08, 2019

Contents

1 Introduction **3**

2 Theory **3**

2.1 Equilibrium Equations 3

2.2 Chandrasekhar Limit 5

2.3 Rescaling Equations 6

3 Method **7**

3.1 Constants 8

3.2 Functions 8

3.2.1 calculate 8

3.2.2 solve_eq_state 9

3.2.3 RK4(s,h) 9

3.2.4 f(s,h) 10

4 Verification of the Program **10**

4.1 Stability and General Trends 11

4.2 Comparison with Observations 12

5 Results **12**

6 Analysis **13**

7 Critique **14**

8 Appendix **14**

1 Introduction

White dwarfs are formed when stars like our sun exhaust their nuclear fuel. Towards the end of nuclear burning stage, they expel their outer layers creating planetary nebulae. The hot core that remains will cool down over the next billion years to become a "white dwarf" [3]. As the nuclear combustion comes to a halt, the white dwarf can no longer create internal pressure. Thus, gravity crushes it down forcing electrons to be together. In ordinary circumstances pauli exclusion principle prevents electrons with same spin occupying the same space. However, in a white dwarf all energy levels are filled up with electrons making it degenerate. This makes it impossible for gravity to collapse a white dwarf.

According to spectroscopic observations, white dwarfs consist of a variety of atmospheric compositions. Most common type is DA, which make up 80 percent of white dwarfs and consist of mostly hydrogen. Next is DB white dwarfs with He atmospheres. Effective temperature of these stars range from 100,000 K to 4000 K [1].

In this project, we calculate the final radius, total mass, and surface density of a white dwarf star by using a python program that solves the equation of state.

2 Theory

2.1 Equilibrium Equations

We assume the star is in equilibrium. So, the number of electrons are in balance with the number of protons. The electrons are assumed to be relativistic since the kinetic energy of protons and electrons are low. Gravitational pull on particles are balanced by the pressure of degenerate gas of electrons at T=0 where T is well below the Fermi temperature. Gravitational force is given by:

$$\frac{dP}{dr} = -\frac{Gm(r)}{r^2}\rho$$

Where ρ (r) is the mass density and m(r) is the mass enclosed in radius r,

$$m(r) = 4\pi \int_0^r \rho(r) r^2 dr$$

Which gives rise to the differential equation,

$$\frac{dm}{dr} = 4\pi r^2 \rho$$

For a white dwarf with radius R we have m(0)=0, ρ (0)=ρ_c, and $\rho(R) = 0$. Together, with the coupled differential equations above, and the expression for relativistic pressure by degenerate electrons we can find the total mass enclosed in radius R. We treat the electrons as relativistic gas of fermions at T=0. Then the particle density is given by,

$$n = \frac{N}{V} = \frac{1}{\pi^2} \int_0 k_F k^2 dk = \frac{k_f^3}{3\pi^2}$$

where k_f is the wave number related to fermi momentum by $k_f = p_k/\hbar$.

The energy density is given by

$$\eta = E/V = \frac{1}{\pi^2} \int_0^{k_F} k^2 dk \sqrt{(\hbar c k)^2 + m_e^2 c^4}$$

Performing integration, we obtain,

$$\eta(x) = \frac{3}{8x^3}(x(1+2x^2)\sqrt(1+x^2) - \ln(x + \sqrt(1+x^2))$$

with $x = \frac{\hbar k_F}{m_e c}$
Writing x in terms of particle density n, we get;

$$\frac{\hbar k_F}{m_e c} = \left(\frac{n\hbar 3\pi^2}{m_e^3 c^3}\right)^{1/3}$$

where we define $n_0 = \frac{(m_e c)^3}{3\pi^2 \hbar^3}$, and m_e is the mass of electrons. Thus,

$$x = \left(\frac{n}{n_0}\right)^{1/3}$$

The mass of protons and neutrons are larger by a factor of 10^3 when compared to electrons. Therefore, we can approximate the total mass of star by mass density of neucleons. Mass density is given by $\rho = M_p n_p$, with M_p being mass of a proton and n_p being the number density of protons. Number density of protons is related to total number density by $n_p = n/Y_e$, where Y_e is number of electrons per nucleon. Mass density is now given by $\rho = M_p n/Y_e$. Plugging this back in the equation for x, we get

$$x = \left(\frac{\rho}{\rho_0}\right)^{1/3}$$

The pressure is obtained using the energy density,

$$P = 1/3 m_e c^2 x^4 \frac{d\eta}{dx}$$

Taking the derivative of P with respect to ρ we obtain,

$$\frac{dP}{d\rho} = Y_e \frac{m_e c^2}{M_p} \gamma(x)$$

where

$$\gamma(x) = \frac{x^2}{3\sqrt{1+x^2}}$$

Thus, we obtain

$$\frac{d\rho}{dr} = -\left(\frac{dP}{d\rho}\right)^{-1} \frac{Gm}{r^2} \rho$$

The above relativistic equation of state gives the correct result in the right ballpark. However, we have constrained the problem by insisting dependency of mass on radius, limiting the accuracy of the calculation. Although, our model predicts masses greater than $1.4 M_\odot$, in reality, i.e. in the case of extreme relativistic degeneracy the total stellar mass depends only on molecular mass and not on radius, and it cannot exceed the Chandrasekhar limiting mass ($1.4 M_\odot$) [2].

2.2 Chandrasekhar Limit

Electron degenerate objects cannot have masses exceeding the Chandrasekhar limit ($1.4 M_\odot$) without collapsing the object. Increased densities and pressure cannot stop the collapse since the relativistic limit has already been reached [4].

2.3 Rescaling Equations

The above differential equations must be re-scaled into dimensionless quantities in order for them to be solved. This is due to the fact that most of the constants and quantities are either being too small or large. Therefore, we scale the equations with $r = R_0\bar{r}$, $M = M_0\bar{m}$, and $\rho = \rho_0\bar{\rho}$.

We know that $\rho_0 = M_p n_0 / Y_e$, but the values of R_0 and M_0 must be found by plugging these back into our differential equations. By doing so we get,

$$M_0 \frac{dm}{dr} = 4\pi R_0^3 \rho_0 r^2 \bar{\rho}$$

Rearranging the equation we obtain,

$$\frac{dm}{dr} = \left[\frac{4\pi R_0^3 \rho_0}{M_0}\right] \bar{r}^2 \bar{\rho}$$

Notice that the fraction inside the brackets is dimensionless, thus we can equate it to one.

$$\frac{4\pi R_0^3 \rho_0}{M_0} = 1$$

and so

$$M_0 = 4\pi R_0^3 \rho_0$$

Next we consider

$$\frac{d\rho}{dr} = -\left(\frac{dP}{d\rho}\right)^{-1} \frac{Gm}{r^2} \rho$$

This can be written as;

$$\frac{d\rho}{dr} = -\left[\frac{M_p G m}{Y_e m_e c^2}\right] \frac{\rho}{r^2 \gamma(x)}$$

Plugging in the scaling variable

$$\frac{\rho_0}{R_0} \frac{d\bar{\rho}}{d\bar{r}} = -\left[\frac{M_p G M_0 \rho_0}{Y_e m_e c^2 R_0^2}\right] \frac{\bar{m}\bar{\rho}}{\bar{r}^2 \gamma(x)}$$

Note that we can plug in M_0,

$$\frac{\rho_0}{R_0}\frac{d\bar{\rho}}{d\bar{r}} = -\left[\frac{M_p G 4\pi R_0^3 \rho_0 \rho_0}{Y_e m_e c^2 R_0^2}\right]\frac{\bar{m}\bar{\rho}}{\bar{r}^2\gamma(x)}$$

Simplifying further we get,

$$\frac{d\bar{\rho}}{d\bar{r}} = -\left[\frac{M_p G 4\pi R_0^2 \rho_0}{Y_e m_e c^2}\right]\frac{\bar{m}\bar{\rho}}{\bar{r}^2\gamma(x)}$$

Notice similar to earlier calculation what is inside the brackets is dimensionless. By equating that to one we can find the value of R_0.

$$\frac{M_p G 4\pi R_0^2 \rho_0}{Y_e m_e c^2} = 1$$

$$R_0 = \sqrt{\frac{Y_e m_e c^2}{M_p G 4\pi \rho_0}}$$

3 Method

The following program solves the equation of state of a white dwarf star, and returns its final radius R, total mass M, and the surface density ρ. The implementation is as follows.

First, the main function **calculate**, takes in the radial step, critical density, and Y_e (ratio of electrons to neucleons) and calls the sub function **solve_eq_state**. This function takes in the input variables and solves the equation of state using the RK4 method. RK4 method was used over other techniques such as the euler method since it gives a balance between accuracy and the cost of computation.

Second, RK4 method is implemented by calling a sub function function **RK4(s,h)** which solves the differential equations pertaining to equation of state by taking in the initial conditions at zero r=0 and r=R by calling another sub function **f(s,h)** that returns the derivatives at each step in radius.

Constants, implementations, inputs, and outputs of functions are given below. Note that all the calculations are carried out in cgs units.

3.1 Constants

Speed of light, $c = 2.99792458e \times 10^{10}$ cm s^{-1}

Gravitational constant, $G = 6.6743e - 8\ Dyn \cdot cm^{-2}/g^2$

Mass of electron $m_e = 9.1093897 \times 10^{-28} g$

Mass of proton, M_p =$1.6726219 \times 10^{-24} g$

$\hbar = 1.0546e \times 10^{-27}$ erg s

number density of electrons, n_0 =$5.89 \times 10^{29}\ cm^{-3}$

Central density, $\rho_0 = \frac{Mpn_0}{Y_e}\ g \cdot cm^{-3}$

Initial scalar mass, $m_0 = 0.0$ g

3.2 Functions

3.2.1 calculate

Input:

h - Step size
ρ_c - Critical density
Y_e - Ratio of electrons to nucleons

Process:

This function calls the sub function solve_eq_state to calculate radius, mass, and densities at different critical densities and save the calculated final radii, total masses, and surface densities as arrays.

Output

Arrays of radius, mass and density

3.2.2 solve_eq_state

Input :

Y_e - Ratio of electrons to nucleons
h - Step size
ρ_c - Critical density

Process:

This function solves the equation of state by calling the sub function RK4(s,h), which returns radius, mass, and density at each step of radius as arrays. Then, these arrays are integrated along the radius to obtain the final scalar mass, and scalar density.

Next, the scalar values are converted to scaled values by multiplying scalar radius, scalar mass, and scalar density by R_0, M_0, and ρ_0 respectively.

Output:

Returns the final radius, mass, and density of the white dwarf.

3.2.3 RK4(s,h)

Input:

s - Array with initial conditions for radius, mass, and density at r=0
h - Step size

Process:

This is a sub function under the main function solve_eq_state which solves the differential equations using the Runge Kutta 4 method by solving two coupled differential equations. RK4 is a method of solving ordinary differential equations by using a trial step at the midpoint of an interval to cancel out lower order error terms. Implementation of RK4 is as follows,

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf(x_n + 0.5k_1, y_n + 0.5h)$$

$$k_3 = hf(x_n + 0.5k_2, y_n + 0.5 * h)$$
$$k_4 = hf(x_n + k3, y_n + h)$$
$$x_n = x_n + ((k_1 + 2k_2 + 2k_3 + k_4)/6)$$

where x_n, y_n are the derivatives calculated at each point. The derivatives at each step are calculated by calling the sub function **f(s,h)**.

Output:

Returns radius, mass, and density values at each step from r=0 to r=R as three arrays.

3.2.4 f(s,h)

Input:

s - Array with initial conditions for radius, mass, and density at r=0
h - Step size

Process:

This is a sub function under RK4 function which calculates the derivatives of mass and densities at each step in radius.

Output:

Returns the derivatives of mass and densities at each step in radius.

4 Verification of the Program

The program was tested by considering special cases. We verify the program using two ways. First, by testing the stability of solutions and analyzing the general trends in radius and density with mass. Second, by comparing numerical calculations to observed data.

4.1 Stability and General Trends

First, we let $Y_e = 1$, for different ρ_c values (0.1, 1, 10, 100, 10^3, 10^4, 10^5, 10^6) $g \cdot cm^{-3}$ and analyze the stability of solutions by varying the radial step h. We use h values 0.1,0.01,0.001,0.001. We find that solutions stabilize as radial step value decreases. However, computation time also increases with the decrease in h value. See Figure 1 and Figure 2.

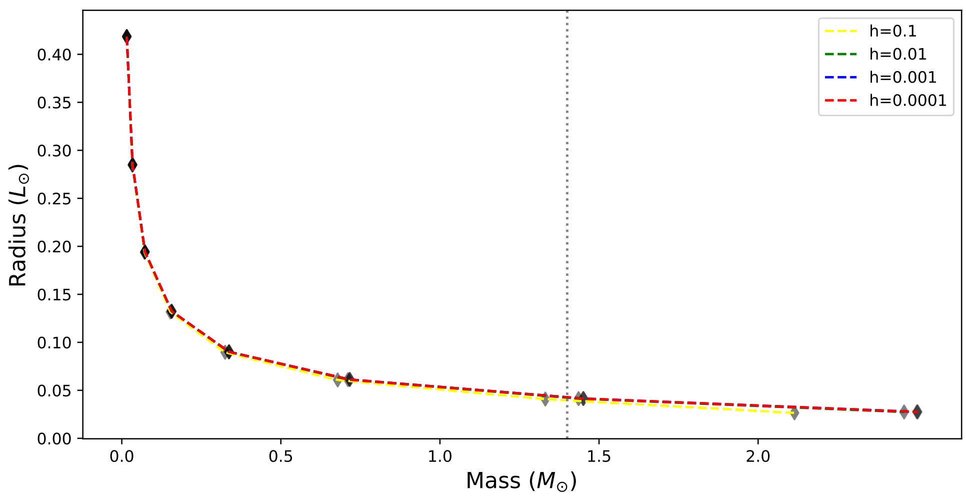


Figure 1: Dependence of final radius on total mass of white dwarfs. Trends are plotted for four cases, h=0.1, h=0.01, h=0.001, h=0.0001. Dotted line denotes the Chandrasekhar mass limit

Next, we analyze the trends in radius and density. According to previous studies on white dwarfs, final radius of a white dwarf decreases as mass increases. However, overall density of white dwarf increases as mass increases. Calculated values stabilizes with decrease in h values. However, this takes a long time to process.Thus, one can choose h=0.001 to balance between speed and accuracy. We find that the resulting trends in radius and density agree with accepted trends for dwarfs and previous studies. See Figure 1 and 2.

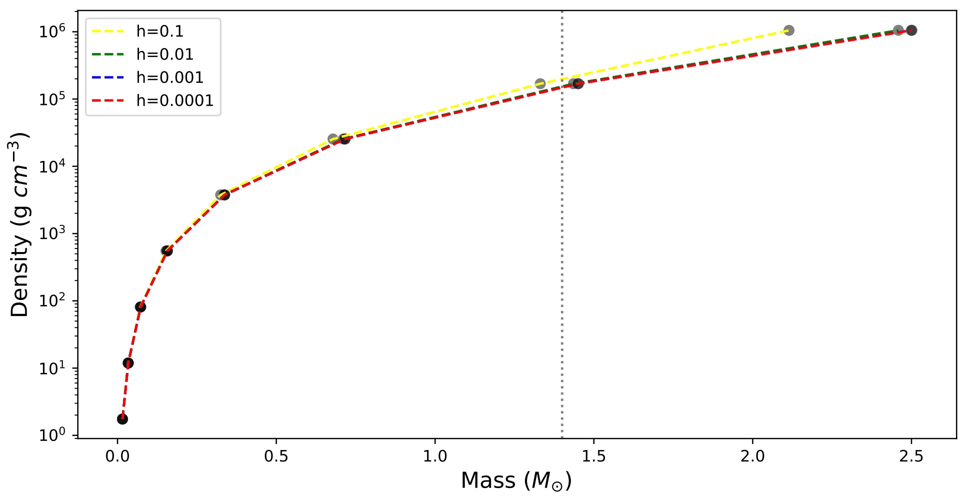


Figure 2: Dependence of surface density on total mass of white dwarfs. Trends are plotted for four cases, h=0.1, h=0.01, h=0.001, h=0.0001. Dotted line denotes the Chandrasekhar mass limit

4.2 Comparison with Observations

In order to verify the output given by the program further, we generate a set of data for ^{56}Fe ($Y_e = 26/56$), ^{12}C ($Y_e = 6/12$), and ^{4}He ($Y_e = 2/4$) with h=0.0001, then plot radius versus mass along with the observed data for stars Sirius B, 40 Eri B, and Stein 2051. Computed trends in radius agree with the observed data for two stars. We have successfully verified the stability and trends of solutions computed by the program. Therefore, we conclude that this program is correct.

5 Results

We computed the final radius, total mass, and surface densities of white dwarf stars at critical densities 0.1 $g \cdot cm^{-3}$ to 10^6 $g \cdot cm^{-3}$ for ^{56}Fe, ^{12}C, and ^{4}He atmospheres using our program.

The final radii of white dwarfs were plotted against their total masses. According to resulting plots, final radius of white dwarf stars decreases with increasing mass while their surface densities increase with increasing mass.

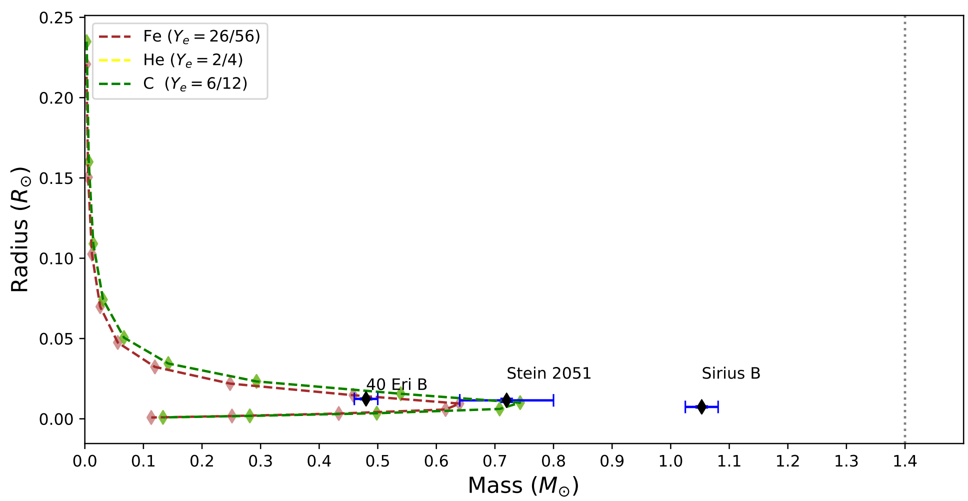


Figure 3: Radius vs Mass relation for Fe, He, and C atmospheres with h=0.0001. Observed data are marked with black diamond shapes

See Figure 3. Computed radii agrees with observed data of 40 Eri B and Stein 2051.

6 Analysis

Final radius of a white dwarfs decreases with increasing total mass while surface density increase with increase in total mass. These trends agree with trends of radius and densities of white dwarfs published in literature. Computed mass radius relation agrees with the observed data of 40 Eri B, and Stein 2051. 40 Eri B lie on data calculated for Iron ($Y_e = 26/56$), while Stein 2051 lie on $Y_e = 0.5$.

Thus, Eri 40 B is likely to be composed of iron while Stein 2051 is likely to be composed of helium or carbon. However, Sirius B do not lie on the computed data. This maybe, due to a different composition of Sirius B rather than being composed of just iron, carbon, or helium.

7 Critique

We have successfully written a program to calculate the final radius, total mass, and surface density of a white dwarf star. The program uses concepts on physical state of white dwarf stars and solves the equation of state with given initial conditions. Although, this simple model is limited by its inability to identify the Chandrasekhar mass limit, computations agree with observed data and accepted trends in radius and mass. This study gives us a better understanding of physics of white dwarfs, equation of state, composition, and their physical limitations such as the Chandrasekhar mass limit.

Computationally, this project allowed us to apply differential equation solving techniques such as Runge-Kutta in solving equation of state and analyze the stability of solutions on step size. Moreover, comparison of calculations with observed data allowed us to verify the results and better understand the composition of white dwarf stars.

This assignment was well rounded in covering the concepts, and testing knowledge of physics and computational techniques of the student with well guided instructions.

However, the central density $\rho_0 = 9.79 \times 10^5\ Y_e^-1g$ for 56^Fe given in the assignment was found to be incorrect since the calculations with that value do not agree with the observed data. Instead central density was calculated using $\rho_0 = \frac{M_p n_0}{Y_e}$. Thus, initial density value should be modified.

8 Appendix

The above program was written using Python 3. A listing of the program is given below.

1 Modelling White Dwarf Structures

1.0.1 by Sachi Weerasooriya

1.0.2 05/03/2019

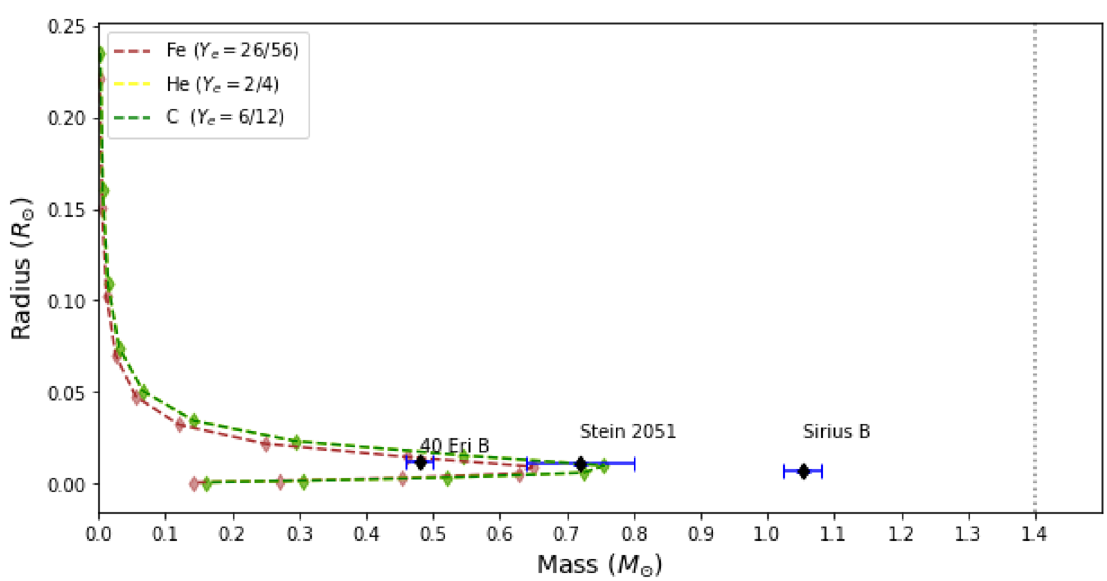
This program calculates the final radius, total mass, and surface density of a white dwarf star. It uses concepts on physical state of white dwarf stars and solves the equation of state with given initial conditions. This simple model is limited by its inability to identify the Chandrasekhar mass limit. See the project report for more details.

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        import scipy.integrate as integrate
```

```
In [2]: def f(s,r):
            '''Calculates the derivative of s         '''
            '''Input initial conditions of s array    '''
            '''Output derivative of s                 '''

            rho     =s[0]
            m       =s[1]

            x       =(rho)**(1/3)
            gamma   =(x*x)/(3*np.sqrt(1+x*x))

            #derivative of density wrt r at r=0 is 0
            if(r==0):
                drho_dr=0.0

            #Calculates the the derivative of density wrt r
            else:
                drho_dr =-(m*rho)/(r*r*gamma)

            #Calculates the derivative of m wrt r
            dm_dr   =r*r*rho

            return np.array([drho_dr,dm_dr],float)
```

```
In [3]: def RK4(s,h):
            '''Solves the differential equations using RK4  '''
            '''Input  : array s, h-step size                '''
            '''Output : radius, density and mass of star    '''

            density   =[]
            mass      =[]
            radius    =[]

            r=0
```

```python
#and compare with observational data
fig2=plt.figure(figsize=(10,5))
ax2=fig2.add_subplot(111)

#Plots computed data for Fe WD
ax2.plot(Fem,Fer,c='brown',ls='--')
ax2.scatter(Fem,Fer,marker='d',c='brown',alpha=0.5)

#Plots computed data for C WD
ax2.plot(Hem,Her,c='yellow',ls='--')
ax2.scatter(Hem,Her,marker='d',c='yellow',alpha=0.5)

#Plots computed data for He WD
ax2.plot(Cm,Cr,c='g',ls='--')
ax2.scatter(Cm,Cr,marker='d',c='g',alpha=0.5)

ax2.set_xlabel('Mass ($M_{\odot}$)',size=14)
ax2.set_ylabel('Radius ($R_{\odot}$)',size=14)
ax2.set_xticks(np.arange(0,1.2,0.1))
ax2.axvline(x=1.4,linestyle=':',c='grey')
ax2.scatter(obs_m,obs_r,marker='d',c='k')
ax2.errorbar(obs_m,obs_r,yerr=[yerr,yerr],xerr=[xerr,xerr],
             ecolor='b',fmt='d',color='k',capsize=4)

ax2.legend(['Fe ($Y_e=26/56$)','He ($Y_e=2/4$)','C  ($Y_e=6/12$)'])
ax2.text(1.053,0.025,'Sirius B')
ax2.text(0.48,0.018,'40 Eri B')
ax2.text(0.72,0.025,'Stein 2051')
ax2.set_xticks(np.arange(0,1.5,0.1))
ax2.set_xlim(0,1.5)
fig2.savefig('compare.pdf')
```



References

[1] David Darling. White dwarf.

[2] C.J. Hansen, S.D. Kawaler, and V. Trimble. *Stellar Interiors: Physical Principles, Structure, and Evolution.* Astronomy and Astrophysics Library. Springer New York, 2012.

[3] NASA. White dwarf stars, 2010.

[4] COSMOS The SAO Encyclopedia of Astronomy. Chandrasekhar limit.