CO31: Structure of White Dwarf Stars

Calum Holker
Somerville College

1 Abstract

We formulate a model for the structure of a white dwarf star under various initial assumptions, both with and without relativistic effects. The model is solved using a computational Runge-Kutta method. Both models agree that as you increase the distance from the centre of the star, the mass increases and density decreases, as expected. When plotting over a range of central densities the results conclude that as mass increases, density increases. In the relativistic model there is a maximum mass, known as the Chandrasekhar mass. A value of $M_c = 1.42M_\odot$ is calculated, which is in agreement with accepted values, within a margin of error induced by the assumptions taken in the model.

2 Introduction

For $\approx 95\%$ of all stars, their life will end as a white dwarf star [2]. As a result of this, the study of the structure of white dwarf stars has huge significance in many aspects of astronomy. White dwarf stars are massive, dense objects that are comprised of heavy nuclei and their electrons. The star structure is defined by the balance of the gravitational force acting on each element, balancing with the electron degeneracy pressure. We assume that the star is solely composed of it's most stable and therefore dominant nuclei - ^{56}Fe. In this derivation we will make the assumption that we are dealing with densities large enough such that the electrons are free from their nuclei, and as such can be modeled as a free Fermi gas. The following equation describes the pressure for the distribution $n(p)$ of electron momenta and is derived in [4]. P_F is defined as the Fermi momentum and in this case is given by $P_F = \left(\frac{3\rho h^3}{8\pi}\right)^{\frac{1}{3}}$:

$$P = \frac{8\pi}{3h^3}\int_0^{P_F} p^3 v_p \; dp \tag{1}$$

The non-relativistic case $v_p = \frac{p}{m_e}$ results in the following equation of state:

$$\frac{dP}{d\rho} = \frac{1}{48}\frac{h^2}{m_e}\frac{2^{\frac{1}{3}}}{m_p^{\frac{5}{3}}}\left(\frac{3\rho}{\pi}\right)^{\frac{2}{3}} \tag{2}$$

The relativistic case $v_p = \frac{pc^2}{\sqrt{p^2c^2 + /m_e^2c^2}}$ results in the following equation of state:

$$\frac{dP}{d\rho} = \frac{8\pi c P_F^4}{9h^2\sqrt{P_F^2 + m_e^2c^2}}\left(\frac{3}{16\pi m_p \rho^2}\right)^{\frac{1}{3}} \tag{3}$$

Equations 2 and 3 can be substituted into the following equations of equilibrium (derived in [1]), defining how density and mass vary with the distance from the centre in a white dwarf star:

$$\frac{d\rho}{dr} = -\frac{d\rho}{dP}\frac{G\rho(r)m(r)}{r^2} \tag{4}$$

$$\frac{dm}{dr} = 4\pi r^2 \rho(r) \tag{5}$$

Equations 4 and 5 are the coupled differential equations we solve to retrieve results for the structure of white dwarf stars. The Runge-Kutta method to solve these is detailed in the following section.

3 Method [1] [6]

The equations given in equations 4 and 5 can be expressed as:

$$\frac{dy_i}{dr} = f_i\left(y_1, y_2\right) \qquad i = 1, 2 \tag{6}$$

From some values y_0 at some radius r we can obtain solutions y_4 at radius $r + \delta r$ by calculating the components of f_i at each step in the method:

$$\begin{aligned} f_0 &= f\left(y_0\right) \\ f_1 &= f\left(y_0 + f_0\frac{\delta r}{2}\right) \\ f_2 &= f\left(y_0 + f_1\frac{\delta r}{2}\right) \\ f_3 &= f\left(y_0 + f_2\delta r\right) \\ y_4 &= y_0 + \left(f_0 + 2f_1 + 2f_2 + f_3\right)\frac{\delta r}{6} \end{aligned} \tag{7}$$

We can then iterate this process for a range of r values, in such a way that y_4 become y_0 in the next step. Using this method for the equations of equilibrium requires initial values of central density and mass. The central mass can be calculated using $m = \frac{4}{3}\pi\delta r^3\rho_c$, where δr is a small displacement from the centre and ρ_c is the value of the central density. Using the solutions to these equations the structure of white dwarf stars can be investigated, looking into the relation between the outer radius and total mass of a star.

4 Results

For a central density $\rho_c = 10^{13}\ kg/m^3$ we solved the equations of equilibrium (4,5) using the non-relativistic equation (2) of state to obtain the following graph (figure 1). This graph shows how the mass and density is affected as you increase the distance from the centre of the white dwarf star.

We can see that the mass increases as you get further from the centre of the star, tending towards a maximum value of the total mass of the star. Density decreases as you increase distance tending towards zero. We can use the point where $\rho = 0.5\% \times \rho_0$ to define the outer radius of the star (this is the point at which the total mass and outer radius of the star are not affected to within 0.01% of their values when r is increased further). This provides the following results for this star with $\rho_c = 10^{13}\ kg/m^3$

$$Outer\ Radius = 2.29 \times 10^6\ m\ \ (3.s.f) \tag{8}$$

$$Total\ Mass = 9.87 \times 10^{31}\ kg\ \ (3.s.f) \tag{9}$$

For central densities in the range $\rho_c \in \left[10^6, 10^{14}\right]$ we solved the equations of equilibrium (4,5) using the non-relativistic (2) and relativistic (3) equations to obtain the following graphs (figure 2) and (figure 3) respectively. These graphs show how the radius and total mass (calculated as above) vary with the central density of the star.

We can see that in both the relativistic and non-relativistic cases the radius decreases as the central density increases, tending towards zero radius. For the the non-relativistic case the mass of the star increases as the central density increases, and seemingly does not reach a maximum value. However for the relativistic case the mass tends towards a maximum value, which we call the Chandrasekhar mass [3]. The value for this mass is calculated below:

$$Chandrasekhar\ Mass = 1.42\ M_{\odot}\ \ (3.s.f) \tag{10}$$

Using the previous results, the outer radius was plotted against the total mass for both the non-relativistic and relativistic cases to obtain the following graph (figure 4). This graph shows how the radius of a white dwarf star is related to the mass of the star. We can clearly see the Chandrasekhar Limit where the mass reaches a maximum value in the relativistic case.

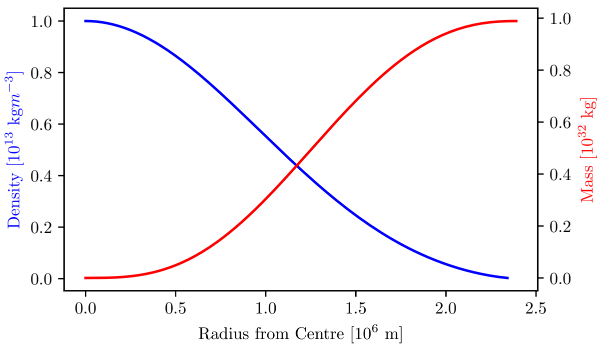

Figure 1: Density vs radius of a non-relativistic Fermi gas white dwarf star of central density $= 10^{13}$

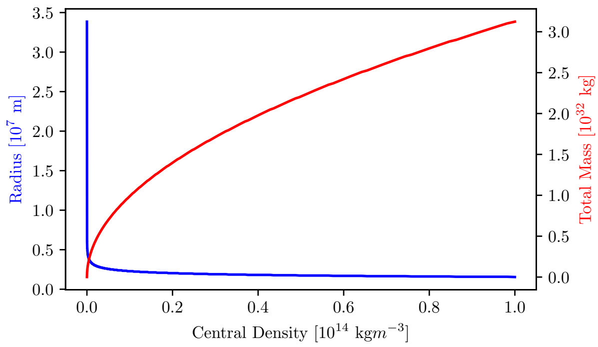

Figure 2: Mass and radius vs central density of a non-relativistic Fermi gas white dwarf star

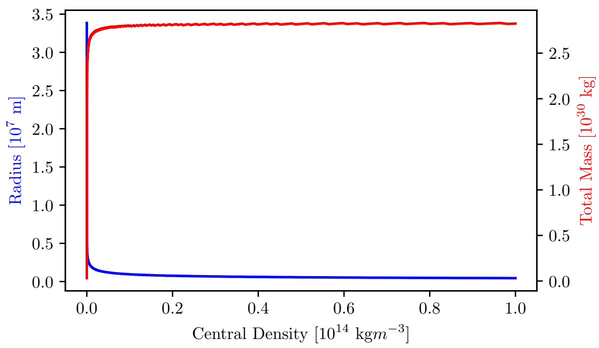

Figure 3: Mass and radius vs central density of a non-relativistic Fermi gas white dwarf star

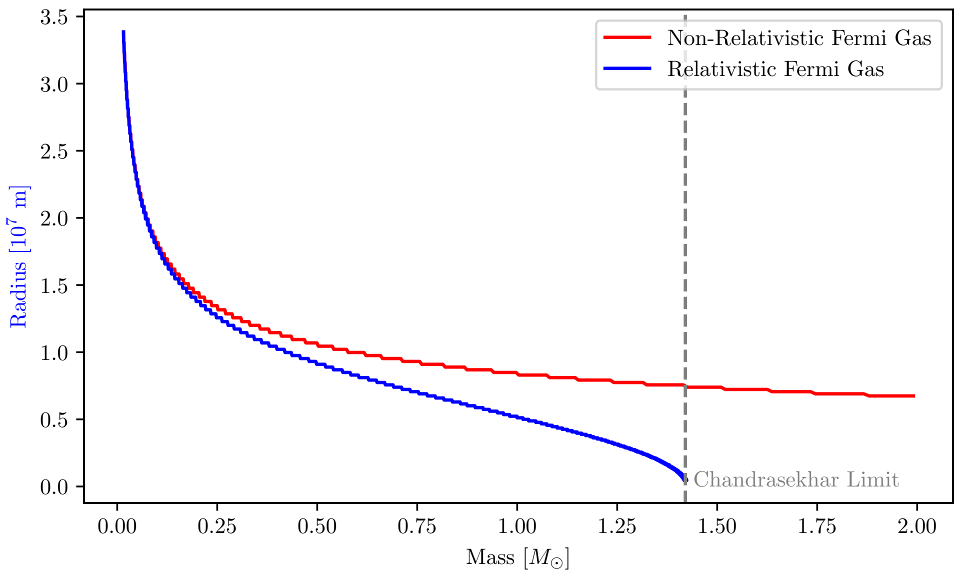

Figure 4: Relation between radius and mass of a white dwarf star

5 Analysis

In the non-relativistic case, we can see that the expected result from regular stellar evolution is obtained that the mass increases and density decreases as the distance from the centre of the star is increased. However there is an unexpected result that the mass increases as the radius of the star decreases, whereas in regular stellar evolution a larger radius is a result of a larger mass. This is due to the larger gravitational force counteracting the similar electron degeneracy pressure. As we can see from the relativistic corrections, this phenomena only occurs up until a limit known as the Chandrasekhar Limit. The calculated value of this limit is in concordance with the accepted value of $\approx 1.4/M_{\odot}$ [3]. At this limit the electron degeneracy pressure can no longer withstand the gravitational forces and the stellar remnant into another form such as neutron star or black hole, rather than a white dwarf star. White dwarf stars are thought to be the final evolutionary stage of a star whose mass is not great enough to exceed this limit.

The 4th order Runge-Kutta method used to solve the equations has an local truncation error of $\approx O(\delta r^5)$ where δr is the relative step size. The global error at the nth radii step is estimated to be n times the local truncation error. This means that the global error is $\approx O(\delta r^4)$. This method essentially a modification of Euler's method to reduce errors. Euler's method is not used in this case as it is very asymmetric over the range of the interval and so produces large truncation errors per step. The 4th order Runge-Kutta method adds three steps, meaning the integration is more symmetric. The obvious disadvantage of this is that it has a very bad time complexity, having to calculate the function 4 times each step. This is the reason we do not use a method with higher order. If we were to have more than two equations, the Runge-Kutta method could still be used, however it's time complexity would increase further, and a lesser order Runge-Kutta method would have to be used. [7] [6]

Our model for white dwarf stars has several assumptions and limitations. For instance:

1. The model does not take into account the rotation of the stars, which could affect the relationship between mass and radius. This remains an ongoing topic of research and it is theorised that this is the cause of variation in luminosity and masses above the limit. [5]
2. The model assumes that white dwarf stars consist of solely iron, however many consist mainly of carbon, which would change our results and value of Chandrasekhar mass significantly for different materials. Accounting for different materials would allow us to use our model to predict the material of a white dwarf star.
3. The model assumes the Fermi gas is at zero temperature, and in order to be more accurate the cooling properties would need to considered. [1]
4. The model does not account for other fields such as electromagnetic forces on the star.
5. The model could account for changes in the molecular structure and therefore density of the different regions in the star, such as the core and surface.
6. In order to fully test the model, measurements of real white dwarf stars would need to be taken and compared with the predictions.

6 Conclusion

A model of the structure of white dwarf stars based on a free Fermi gas at zero temperature is constructed. Using a computational 4th order Runge-Kutta method parameters such as density, radius, and mass are found and plotted. The results clearly show the Chandrasekhar limit, and a value of $M_c = 1.42M_{\odot}$ is found, which is concordant with the accepted value. The model can be improved by more accurately representing the molecular structure and state of a white dwarf, and constructing a more accurate density function. The model also should be tested using values of measured stars.

References

[1] EG, MK, NC, *CO31 Structure of White Dwarf Stars*, Oxford Physics, 2020[1].

[2] Fontaine, G. et al. *An overview of white dwarf stars*, EPJ Web of Conferences, 43, p.05001, 2013[2].

[3] Chandrasekhar S. *An Introduction to the Study of Stellar Structure*, (Dover, New York), 1957[3].

[4] P. Podsiadlowski, *B3 Astrophysics Lecture Series*, Oxford Physics, 2006[4].

[5] Hachisu, I., Kato, M. and Nomoto, K. *Final Fates of Rotating White Dwarfs and their Companions in the Single Degenerate Model of Type 1a Supernovae*, ApJ, 756(1), p.L4., 2012[5].

[6] Fitzpatrick, R. *Runge-Kutta methods*, 2006[6].

[7] *Runge-Kutta Errors*[7].

[1] https://www-teaching.physics.ox.ac.uk/practical_course/scripts/srv/local/rscripts/trunk/Computing/CO31/CO31.pdf

[2] https://www.epj-conferences.org/articles/epjconf/pdf/2013/04/epjconf_liac40_05001.pdf

[3] https://www.ias.ac.in/article/fulltext/pram/077/01/0097-0105

[4] http://www-astro.physics.ox.ac.uk/%7Epodsi/b3$_s$*tellar.pdf*

[5] https://arxiv.org/pdf/1207.2261.pdf

[6] http://farside.ph.utexas.edu/teaching/329/lectures/node35.html

[7] https://www.phys.uconn.edu/ rozman/Courses/m3510$_1 7f/downloads/runge - kutta.pdf$

Appendix A Source Code for Runge-Kutta Solution

The source code for implementing the Runge-Kutta algorithm for a white dwarf star is:

```python
import scipy.constants as sc
import numpy as np

def ode_solve_rk(f, y0, r):
    """
    Solves dy/dt = f(y,t) using a Runge-Kutta algorithm
    Input:  f - a function that receives the current state, y, and the current position and returns the derivative
                value of the statement, dy/dr
            y0 - the initial state of the system, given in a column matrix (Mx1)
            r - vector of position steps with length N where the values of y will be returned
    Output: y - (MxN) matrix that contains the values of y at every position/time step. Columns correspond to the
                position/time and rows to the element of y
    """
    y0 = np.array(y0)
    y = np.zeros((len(y0), len(r))) # Initiates output y matrix
    for i, ri in enumerate(r): # Cycling through r, applying the Runge-Kutta algorithm at each step
        dr = ri if i==0 else (r[i]-r[i-1]) # Finds gap between r values
        f0 = f(y0, ri)
        f1 = f((y0 + f0*(dr/2)), ri)
        f2 = f((y0 + f1*(dr/2)), ri)
        f3 = f((y0 + f2*dr), ri)
        y4 = y0 + ((f0 + 2*f1 + 2*f2 + f3) * (dr/6))
        y[:,i] = y4
        y0 = y4
    return y

def get_density(rho0, r, rel=False):
    """
    Obtains the density as a function of the radial distance using the ODE solver and the non-relativistic
     equation
    Input:  rho0 - the central density at r = 0
            r - the grid points of radial distance where the density is calculated in the form of a vector with
               N elements
            rel - boolean distinguishing relativistic and non-relativistic cases
    Output: rho - an N-element vector that contains the density at the radial grid points given in r
            mass - an N-element vector of the cumulative mass of the white dwarf star from r=0 to the
                 radial grid point
            given in r
    """
    dr = r[0]
    m0 = (4/3)*sc.pi*(dr**3)*rho0 # Initial value of m
    y0 = [rho0, m0]
    if rel == False:
        y = ode_solve_rk(DyDr, y0, r) # Use non-relativistic equation of state
    else:
        y = ode_solve_rk(DyDr_rel, y0, r) # Use relativistic equation of state
    rho = y[0,:]
    mass = y[1,:]
    return(rho, mass)

def DyDr(y, r):
    """
    Calculates the mass and density derivatives at values y, r using the non-relativistic equations of state
    Input:  y - 2x1 dimensional array containing values of the density and mass
            r - the distance from the centre of the star
    Output: dydr - 2x1 dimensional array containing values of the density and mass derivatives
    """
    dydr = []
    dPdp = (((sc.h**2)*(2**(1/3)))/(48*sc.electron_mass*(sc.proton_mass**(5/3))))*(((3*y[0])/(sc.pi))**(2/3))
    dpdr = (-1 * (1/dPdp) * ((sc.G*y[0]*y[1])/(r**2)))
    dydr.append(dpdr)
    dmdr = (4 * sc.pi * (r**2) * y[0])
    dydr.append(dmdr)
    dydr = np.array(dydr)
    return dydr

def DyDr_rel(y, r):
    """
    Calculates the mass and density derivatives at values y, r using the relativistic equations of state
    Input:  y - 2x1 dimensional array containing values of the density and mass
            r - the distance from the centre of the star
    Output: dydr - 2x1 dimensional array containing values of the density and mass derivatives
    """
    dydr = []
    A = (8 * sc.pi * (sc.electron_mass**4) * (sc.c**5))/(3 * (sc.h**3))
    Pf = ((3 * (sc.h**3) * y[0])/(16 * sc.pi * sc.proton_mass))**(1/3)
    a = np.arcsinh(Pf/(sc.electron_mass*sc.c))
    dPda = A * (np.sinh(a)**4)
    dadPf = ((Pf**2)+((sc.electron_mass*sc.c)**2))**(-1/2)
    dPfdp = (((3 * (sc.h**3))/(16 * sc.pi * sc.proton_mass))**(1/3)) * (1/3) * (y[0]**(-2/3))
    dPdp = dPda * dadPf * dPfdp
    dpdr = (-1/dPdp) * ((sc.G * y[0] * y[1])/(r**2))
    dydr.append(dpdr)
    dmdr = 4 * sc.pi * (r**2) * y[0]
    dydr.append(dmdr)
    dydr = np.array(dydr)
    return dydr
```

Appendix B Source Code for Solving for Desired Results

The source code for producing the desired results, using the functions defined in appendix A is:

```
# Calculating mass and density values for a central density of 1e13 (non-relativistic)

rho0 = 1e13
r = np.logspace(-1, 3.5 , 1000, base=100) # Using a logarithmic separation to get an even spread of values
rho, mass = get_density(rho0, r)

# Calculating the outer radius and total mass of the star with central density 1e13 (non-relativistic)

# Finds the first value of the radius when the density is less than 0.5% of the central density
Outer_radius = r[np.argmax(rho<(0.005*rho0))-1]
# Finds the first value of the mass when the density is less than 0.5% of the central density
Total_mass = mass[np.argmax(rho<(0.005*rho0))-1]
print(Outer_radius, Total_mass)

# Calculating outer radii and total mass of stars over a range of central densities (non-relativistic)

rho0 = np.logspace(6, 14, 1000) # Central density ranging from 1e6 to 1e14
r = np.logspace(-1, 4 , 1000, base=100)
rad = []
total_mass = []
for r0 in rho0:
    rho, mass = get_density(r0, r, rel=False) # Calculates values of mass and density for each central density
    rad.append(r[np.argmax(rho<(0.005*r0))-1])
    total_mass.append(mass[np.argmax(rho<(0.005*r0))-1])

# Calculating outer radii and total mass of stars over a range of central densities (relativistic)

rho0 = np.logspace(6, 14, 1000) # Central density values ranging from 1e6 to 1e14
r = np.logspace(-1, 4, 1000, base=100)
rad_rel = []
total_mass_rel = []
for r0 in rho0:
    rho, mass = get_density(r0, r, rel=True) # Calculates values of mass and density for each central density
    rad_rel.append(r[np.argmax(rho<(0.005*r0))-1])
    total_mass_rel.append(mass[np.argmax(rho<(0.005*r0))-1])
```