Structure of White Dwarf Stars

Michael Papasimeon
August 27, 1997

***Abstract*—This paper was written for an undergradue class in computational physics in 1997. It focuses on computational/numerical techniques such as Euler's and the Runge-Kutta method of numerical integration for modelling the mass and density profiles of white dwarf stars. The stars *Sirius B* and *40 Eri B* are used as case studies. The code to compute the density profiles was written in FORTRAN77 and is listed in the paper's appendix.**

I. Aims

The purpose is to write a computer program to numerically solve the coupled differential equations 1 and 2, describing the mass profile $\bar{m}(\bar{r})$ and density profile $\bar{\rho}(\bar{r})$ of a white dwarf star.

$$\frac{d\bar{\rho}}{d\bar{r}} = -\frac{\bar{m}\bar{\rho}}{\gamma(\bar{\rho}^{1/3})\bar{r}^2} \tag{1}$$

$$\frac{d\bar{m}}{d\bar{r}} = \bar{r}^2\bar{\rho} \tag{2}$$

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

The main aims are:

- Solve the differential equations first using Euler's method, and then the using the fourth order Runge-Kutta method.
- Compare the results of the Euler method with the results of the Runge-Kutta method.
- Obtain solutions for $\bar{m}(\bar{r})$ and $\bar{\rho}(\bar{r})$ (and hence $\bar{M}$ and $\bar{R}$) for different values of the central density ρ_c.
- Investigate the mass and density profiles for large values of the central density.
- Solve the differential equations for the white dwarf stars Sirius B and 40 Eri B.

II. Background on White Dwarf Stars

There are two main forces at work in white dwarf stars. The first is the force of gravity caused by the mass of the star. Most of the star's mass is due to the mass of all the nuclei of the atoms which make up the star. Although stars are made primarily of lighter elements such as Hydrogen and Helium, heavier nuclei such as Carbon (^{12}C) and Iron (^{56}Fe) also make a large contribution to the mass of the star.

The other important force present in a white dwarf is that which is generated by the electron degeneracy pressure. Due to high temperatures in a star, the electrons are very energetic, are not bound to nuclei and move around in a Fermi gas. The electron degeneracy pressure is due to the Pauli exclusion principle.

Therefore, it is when the gravitational force and the force caused by the electron degeneracy pressure are in equilibrium, that the white dwarf is stable and does not collapse in on itself.

If P is the electron degeneracy pressure we can equate the corresponding force dP/dr with Newton's law of gravitation.

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

where G is the gravitational constant, r is the radius, $m(r)$ is the mass profile, and $\rho(r)$ is the density profile.

From this we can differential equations relating the mass and density with the radius.

$$\frac{d\rho}{dr} = -\left(\frac{dP}{d\rho}\right)^{-1}\frac{Gm(r)\rho(r)}{r^2} \tag{5}$$

$$\frac{dm}{dr} = 4\pi r^2\rho(r) \tag{6}$$

To find the equation of state for P, the electrons in the star can be treated as a Fermi gas of N electrons. Using statistical mechanics, we calculate the average occupation number of the electron gas and hence the total energy (using relativistic calculations because the electrons are very energetic). The energy is calculated because it allows us to calculate the pressure, because $P = -dE/dV$. From this we find that

$$\frac{dP}{d\rho} = Y_e\frac{m_e}{m_p}\gamma(x) \tag{7}$$

where Y_e is the number of electrons per nucleon, m_e is the electron mass, m_p is the nucleon (proton) mass and $\gamma(x) = x^2/(3\sqrt{1+x^2})$. Equations 5 and 6 then become the coupled differential equations describing the mass and density profiles of a white dwarf star which are now given by:

$$\frac{d\rho}{dr} = -\frac{m_pG}{Y_em_e}\frac{m(r)\rho(r)}{r^2\gamma([\frac{\rho}{\rho_0}]^{1/3})} \tag{8}$$

$$\frac{dm}{dr} = 4\pi r^2\rho(r) \tag{9}$$

These equations are scaled to make it easier to solve them by computer. The corresponding scaled equations which are solved are equations 1 and 2.

III. Method

The method followed was:

- Write a FORTRAN program to solve the coupled differential equations for a white dwarf star using Euler's method.
- Modify the program, by adding a subroutine to solve the differential equations using the fourth order Runge-Kutta

Fig. 1. Flow Chart for Runge-Kutta Method

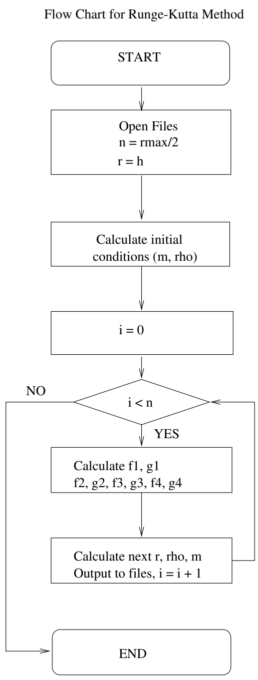


TABLE I
COMPARING EULER AND RUNGE-KUTTA FOR $h = 0.01$.

h	Euler	Runge-Kutta	Expected
M	1.550	1.580	1.58
$\bar{R}$	1.313	1.298	1.298

method. A flow chart showing a rough outline of the algorithm is shown in Figure 1.

- The source code for the program (`star.f`) is in Appendix A.
- The code was written so that the program stops if the density $\bar{\rho}$ becomes negative. This is because an error is generated when the program attempts to compute $\bar{\rho}^{1/3}$ when $\bar{\rho} < 0$.

IV. RESULTS

A. Comparing Euler and Runge-Kutta Methods

Table I shows the values obtained for $\bar{M}$ and $\bar{R}$ using Euler's and Runge-Kutta methods for a step size, $h = 0.01$ and central density $\rho_c = 10.0$.

We can see from Table I that the Runge-Kutta method is more accurate.

1) Mass Profile: Figures 2 and 3 show the mass profile of a white dwarf star calculated using Euler's and the Runge-Kutta methods with:

- Step size $h = 0.01$
- $\rho_c = 10.0$

2) Density Profile: Figures 4 and 5 show the density profile of a white dwarf star calculated using Euler's and the Runge-Kutta methods with:

Fig. 2. Plot of $\bar{m}(\bar{r})$ using Euler's method.

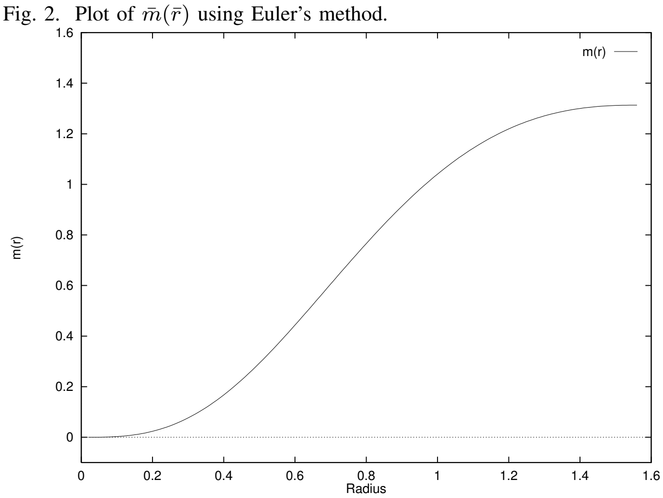
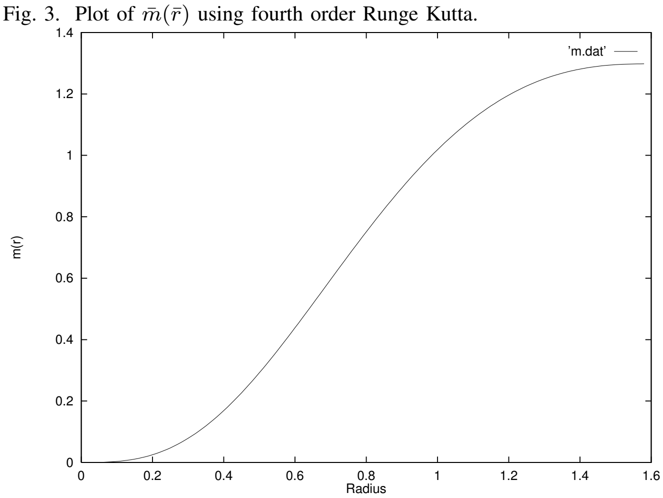


Fig. 3. Plot of $\bar{m}(\bar{r})$ using fourth order Runge Kutta.



- Step size $h = 0.01$
- ρ_c = 10.0

B. Investigating Stability with different step sizes

Table II and III show values of $\bar{R}$ and $\bar{M}$, with the central density $\rho_c = 10.0$, for different step sizes h using both the Euler and the Runge-Kutta methods.

The first observation is that the Runge-Kutta algorithm reaches the answer faster as we increase the step size. This is to be expected as we know it provides more accurate results than the Euler algorithm.

By looking at the column for $\bar{M}$, we see that the Runge-Kutta algorithm is more stable than the Euler algorithm as we decrease the step size.

C. Investigation of mass and density at large central densities

Figure 6 and 7 plot the mass and density profiles for the white dwarf star for a range of values of the central density ρ_c, (with $Y_e = 1$), using the fourth order Runge-Kutta algorithm with a step size of $h = 0.01$. These plots are the solutions for $\bar{\rho}(\bar{r})$ and $\bar{m}(\bar{r})$, of 1 and 2.

Fig. 4. Plot of $\bar{\rho}(\bar{r})$ using Euler's method.

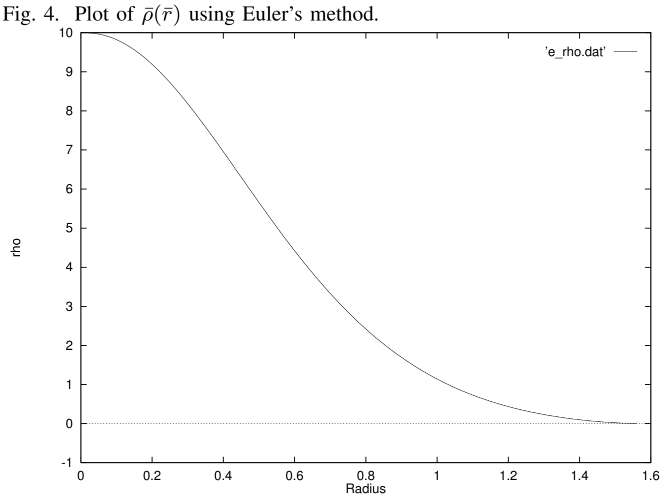


Fig. 5. Plot of $\bar{\rho}(\bar{r})$ using fourth order Runge Kutta.

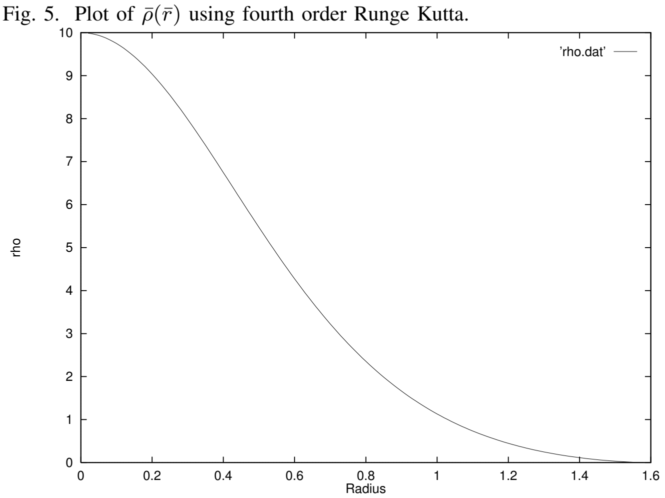


TABLE II
$\bar{R}$ AND $\bar{M}$ FOR DIFFERENT VALUES OF THE STEP SIZE h, USING EULER'S METHOD.

h	$\bar{R}$	$\bar{M}$
0.1	1.3	1.47581763
0.01	1.55	1.31320078
0.001	1.587	1.29950998
0.0001	1.5911	1.29816256
0.00001	1.59157	1.29802791
0.000001	1.591623	1.29801445

TABLE III
$\bar{R}$ AND $\bar{M}$ FOR DIFFERENT VALUES OF THE STEP SIZE h, USING FOURTH ORDER RUNGE-KUTTA.

h	$\bar{R}$	$\bar{M}$
0.1	1.5	1.29518167
0.01	1.58	1.29799616
0.001	1.591	1.29801293
0.0001	1.5916	1.29801295
0.00001	1.59162	1.29801295
0.000001	1.591629	1.29801295

TABLE IV
$\bar{R}$ AND $\bar{M}$ FOR $h = 0.01$ AT DIFFERENT VALUES OF THE CENTRAL DENSITY ρ_c.

ρ_c	$\bar{R}$	$\bar{M}$
0.1	2.57	0.22178884
1	2.49	0.70706357
10	1.58	1.29799616
100	0.95	1.73552752
1000	0.53	1.93285505
10000	0.27	1.99681784
100000	0.13	2.01300131
1000000	0.05	1.76970419

Using the solutions to the differential equations we determine $\bar{R}$ and $\bar{M}$ for the different values of ρ_c. These results are shown in Table IV.

As the central density increases, the mass of the star increases. As the central density increases the radius decreases because there is more mass and hence the star is falling in on itself yet there still exists the equilibrium between the gravitational force and the electron degeneracy pressure. However, a smaller radius increases the gravitational field of the star, since we know from Newton's law of gravitation the gravitational force proportional to $1/r^2$.

$$F_g = -\frac{GmM}{r^2} \tag{10}$$

Therefore, the greater the mass and the smaller the radius of the star the greater the gravitational force of the star, pushing the star in on itself.

For extremely high values of the central density, the resulting radius and mass may by be so high it results in an extremely large gravitational field which is too strong for it to stay in equilibrium and hence the assumptions made about equations 1 and 2 no longer hold and the star will collapse in on itself. Then depending on the mass of the star it may become a neutron star, or if the mass is extremely large, a black hole.

Figure 6 shows the computed mass profiles for different values of ρ_c and Figure 7 shows the computed density profiles for different values of ρ_c.

V. CALCULATIONS FOR SIRIUS B AND 40 ERI B

For values of $Y_e \neq 1$, we have:

$$M(Y_e) = Y_e^2 \frac{M_0}{M_{sun}} \bar{M}(Y_e = 1)$$

$$R(Y_e) = Y_e \frac{R_0}{R_{sun}} \bar{R}(Y_e = 1)$$

Using values from Koonin [1] we can calculate:

$$\frac{M_0}{M_{sun}} = \frac{5.67 \times 10^{33} Y_e^2 gm}{1.98 \times 10^{33} gm} = 2.863636 Y_e^2$$

$$\frac{R_0}{R_{sun}} = \frac{7.72 \times 10^8 Y_e cm}{6.95 \times 10^{10} cm} = 1.11079 \times 10^{-2} Y_e$$

Fig. 6. Mass profiles for different values of ρ_c

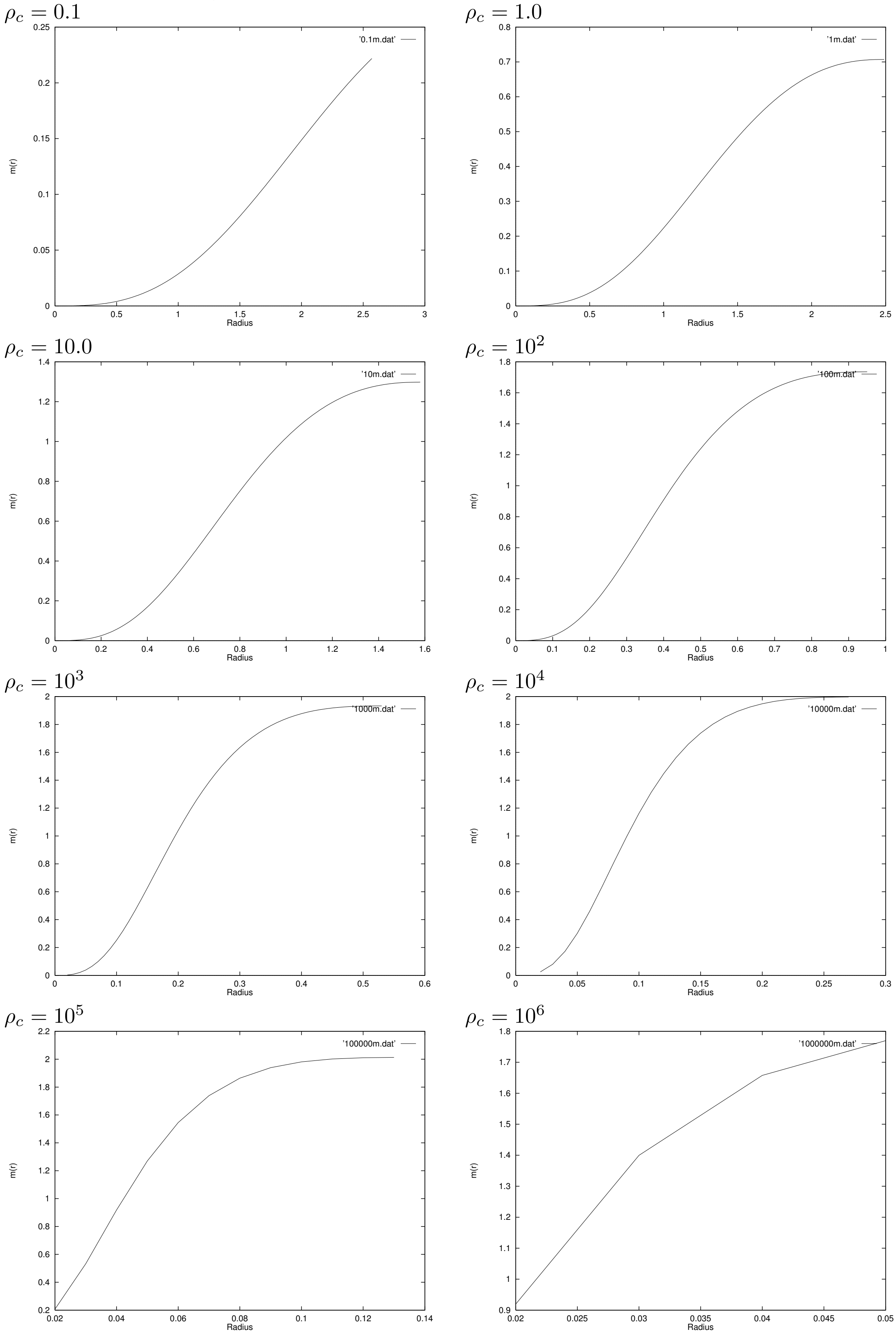


Fig. 7. Density profiles for different values of ρ_c

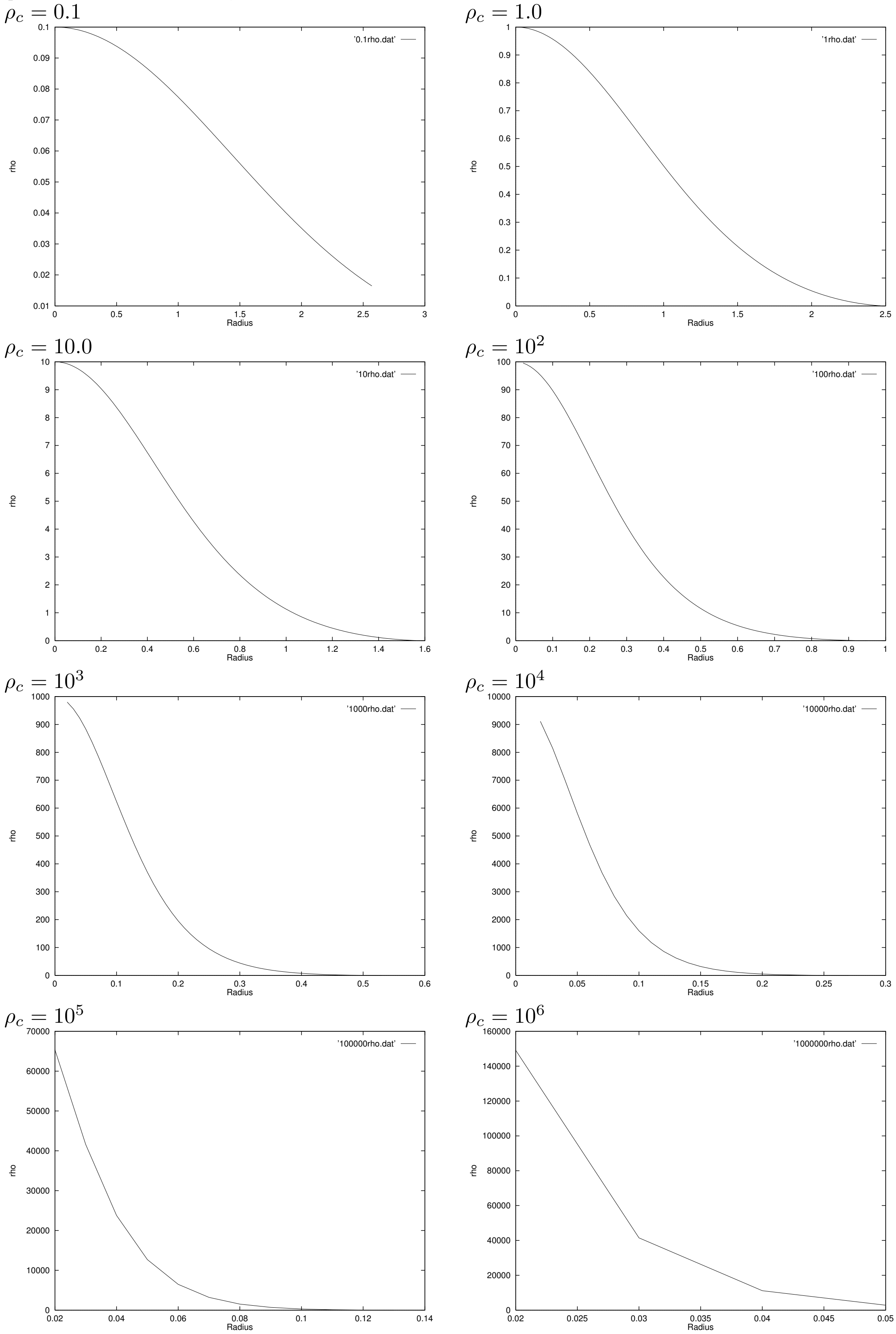


TABLE V
RESULTS FOR SIRIUS B

ρ_c	$\bar{R}(Y_e=1)$	R ($Y_e=0.5$)	R ($Y_e=0.464$)
10.0	1.58	0.00877525071	0.00814343218
15.0	1.45	0.00877525071	0.00814343218
16.0	1.45	0.00805323641	0.00747340295
18.0	1.40	0.00777553860	0.00721569940
20.0	1.37	0.00760891992	0.00706107727
22.0	1.34	0.00744230123	0.00690645514

TABLE VI
RESULTS FOR 40 ERI B

ρ_c	$\bar{R}(Y_e=1)$	R ($Y_e=0.5$)	R ($Y_e=0.464$)
1.0	2.49	0.0138293508	0.0128336368
1.2	2.41	0.0133850343	0.0124213111
1.5	2.31	0.0128296387	0.0119059040
1.7	2.25	0.0124964013	0.0115966597
2.0	2.19	0.0121631640	0.0112874155
3.0	2.02	0.0112189914	0.0104112234
5.0	1.83	0.0101637397	0.0094319499
6.0	1.76	0.0097749628	0.0090711649

A. *Sirius B*

- $M = 1.053 \pm 0.028$
- $R = 0.0074 \pm 0.0006$

Table V shows the results for the radius R, for the program running for a number of different central densities. Here R is:

$$R = Y_e \frac{R_0}{R_{sun}} \bar{R}(Y_e = 1)$$

We see from Table V that the central density of Sirius B is:

- $\rho_c = 22.0$ (Solar Units) for $Y_e = 0.5$ (Carbon)
- $\rho_c = 16.0$ (Solar Units) for $Y_e = 0.464$ (Iron)

We see that these results are within error. Sirius B has a mass very similar to that of the Sun but a much smaller radius. This implies that the density of Sirius B is greater than the Sun, resulting in a smaller radius. What this means is that the atoms which make up Sirius B are likely to be heavier than those in the Sun. We are more likely to find greater amounts of heavy nuclei such as Carbon and Iron. Without knowing the exact proportions of Carbon and Iron in Sirius B, we can only estimate the the central density of this what dwarf is somewhere in the range $16.0 \leq \rho_c \leq 22.0$ in solar units.

B. *40 Eri B*

- $M = 0.48 \pm 0.02$
- $R = 0.0124 \pm 0.0005$

Table VI shows the results for R, with the program running with different values of the central density.

We see from Table VI that the central density of 40 Eri B is:

- $\rho_c = 1.7$ (Solar Units) for $Y_e = 0.5$ (Carbon)
- $\rho_c = 1.2$ (Solar Units) for $Y_e = 0.464$ (Iron)

We see that 40 Eri B has mass that is is approximately half that of the Sun, and a radius only 1% that of the Sun's. However we see that the central density of 40 Eri B is in the range $1.2 \leq \rho_c \leq 1.7$ in solar units. If a star half the like the Sun where to have it's radius reduced to 1% of it's original size, it's density would increase dramatically because:

$$\rho = \frac{M}{V} = \frac{M}{\frac{4}{3}\pi r^3}$$

This suggests that that the atoms in 40 Eri B are lighter than those found in the Sun. It is likely that 40 Eri B consists almost totally of Hydrogen and Helium, with only traces of heavier nuclei.

VI. REFERENCES

1) *Computational Physics*, Koonin and Meredith

APPENDIX
CODE LISTING: STAR.F

```
c-------------------------------------------------------------
c PROGRAM : WHITE DWARF STARS
c-------------------------------------------------------------
      program main
      implicit none
      real*8 h, rhoc, rmax

      rmax = 2.55d0
      write(*,*)'Enter␣rhoc'
      read(*,*) rhoc
c call euler(h, rhoc, rmax)
      call runge_kutta(h, rhoc, rmax)

      end
c-------------------------------------------------------------
c SUBROUTINE : EULER
c-------------------------------------------------------------

      subroutine euler(h, rhoc, rmax)
      implicit none
      real*8 h, rhoc, rmax
      real*8 f, g, rho0, m0
      real*8 r, rho, m
      real*8 ri, rhoi, mi
      integer*4 n, i

      open(unit=1, file='e_rho.dat', status='unknown')
      open(unit=2, file='e_m.dat', status='unknown')

      n = nint(rmax/h)

      r = h
      m = m0(rhoc, h)
      rho = rho0(rhoc, h)

      do 100 i = 0, n, +1
            ri = r
            rhoi = rho
            mi = m

            r = ri + h
            rho = rhoi + h*f(ri,rhoi,mi)
            m = mi + h*g(ri, rhoi, mi)

            write(1,*)r, rho
            write(2,*)r, m
100 continue

      close(unit=1)
      close(unit=2)

      end

c-------------------------------------------------------------
c SUBROUTINE : RUNGE_KUTTA
c-------------------------------------------------------------

      subroutine runge_kutta(h, rhoc, rmax)
      implicit none
      real*8 h, rhoc, rmax
      real*8 f, g, rho0, m0
      real*8 r, rho, m
      real*8 ri, rhoi, mi, hon2
      real*8 f1, f2, f3, f4
```

```
      real*8 g1, g2, g3, g4
      integer*4 n, i

      open(unit=1, file='rho.dat', status='unknown')
      open(unit=2, file='m.dat', status='unknown')

      n = nint(rmax/h)
      hon2 = h/2.0d0

      r = h
      m = m0(rhoc, h)
      rho = rho0(rhoc, h)

      do 100 i = 0, n, +1
           ri = r
           rhoi = rho
           mi = m

           f1 = f(ri, rhoi, mi)
           g1 = g(ri, rhoi, mi)

           f2 = f(ri + hon2, rhoi + hon2*f1, mi + hon2*g1)
           g2 = g(ri + hon2, rhoi + hon2*f1, mi + hon2*g1)

           f3 = f(ri + hon2, rhoi + hon2*f2, mi + hon2*g2)
           g3 = g(ri + hon2, rhoi + hon2*f2, mi + hon2*g2)

           f4 = f(ri + h, rhoi + h*f3, mi + h*g3)
           g4 = g(ri + h, rhoi + h*f3, mi + h*g3)

           r = ri + h
           rho = rhoi + (h/6.0d0)*(f1 + 2*f2 + 2*f3 + f4)
           m = mi + (h/6.0d0)*(g1 + 2*g2 + 2*g3 + g4)

           write(1,*)r, rho
           write(2,*)r, m, (r*0.005553956), (r*0.005154071d0)
100 continue

      close(unit=1)
      close(unit=2)

      end

c--------------------------------------------------------------
c FUNCTION : f
c--------------------------------------------------------------

      double precision function f(r, rho, m)
      implicit none
      real*8 r, rho, m, x
      real*8 gama

        if (rho .lt. 0.0d0) then
               write(*,*)'Rho␣<␣0␣:␣Stopping␣Program'
               stop
        endif

      f = -(m*rho)/(gama(rho**(1.0d0/3.0d0))*r**2.0d0)
      return

      end

c--------------------------------------------------------------
c FUNCTION : g
c--------------------------------------------------------------

      double precision function g(r, rho, m)
      implicit none
```

```
      real*8 r, rho, m

      g = (r**2.0d0)*rho
      return

      end

c-------------------------------------------------------------
c FUNCTION : gama
c-------------------------------------------------------------

      double precision function gama(x)
      implicit none
      real*8 x

      gama = (x**2.0d0)/(3.0d0*dsqrt(1.0d0 + x**2.0d0))
      return

      end

c-------------------------------------------------------------
c FUNCTION : m0
c : initial value of m
c : rhoc = central density
c-------------------------------------------------------------

      double precision function m0(rhoc, h)
      implicit none
      real*8 rhoc, h

      m0 = rhoc*(h**3.0d0)/3.0d0
      return

      end

c-------------------------------------------------------------
c FUNCTION : rho0
c : initial value of rho
c : rhoc = central density
c-------------------------------------------------------------

      double precision function rho0(rhoc, h)
      implicit none
      real*8 rhoc, h, x
      real*8 gama

      x = rhoc**(1.0d0/3.0d0)
      rho0 = rhoc*(1 - ((h**2.0d0)*rhoc)/(6*gama(x)) )
      return

      end
```