

# Computational Physics with Maxima or R

## Project 2

### Structure of White Dwarf Stars \*

Edwin (Ted) Woollett

August 25, 2015

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	The Equations of Equilibrium . . . . .	3
1.2	Background on Quantum Mechanics and Pressure Ionization . . . . .	4
1.3	The Approximate Equation of State . . . . .	5
1.4	Internal Kinetic Energy of the Star . . . . .	8
1.5	Internal Gravitational Energy of the Star . . . . .	8
<b>2</b>	<b>Scaling the Differential Equations</b>	<b>8</b>
<b>3</b>	<b>Scaling the Kinetic and Gravitational Energy</b>	<b>9</b>
<b>4</b>	<b>Scaled Dimensionless Differential Equations Summary</b>	<b>9</b>
<b>5</b>	<b>Expansion of <math>\bar{m}(\bar{r})</math> and <math>\bar{\rho}(\bar{r})</math> about <math>\bar{r} = 0</math></b>	<b>10</b>
<b>6</b>	<b>Behavior of <math>\bar{\rho}(\bar{r})</math> Near the Surface</b>	<b>10</b>
<b>7</b>	<b>Maxima Code dwarf1(rhoc, dr, rtol)</b>	<b>11</b>
7.1	Maxima Plots of $\bar{m}(\bar{r})$ and $\bar{\rho}(\bar{r})$ . . . . .	12
<b>8</b>	<b>R Code dwarf1(rhoc, dr, rtol, nmax)</b>	<b>15</b>
8.1	R Plots of $\bar{m}(\bar{r})$ and $\bar{\rho}(\bar{r})$ . . . . .	17
<b>9</b>	<b>Energy Plots Using Maxima</b>	<b>18</b>
<b>10</b>	<b>Energy Plots Using R</b>	<b>19</b>
<b>11</b>	<b>Plot of <math>\bar{R}</math> Versus <math>\bar{M}</math> Using Maxima</b>	<b>21</b>
<b>12</b>	<b>Plot of <math>\bar{R}</math> Versus <math>\bar{M}</math> Using R</b>	<b>22</b>
<b>13</b>	<b>Energy Components as a Function of <math>\bar{\rho}_c</math> and <math>\bar{M}</math> Using Maxima</b>	<b>24</b>
<b>14</b>	<b>Comparison of Non-Relativistic Approximation with Relativistic Solutions</b>	<b>26</b>
14.1	Plot comparisons . . . . .	26
<b>15</b>	<b>Suggested Further Explorations</b>	<b>27</b>
<b>16</b>	<b>Astrophysics References</b>	<b>27</b>

---

\*The code examples use **R ver. 3.0.2** and **Maxima ver. 5.31.2** using **Windows XP**. This is a live document which will be updated when needed. Check <http://www.csulb.edu/~woollett/> for the latest version of these notes. Send comments and suggestions for improvements to [woollett@charter.net](mailto:woollett@charter.net)

project2.pdf describes the major project proposed in Chapter 2 of Computational Physics with Maxima or R and is made available to encourage the use of the R and Maxima languages for computational physics projects of modest size.

R language free and open-source software:

<http://www.r-project.org/>

Maxima language free and open-source software:

<http://maxima.sourceforge.net/>

Code files available on the author's webpage are

project2.pdf  
project2.mac  
project2.R  
project2.tex  
project2.zip

The author uses the XMaxima interface to Maxima exclusively, with the startup file setting: `display2d:false$`, which allows denser screen output.

The author normally uses the default RGui interface when coding in R.

COPYING AND DISTRIBUTION POLICY:

NON-PROFIT PRINTING AND DISTRIBUTION IS PERMITTED.

You may make copies of this document and distribute them to others as long as you charge no more than the costs of printing.

Keeping a set of notes about using R and Maxima up to date is easier than keeping a published book up to date.

Feedback from readers is the best way for this series of notes to become more helpful to users of **R** and **Maxima**. *All* comments and suggestions for improvements will be appreciated and carefully considered.

# 1 Introduction

The major project proposed in Chapter 2 of Steven Koonin's text **Computational Physics** applies numerical integration of ordinary differential equations to the prediction of the structure of white dwarf stars.

The model chosen, for simplicity, is a system at zero degrees Kelvin temperature (throughout the star, including the surface region) in which the electrons are a degenerate gas (similar to the electrons in a metal) and are responsible for the internal pressure (the heavy nuclei random motion and degeneracy pressure is neglected) of the star, and the heavy nuclei are responsible for the force of gravity holding the star together (neglecting the relatively small mass of the electrons).

## 1.1 The Equations of Equilibrium

Quoting Koonin

If the star is in mechanical (hydrostatic) equilibrium, the gravitational force on each bit of matter is balanced by the force due to the spatial variation of the pressure  $P$ . The [radial component of the] gravitational force acting on a unit volume of matter at a radius  $r$  is

$$F_{grav} = -G m(r) \rho(r)/r^2, \quad (1.1)$$

where  $G$  is the gravitational constant,  $\rho(r)$  is the mass density, and  $m(r)$  is the mass of the star interior to the radius  $r$ :

$$m(r) = \int_0^r \rho(r') 4\pi r'^2 dr' \quad (1.2)$$

The [radial component of the] force per unit volume of matter due to the changing pressure is  $-dP/dr$ . When the star is in equilibrium, the net [radial component of the] force (gravitational plus pressure) on each bit of matter vanishes, so that, using Eq.(1.1), we have

$$\frac{dP}{dr} = -\frac{G m(r)}{r^2} \rho(r) \quad (1.3)$$

A differential relation between the mass and the density can be obtained by differentiating the integral defining  $m(r)$  with respect to  $r$ :

$$\frac{dm}{dr} = 4\pi r^2 \rho(r). \quad (1.4)$$

The description is completed by specifying the "equation of state", an intrinsic property of the matter giving the pressure,  $P(\rho)$ , required to maintain it at a given density. Using the identity

$$\frac{dP}{dr} = \frac{d\rho}{dr} \frac{dP}{d\rho}, \quad (1.5)$$

Eq.(1.3) can be written as

$$\frac{d\rho}{dr} = -\left(\frac{dP}{d\rho}\right)^{-1} \frac{G m(r)}{r^2} \rho(r). \quad (1.6)$$

Equations (1.4) and (1.6) are two coupled first-order differential equations that determine the structure of the star for a given equation of state. The values of the dependent variables at  $r = 0$  are  $\rho = \rho_c$ , the central density, and  $m = 0$ . Integration outward in  $r$  then gives the density profile, the radius of the star,  $R$ , being determined by the point at which  $\rho$  vanishes. (On very general grounds, we expect the density to decrease with increasing distance from the center.) The total mass of the star is then  $M = m(R)$ . Since both  $R$  and  $M$  depend upon  $\rho_c$ , variation of this parameter allows stars of different mass to be studied.

## 1.2 Background on Quantum Mechanics and Pressure Ionization

Some introductory background material is well summarized in the on-line notes of Mike Guidry (Physics, U. of Tennessee) at the web page:

<http://eagle.phys.utk.edu/guidry/astro615/>

With some light editing we quote portions of Guidry's Chapter 3 here:

### Quantum Mechanics and Equations of State

Stellar equations of state reflect microscopic properties of the gas in stars. At low densities this gas tends to behave classically, but the correct microscopic theory of matter is quantum mechanics and at higher densities a quantum description becomes essential to an accurate treatment. The requisite physics can be understood conceptually in terms of three basic ideas.

1. de Broglie Wavelength: A particle at the microscopic level takes on wave properties characterized by a de Broglie wavelength  $\lambda = h/p$ , where  $p$  is the momentum [magnitude] of the particle and  $h$  is Planck's constant. Thus in quantum mechanics the location of a particle becomes fuzzy, spread out over a spatial interval comparable to the de Broglie wavelength

2. Quantum Statistics: All elementary particles may be classified as either fermion or bosons. These classifications have to do with how aggregates of elementary particles behave. Fermions (such as electrons, or neutrons and protons if we neglect their internal quark and gluon structure) obey Fermi-Dirac statistics. The most notable consequence is the Pauli exclusion principle: no two fermions can have an identical set of quantum numbers. All elementary particles of *half-integer spin* are fermions. Bosons (photons are the most important example for our purposes) obey Bose-Einstein statistics. Unlike fermions, there is no restriction on how many bosons can occupy the same quantum state. All elementary particles of *integer spin* are bosons. Matter is made from fermions (electrons, protons, neutrons, . . .). Forces between particles are mediated by the exchange of bosons (for example, the electromagnetic force results from an exchange of photons between charged particles).

3. Degeneracy: The Pauli exclusion principle implies that in a many-fermion system each fermion must be in a different quantum state. Thus the lowest energy state [of the system] results from filling energy levels from the bottom up. Degenerate matter corresponds to a many-fermion state in which all the lowest energy levels are filled and all the higher ones are unoccupied. Degenerate matter occurs frequently at high densities and has a very unusual equation of state with a number of implications for astrophysics.

### Equations of State for Degenerate Gases

Degenerate equations of state play an important role in a variety of astrophysical applications. For example, in white dwarf stars the electrons are highly degenerate, and in neutron stars the neutrons are highly degenerate. Let us look at this in a little more detail for the case of degenerate electrons. We first demonstrate that (as a consequence of quantum mechanics) most stars are completely ionized over much of their volume because ionization can be induced by sufficiently high pressure, even at low temperature. This implies the possibility of producing a (relatively) cold gas of electrons, which is the necessary condition for a degenerate electron equation of state.

### Pressure Ionization

Suppose the radius of each atom is  $r$  and the average spacing between atoms is  $d$ . We assume that the stellar material consists only of ions of a single species and the electrons produced by ionizing that species. Electrons in the atoms obey Heisenberg relations of the form  $\Delta p_x \cdot \Delta x \geq \hbar$ , with  $\hbar \equiv h/(2\pi)$ , and with  $\Delta x$ ,  $\Delta p_x$  being respectively the uncertainty in the  $x$  component of electron position and the uncertainty in the  $x$  component of the electron momentum, with similar relations for the  $y$  and  $z$  components.

In terms of the electron momentum magnitude  $p$  and the corresponding electron de Broglie wavelength  $\lambda_e$ , a consequence of the Heisenberg relations is that  $p \cdot \lambda_e \geq \hbar$ . Taking the average volume needed per electron to be  $V_0 \approx (\lambda_e)^3$ , we can write this as  $p \geq \hbar/V_0^{1/3}$

The uncertainty principle produces ionization when the effective volume of the atoms becomes too small to confine the electrons. The average volume needed per electron  $V_0$  is related to the average volume needed per ion  $V_i$  by  $Z V_0 = V_i$ , since there are  $Z$  electrons per ion. Thus  $p \geq (\hbar Z^{1/3})/V_i^{1/3}$ .

From atomic physics, the atomic radius  $r$  may be approximated by  $r \simeq a_0 Z^{-1/3}$ , where  $a_0 = 5.3 \times 10^{-9}$  cm is the Bohr radius. If the star is composed entirely of an element with atomic number  $Z$  and mass number  $A$ , there are  $Z$  electrons in each sphere of radius  $d$  (the average spacing between atoms) and the average number density of electrons  $n_e$  is related to  $d$  by  $n_e = Z/(\frac{4}{3}\pi d^3)$ .

This can be solved for  $d$  to give  $d = ((3Z)/(4\pi n_e))^{1/3}$ , which shows that  $d$  becomes smaller as  $n_e$  becomes larger. If  $d < r$  we expect pressure ionization. With increasing density fewer locally bound states are possible until none remain and the electrons are all ionized. Thus, sufficiently high density can cause complete ionization, even at zero temperature.

Since there are  $A$  nucleons in each volume of radius  $d$ , the mass density  $\rho$  is  $\rho \approx (A M_p)/(\frac{4}{3}\pi d^3)$ , in which  $M_p$  is the proton mass. and requiring that  $d \simeq r \simeq a_0 Z^{-1/3}$  defines a critical density  $\rho_{crit} = (Z A M_p)/(\frac{4}{3}\pi a_0^3)$ . We may expect that for densities greater than  $\rho_{crit}$  there will be almost complete pressure ionization, irrespective of the temperature.

For pure Carbon ( $Z = 6, A = 12$ ),  $\rho_{crit} = 230 \text{ g cm}^{-3}$ . For pure Oxygen ( $Z = 8, A = 16$ ),  $\rho_{crit} = 410 \text{ g cm}^{-3}$ . For pure Iron ( $Z = 26, A = 56$ ),  $\rho_{crit} = 4660 \text{ g cm}^{-3}$ . The actual typical density of a half Carbon and half Oxygen white dwarf is of the order  $10^6 \text{ g cm}^{-3}$ , much greater than  $\rho_{crit}$ .

These considerations imply that Saha ionization equations, which are derived assuming ionization to be caused by thermal effects, are no longer reliable in the deep interior of stars.

### 1.3 The Approximate Equation of State

Koonin assumes the reader is aware of the rules for counting the number of momentum microstates available to a particle because of quantum mechanics. A quick and dirty argument is to say

Since the uncertainty principle requires  $\Delta x \Delta p \gtrsim 2\pi\hbar$ , we can associate  $(d^3 x d^3 p)/(2\pi\hbar)^3$  micro-states with a phase space volume  $(d^3 x d^3 p)$ . Therefore, the number of momentum states with the  $x$  component of momentum in the range  $(p_x, p_x + dp_x)$ , the  $y$  component of momentum in the range  $(p_y, p_y + dp_y)$ , the  $z$  component of momentum in the range  $(p_z, p_z + dp_z)$ , and the position vector somewhere in a volume  $V$  is  $V d^3 p/(2\pi\hbar)^3$ , where  $d^3 p = dp_x dp_y dp_z$ .

You can find a better discussion in any text covering quantum statistics, such as **Fundamentals of Statistical and Thermal Physics** by Frederick Reif, (1965), Sec. 9.9: Quantum States of a Single Particle, where periodic boundary conditions are enforced on the quantum wave function of a particle.

Again quoting Koonin (with considerable editing, and retaining c.g.s. units):

We must now determine the equation of state appropriate for a white dwarf. ... we will assume that the matter consists of large nuclei and their electrons. The nuclei, being heavy, contribute nearly all of the mass but make almost no contribution to the pressure since they hardly move at all [and are non-degenerate]. The electrons, however, contribute virtually all of the pressure but essentially none of the mass. We will be interested in densities far greater than that of ordinary matter, where the electrons are no longer bound to individual nuclei, but rather move freely through the material. A good model is then a free Fermi gas of electrons at zero temperature, treated with relativistic kinematics.

The number of nucleons per unit volume at radius  $r$  is approximately  $\rho(r)/M_p$ , where  $M_p$  is the proton mass (we neglect the small difference between the neutron and proton masses). If  $Y_e$  is the number of electrons per nucleon, then the number density (concentration) of electrons at radius  $r$  is

$$n(r) \simeq Y_e \frac{\rho(r)}{M_p} \quad (1.7)$$

If the nuclei are all  $^{56}\text{Fe}$ , then  $Y_e = 26/56 = 0.464$ , while  $Y_e = 1/2$  if the nuclei are  $^{12}\text{C}$ ; electrical neutrality of the matter requires one electron for every proton.

The free Fermi gas – independent identical fermions obeying Fermi-Dirac statistics (shielding effects allow us to approximately ignore Coulomb interactions) – is studied by considering a small but macroscopic volume  $V$  containing a

group of  $N$  electrons at a given direction and located in the interval  $[r, r + dr]$  that occupy the lowest available energy plane-wave states with magnitude of momentum  $0 \leq p \leq p_f$ . Remembering the two-fold spin degeneracy of each plane wave, we have

$$N = 2V \int_0^{p_f} \frac{d^3p}{(2\pi\hbar)^3} = 2V \int_0^{p_f} \frac{4\pi p^2 dp}{(2\pi\hbar)^3} \quad (1.8)$$

which determines the value of the local ‘‘Fermi momentum’’ magnitude  $p_f$  in terms of the local electron density:

$$p_f(r) = [3\pi^2 \hbar^3 n(r)]^{1/3} \quad (1.9)$$

where  $n(r) = N/V$ . The total kinetic plus rest mass energy of this group of  $N$  electrons occupying the lowest possible momentum eigenstates is

$$E = 2V \int_0^{p_f} \frac{4\pi p^2 dp}{(2\pi\hbar)^3} \varepsilon_p \quad (1.10)$$

where  $\varepsilon_p = [p^2 c^2 + m_e^2 c^4]^{1/2}$  is the relativistic energy of an electron with mass  $m_e$  and momentum magnitude  $p$ .

Changing the variable of integration from  $p$  to  $y = p/(m_e c)$ , we need the integral

$$\int_0^{p_f} p^2 dp \sqrt{p^2 c^2 + m_e^2 c^4} = m_e^4 c^5 \int_0^x y^2 dy \sqrt{1 + y^2} \quad (1.11)$$

in which dimensionless  $x$  is:

$$x = \frac{p_f}{m_e c} = \left[ \frac{n(r)}{n_0} \right]^{1/3} = \left[ \frac{\rho(r)}{\rho_0} \right]^{1/3}, \quad (1.12)$$

$n_0$  is the local electron density at which the local Fermi momentum  $p_f(r) = m_e c$

$$n_0 = \frac{1}{3\pi^2} \frac{1}{\lambda_c^3} = 5.87 \times 10^{29} \text{ cm}^{-3}, \quad (1.13)$$

$\lambda_c$  is the electron Compton wavelength

$$\lambda_c = \frac{\hbar}{m_e c} = 3.86 \times 10^{-11} \text{ cm} \quad (1.14)$$

and  $\rho_0$  is the local mass density of matter when the local electron density is  $n_0$ :

$$\rho_0 = \frac{M_p n_0}{Y_e} = 9.82 \times 10^5 Y_e^{-1} \text{ gm cm}^{-3}. \quad (1.15)$$

The indefinite integral needed is a standard integral, but we can also practice using Maxima:

```
(%i1) assume(x>0);
(%o1) [x > 0]
(%i2) ival:integrate(y^2*sqrt(1+y^2),y,0,x);
(%o2) (sqrt(x^2+1)*(2*x^3+x)-asinh(x))/8
(%i3) display2d:true$
(%i4) ival;
          2      3
      sqrt(x  + 1) (2 x  + x) - asinh(x)
(%o4) -----
          8
```

We can then use the identity (for  $a > 0$ ):

$$\ln \left( \frac{x + \sqrt{a^2 + x^2}}{a} \right) = \sinh^{-1}(x/a). \quad (1.16)$$

to finally get the local energy of this group of electrons in the form

$$E = V n_0 m_e c^2 x^3 \varepsilon(x) \quad (1.17)$$

in which

$$\varepsilon(x) = \frac{3}{8x^3} \left[ x(1 + 2x^2) \sqrt{1 + x^2} - \ln \left( x + \sqrt{1 + x^2} \right) \right]. \quad (1.18)$$

In the usual thermodynamic manner, the local pressure is related to how the energy of this group of  $N$  electrons changes with volume  $V$  at fixed  $N$ :

$$P(r) = -\frac{\partial E}{\partial V} = - (n_0 m_e c^2) \left\{ x^3 \varepsilon(x) + V \frac{d}{dx} [x^3 \varepsilon(x)] \frac{\partial x}{\partial V} \right\} \quad (1.19)$$

From Eq.(1.12),  $x = [n(r)/n_0]^{1/3} = [N/(n_0 V)]^{1/3}$ , so

$$\frac{\partial x}{\partial V} = -\frac{x}{3V} \quad (1.20)$$

and the local pressure then takes the form

$$P(x) = \frac{1}{3} n_0 m_e c^2 x^4 \varepsilon'(x) \quad (1.21)$$

where  $\varepsilon'(x) = d\varepsilon/dx$ . The pressure has the units: energy per unit volume or force per unit area (since energy has the units: force times distance).

We then need the derivative  $dP/d\rho$  to make use of Eq.(1.6). We use the chain rule

$$\frac{dP}{d\rho} = \frac{dP}{dx} \frac{dx}{d\rho}. \quad (1.22)$$

From Eq.(1.12),  $x = [\rho(r)/\rho_0]^{1/3}$ , or  $\rho/\rho_0 = x^3$ , and differentiating both sides of the latter equation with respect to  $\rho$  yields

$$\frac{dx}{d\rho} = \frac{Y_e}{3 n_0 M_p x^2} \quad (1.23)$$

so

$$\frac{dP}{d\rho} = Y_e \frac{m_e c^2}{M_p} \gamma(x) \quad (1.24)$$

in which dimensionless  $\gamma(x)$  is

$$\gamma(x) = \frac{1}{9 x^2} \frac{d}{dx} [x^4 \varepsilon'(x)] = \frac{x^2}{3 \sqrt{1+x^2}} \quad (1.25)$$

If you check the final simplified form of  $\gamma(x)$  by hand, you can use

$$\varepsilon'(x) = \frac{3}{x} [\sqrt{1+x^2} - \varepsilon(x)]. \quad (1.26)$$

We can also use Maxima (after some trial and error) to verify that

$$\sqrt{1+x^2} \frac{d}{dx} [x^4 \varepsilon'(x)] = 3x^4. \quad (1.27)$$

In our Maxima work we let **ddx** be the quantity  $\frac{d}{dx} [x^4 \varepsilon'(x)]$ , and let **rn** be  $\sqrt{1+x^2}$  times the numerator of **ddx**, and let **rd** be the denominator of **ddx**. We then want the ratio **rn/rd** to be  $3x^4$ . Some experimentation is needed to find a path to a simplified form which results from cancellation of factors in the numerator with factors in the denominator. The Maxima functions **num**, **denom**, **expand**, **factor**, and **ratsimp** are useful things to experiment with when starting with a complicated expression. Often, it is useful to work with the numerator and denominator separately, and combine the results at the end. In our work (always using the Xmaxima interface), we routinely set **display2d:false** in our startup file.

```
(%i1) eps : 3*((x+2*x^3)*sqrt(1+x^2) - log(x+sqrt(1+x^2)))/8/x^3$
(%i2) ddx : ratsimp(diff(x^4*diff(eps,x),x));
(%o2) (12*x^7+sqrt(x^2+1)*(12*x^6+3*x^4)+9*x^5)
/(4*x^4+sqrt(x^2+1)*(4*x^3+3*x)+5*x^2+1)
(%i3) rn : factor(expand(sqrt(1+x^2)*num(ddx)));
(%o3) 3*x^4*(4*x^3*sqrt(x^2+1)+3*x*sqrt(x^2+1)+4*x^4+5*x^2+1)
(%i4) rd : factor(expand(denom(ddx)));
(%o4) 4*x^3*sqrt(x^2+1)+3*x*sqrt(x^2+1)+4*x^4+5*x^2+1
(%i5) rnd : rn/rd;
(%o5) 3*x^4
```

Using Eq.(1.3) in Eq.(1.6) we get an explicit differential equation governing the evolution of  $\rho$  (recall that dimensionless  $\gamma$  is a function of  $x$  which is a function of  $\rho$  which is a function of  $r$ ):

$$\frac{d\rho}{dr} = - \left( \frac{M_p}{m_e c^2 Y_e} \right) \frac{G m(r)}{\gamma r^2} \rho(r). \quad (1.28)$$

### 1.4 Internal Kinetic Energy of the Star

Using the same type of calculation as done in Eq.(1.10), the kinetic energy of the group of  $N$  electrons in a given direction and having radii in the range  $[r, r + dr]$  and occupying a small but macroscopic volume  $V$  is

$$E_{kinetic} = 2V \int_0^{p_f} \frac{4\pi p^2 dp}{(2\pi\hbar)^3} (\varepsilon_p - m_e c^2), \quad (1.29)$$

where  $\varepsilon_p = [p^2 c^2 + m_e^2 c^4]^{1/2}$  is the relativistic energy of an electron with mass  $m_e$  and momentum magnitude  $p$ . Dividing  $E_{kinetic}$  by  $V$  yields the local kinetic energy density  $k(r)$

$$k(r) = n_0 m_e c^2 x^3 (\varepsilon(x) - 1), \quad (1.30)$$

in which  $\varepsilon(x)$  is defined in Eq.(1.18). The internal kinetic energy of the star (neglecting the nuclei) is then

$$K = \int_0^R k(r) 4\pi r^2 dr. \quad (1.31)$$

### 1.5 Internal Gravitational Energy of the Star

The gravitational energy due to adding a small mass element  $\delta m = \rho(r) \delta V$  with center at  $(r, \theta, \phi)$  to a body of mass  $m(r)$  and radius  $r$  is  $\delta U = -\frac{Gm(r)}{r} \delta m$ . With  $\delta V = (r d\theta)(r \sin\theta d\phi) dr = r^2 d\Omega dr$ , integration over the solid angle  $d\Omega$  yields  $4\pi$ . Integration over  $r$  from 0 to  $R$  then accumulates the total internal gravitational energy of the star:

$$U = - \int_0^R \frac{Gm(r)}{r} \rho(r) 4\pi r^2 dr \quad (1.32)$$

## 2 Scaling the Differential Equations

Quoting Koonin again:

It is often useful to reduce equations describing a physical system to dimensionless form, both for physical insight and for numerical convenience (i.e., to avoid dealing with very large or very small numbers in the computer). To do this for the equations of white dwarf structure, we introduce dimensionless radius, density, and mass variables:

$$r = R_0 \bar{r}, \quad \rho = \rho_0 \bar{\rho}, \quad m = M_0 \bar{m} \quad (2.1)$$

with the radius and mass scales,  $R_0$  and  $M_0$  to be determined for convenience. Substituting into Eqs. (1.4, 1.28) yields

$$\frac{d\bar{m}}{d\bar{r}} = \left( \frac{4\pi R_0^3 \rho_0}{M_0} \right) \bar{r}^2 \bar{\rho} \quad (2.2)$$

and

$$\frac{d\bar{\rho}}{d\bar{r}} = - \left( \frac{G M_p M_0}{m_e c^2 Y_e R_0} \right) \frac{\bar{m} \bar{\rho}}{\gamma \bar{r}^2}. \quad (2.3)$$

If we now choose  $M_0$  and  $R_0$  so that the coefficients in parentheses in these two equations are unity, we find

$$R_0 = \left( \frac{m_e c^2 Y_e}{4\pi \rho_0 G M_p} \right) = 7.71 \times 10^8 Y_e \text{ cm}, \quad (2.4)$$

and

$$M_0 = 4\pi R_0^3 \rho_0 = 5.66 \times 10^{33} Y_e^2 \text{ gm}, \quad (2.5)$$

and the dimensionless differential equations are

$$\frac{d\bar{m}}{d\bar{r}} = \bar{r}^2 \bar{\rho}, \quad \frac{d\bar{\rho}}{d\bar{r}} = - \frac{\bar{m} \bar{\rho}}{\gamma \bar{r}^2}. \quad (2.6)$$

These equations are completed by recalling that  $\gamma$  is given by Eq. (1.25) with  $x = \bar{\rho}^{1/3}$ .

$$\gamma = \frac{\bar{\rho}^{2/3}}{3\sqrt{1 + \bar{\rho}^{2/3}}} \quad (2.7)$$

This pair of equations is then integrated from  $\bar{r} = 0$ ,  $\bar{\rho} = \bar{\rho}_c$ ,  $\bar{m} = 0$  to the value of  $\bar{r}$  at which  $\bar{\rho} = 0$ , which defines the dimensionless radius of the star  $\bar{R}$ , and the dimensionless mass of the star is then  $\bar{M} = \bar{m}(\bar{R})$ . The scaled solution then depends on the dimensionless central mass density  $\bar{\rho}_c$ .



### 3 Scaling the Kinetic and Gravitational Energy

We can write the sum of the kinetic and gravitational energy of the star  $E_{star}$  (neglecting electrostatic contributions) in terms of an energy scale factor  $E_0$  which carries the dimensions, and is defined in terms of  $M_0$ ,  $R_0$ , and  $\rho_0$ .

$$E_{star} = K + U = E_0 \bar{E}_{star} \quad (3.1)$$

in which

$$E_0 = 4 \pi n_0 m_e c^2 R_0^3 = 4 \pi G M_0 \rho_0 R_0^2 = 2.77 Y_e^3 \times 10^{51} \text{ ergs} \quad (3.2)$$

Then the dimensionless energy of the star (neglecting rest mass energy) is

$$\bar{E}_{star} = \bar{K} + \bar{U} \quad (3.3)$$

in which  $\bar{K}$  is the dimensionless internal kinetic energy of the electrons (the nuclei kinetic energy is ignored), and  $\bar{U}$  is the dimensionless gravitational energy due to the nuclei (ignoring the mass of the electrons).

$$\bar{K} = \int_0^{\bar{R}} \bar{k}(\bar{r}) d\bar{r} \quad (3.4)$$

in which

$$\bar{k}(\bar{r}) = \bar{r}^2 \bar{\rho} (\bar{\varepsilon}(\bar{\rho}) - 1) \quad (3.5)$$

with

$$\bar{\varepsilon}(\bar{\rho}) = \frac{3}{8\bar{\rho}} \left\{ \bar{\rho}^{1/3} \left( 1 + 2\bar{\rho}^{2/3} \right) \sqrt{1 + \bar{\rho}^{2/3}} - \ln \left[ \bar{\rho}^{1/3} + \sqrt{1 + \bar{\rho}^{2/3}} \right] \right\}. \quad (3.6)$$

The dimensionless gravitational energy of the star is

$$\bar{U} = \int_0^{\bar{R}} \bar{u}(\bar{r}) d\bar{r} \quad (3.7)$$

in which

$$\bar{u}(\bar{r}) = -\bar{r} \bar{\rho}(\bar{r}) \bar{m}(\bar{r}). \quad (3.8)$$

### 4 Scaled Dimensionless Differential Equations Summary

We drop the over-bars on the symbols for the scaled dimensionless variables in this and the next two sections, and in the code:  $\bar{r} \rightarrow r$ ,  $\bar{m} \rightarrow m$ , and  $\bar{\rho} \rightarrow \rho$ , for simplicity; we can always restore the over-bars before converting results to c.g.s. units. The pair of ordinary differential equations to be integrated are then:

$$\frac{dm}{dr} = \rho r^2 \quad (4.1)$$

$$\frac{d\rho}{dr} = -\frac{m\rho}{\gamma(\rho)r^2} \quad (4.2)$$

in which

$$\gamma(\rho) = \frac{\rho^{2/3}}{3\sqrt{1 + \rho^{2/3}}} \quad (4.3)$$

This pair of first order differential equations is then integrated from  $r = 0$ ,  $\rho = \rho_c$ ,  $m = 0$  to the value of  $r$  at which  $\rho = 0$ , which defines the dimensionless radius of the star  $R$ , and the dimensionless mass of the star is then  $M = m(R)$ . The scaled solution then depends on the dimensionless central mass density  $\rho_c$ .

As we discuss below, for  $r \ll 1$

$$m(r) \approx \frac{1}{3} \rho_c r^3 \quad (4.4)$$

$$\rho(r) \approx \rho_c - \frac{\rho_c^2 r^2}{6\gamma_c} \quad (4.5)$$

where  $\rho_c = \rho(r = 0)$  and  $\gamma_c = \gamma(\rho_c)$ .

## 5 Expansion of $\bar{m}(\bar{r})$ and $\bar{\rho}(\bar{r})$ about $\bar{r} = 0$

To avoid “division by zero” errors at  $r = 0$ , we need to Taylor expand the dependent variables about  $r = 0$ , and start the Runge-Kutta integration a tiny distance away from  $r = 0$ .

We know that  $m(r = 0) = 0$ . For  $r \ll 1$  we can replace  $\rho$  by  $\rho_c$  in Eq.(4.1) to get  $dm/dr \approx \rho_c r^2$ , which vanishes as  $r \rightarrow 0$ . Using Eqs.(4.1 and 4.2), we can evaluate the second derivative of  $m(r)$  as

$$\frac{d^2 m}{dr^2} = \frac{d}{dr}(\rho r^2) = -\frac{m \rho}{\gamma} + 2 \rho r \quad (5.1)$$

which also vanishes as  $r \rightarrow 0$  and  $m(r) \rightarrow 0$  together. Since  $(d^2 m/dr^2)_0 = 0$ , the expansion of  $m$  about  $r = 0$  begins with the  $r^3$  term. For some positive value of  $a$ ,  $m \approx a r^3$  for small  $r$ . Then the first derivative of  $a r^3$  must equal  $\rho_c r^2$  for small  $r$ , so  $a = \rho_c/3$  and  $m(r) \approx \frac{1}{3} \rho_c r^3$ .

We can then use  $m(r) \propto r^3$  for  $r \rightarrow 0$  in Eq.(4.2) to see that for small  $r$ ,  $d\rho/dr \propto r$  and hence vanishes at  $r = 0$ . We can use Maxima to evaluate  $d^2 \rho/dr^2$  and its limit as  $r \rightarrow 0$ . In this work, **gam** represents  $\gamma(\rho)$ , **rh** represents  $\rho$ , and we mentally let these variables (as well as  $d\gamma/d\rho$ ) take on their finite  $r = 0$  values in our limiting process. Our code uses the **depends** Maxima function, which allows the differentiation to automatically use the “chain rule”,

$$\frac{d\gamma(\rho)}{dr} = \frac{d\gamma(\rho)}{d\rho} \frac{d\rho}{dr} \quad (5.2)$$

```
(%i1) depends(gam,rh);
(%o1) [gam(rh)]
(%i2) depends([m,rh],r);
(%o2) [m(r),rh(r)]
(%i3) drh : -m*rh/gam/r^2$
(%i4) d2rh : diff(drh,r);
(%o4) 'diff(gam,rh,1)*m*rh*'diff(rh,r,1)/(gam^2*r^2)
-m*'diff(rh,r,1)/(gam*r^2)-'diff(m,r,1)*rh/(gam*r^2)
+2*m*rh/(gam*r^3)
(%i5) d2rh_1 : subst(['diff(m,r,1)= rh*r^2,'diff(rh,r,1)=-m*rh/gam/r^2],d2rh);
(%o5) -'diff(gam,rh,1)*m^2*rh^2/(gam^3*r^4)
-rh^2/gam+2*m*rh/(gam*r^3)+m^2*rh/(gam^2*r^4)
(%i6) d2rh_2 : subst(m = rh*r^3/3,d2rh_1);
(%o6) -'diff(gam,rh,1)*r^2*rh^4/(9*gam^3)+r^2*rh^3/(9*gam^2)-rh^2/(3*gam)
(%i7) d2rh_3 : limit(d2rh_2,r,0);
(%o7) -rh^2/(3*gam)
```

Thus

$$\left(\frac{d^2 \rho}{dr^2}\right)_0 \approx -\frac{\rho_c^2}{3 \gamma_c} \quad (5.3)$$

and hence  $\rho(r) \approx \rho_c - \frac{r^2 \rho_c^2}{6 \gamma_c}$ .

## 6 Behavior of $\bar{\rho}(\bar{r})$ Near the Surface

Near the surface  $m(r) \approx M$  and  $r \approx R$ , so in the differential equation Eq.(4.2) we insert  $\gamma(\rho)$  from Eq.(4.3) and Taylor expand the dependence on  $\rho$  about  $\rho = 0$  for fixed  $m(r) \approx M$  and  $r \approx R$ .

```
(%i1) gam : rh^(2/3)/3/sqrt(1+rh^(2/3))$
(%i2) drh_dr : -m*rh/gam/r^2;
(%o2) -3*m*sqrt(rh^(2/3)+1)*rh^(1/3)/r^2
(%i3) drh_surf : taylor(drh_dr,rh,0,3);
(%o3) -3*m*rh^(1/3)/r^2-3*m*rh/(2*r^2)+3*m*rh^(5/3)/(8*r^2)
-3*m*rh^(7/3)/(16*r^2)+15*m*rh^3/(128*r^2)
(%i4) drh_surf : first(drh_surf);
(%o4) -3*m*rh^(1/3)/r^2
(%i5) drh_surf : subst([m=M,r=R],drh_surf);
(%o5) -3*rh^(1/3)*M/R^2
```

We then solve the equation

$$\frac{d\rho}{dr} \approx -\frac{3M}{R^2} \rho^{1/3} \quad (6.1)$$

by separation of variables, dividing both sides by  $\rho^{1/3}$  and multiplying both sides by  $dr$ , and then integrating both sides over corresponding intervals. This gives (near the surface)

$$\rho(r) \approx \left( \frac{2M(R-r)}{R^2} \right)^{3/2} \propto (R-r)^{3/2} \quad (6.2)$$

and using this in Eq.(4.1) gives (near the surface)

$$\frac{dm}{dr} \approx r^2 \left( \frac{2M(R-r)}{R^2} \right)^{3/2} \quad (6.3)$$

which shows that  $dm/dr \rightarrow 0$  as  $r \rightarrow R$ , while  $d\rho/dr \propto -\sqrt{R-r}$ .

## 7 Maxima Code dwarf1(rhoc, dr, rtol)

The Maxima code `dwarf1` is in the file `project2.mac`, and calls the single step Runge-Kutta code `rk4_step` (also in `project2.mac`). The code `rk4_step` was also used in Example 2.

```

/* dwarf1(rhoc, dr, rtol) integrates white dwarf ode's using rk4_step with step size dr,
(after taylor expanding the dependent variables out to r1=small) watching
for rho to pass from positive to negative.
If rho is found to be negative, we return to the previous step and let rtol be the
step size, integrating forward again until rho is found to be negative, then
taking the previous step as the final value of r,m and rho
and using that value of r to define the radius R and that value of m(r) to define M.
In this code r stands for Koonin's dimensionless rbar, where r[cgs] = R0 * rbar,
m stands for Koonin's dimensionless mbar, where m[cgs] = M0*mbar,
and rho stands for Koonin's dimensionless rhobar, where rho[cgs] = rho0*rhobar.
The signal that rho is negative is that rk4_step returns a value for
rho that is not a floating point number, containing things like
(-1)^0.3333.

(%i4) gam : rho^(2/3)/3/sqrt(1 + rho^(2/3))$
(%i5) rk4_step([rho*r^2, -m*rho/gam/r^2],
              [m,rho],[0.70582,0.0036087],[r,2.5,0.1]);
(%o5) [2.6,0.70678,
       2.7270606E-4-4.192083E-4*(-1)^0.33333*(0.0064245*(-1)^0.66667+1.0)^0.5]

*/

dwarf1(rhoc, dr, rtol) :=
block([r1:1e-12, gam, gamc, m1, rho1, rkso1n,rkstep,
      rho,r,m, previous, num:1, nmax:1000, numer:true],

      gam : rho^(2/3)/3/sqrt(1 + rho^(2/3)),
      gamc : subst( rho = rhoc, gam),

      /* taylor expansion away from r = 0 */

      m1 : rhoc*r1^3/3,
      rho1 : rhoc - r1^2*rhoc^2/gamc/6,
      rkso1n : [ [r1,m1,rho1], [0, 0, rhoc]],

      /* first do loop: search for rho = 0 position using step dr */

      do ( rkstep : rk4_step([rho*r^2, -m*rho/gam/r^2],
                            [m,rho],[m1,rho1],[r,r1,dr]),
          rho1 : rkstep[3],
          if not floatnump(rho1) then return(),
          if num = nmax then (
            print(" num = nmax; abort integration"),
            return()),
          rkso1n : cons(rkstep,rkso1n),

```

```

    r1 : rkstep[1],
    m1 : rkstep[2],
    num : num + 1),

/* second do loop: search for rho=0 location using step = rtol */

num : 1,
previous : first(rksoln),
r1 : previous[1],
m1 : previous[2],
rho1 : previous[3],
do (rkstep : rk4_step([rho*r^2, -m*rho/gam/r^2],
                    [m,rho],[m1,rho1],[r,r1,rtol]),
    rho1 : rkstep[3],
    if not floatnump(rho1) then return(),
    if num = nmax then return(),
    rksoln : cons(rkstep,rksoln),
    r1 : rkstep[1],
    m1 : rkstep[2],
    num : num + 1),
reverse(rksoln))$

```

Here is an example of the use of `dwarf1` for  $\bar{\rho}_c = 1$ .

```

(%i1) load(project2);
(%o1) "c:/k2/project2.mac"
(%i2) soln : dwarf1(1,0.1,0.01)$
(%i3) fll(soln);
(%o3) [[0,0,1],[2.49,0.70664,9.7524488E-5],35]
(%i4) soln;
(%o4) [[0,0,1],[1.0E-12,3.3333333E-37,1.0],[0.1,3.297978E-4,0.99118],
 [0.2,0.0026136,0.97034],[0.3,0.0086434,0.9368],[0.4,0.019904,0.89197],
 [0.5,0.037461,0.83773],[0.6,0.061897,0.77623],[0.7,0.09329,0.70971],
 [0.8,0.13124,0.6404],[0.9,0.17493,0.57035],[1.0,0.22321,0.50135],
 [1.1,0.27471,0.43491],[1.2,0.32795,0.3722],[1.3,0.38141,0.31406],
 [1.4,0.43364,0.26104],[1.5,0.48333,0.21343],[1.6,0.52934,0.17131],
 [1.7,0.57077,0.13456],[1.8,0.60694,0.10298],[1.9,0.6374,0.076269],
 [2.0,0.66198,0.054064],[2.1,0.68071,0.036007],[2.2,0.69387,0.021749],
 [2.3,0.70197,0.010999],[2.4,0.70582,0.0036087],
 [2.41,0.70601,0.003056],[2.42,0.70617,0.0025395],
 [2.43,0.70631,0.0020602],[2.44,0.70642,0.0016195],
 [2.45,0.7065,0.0012191],[2.46,0.70656,8.6174615E-4],
 [2.47,0.70661,5.5097716E-4],[2.48,0.70663,2.9262378E-4],
 [2.49,0.70664,9.7524488E-5]]

```

The function `fll(aL)` returns a list of the first and last elements of list `aL`, and also the length of `aL`, and is in `project2.mac`.

## 7.1 Maxima Plots of $\bar{m}(\bar{r})$ and $\bar{\rho}(\bar{r})$

We can use the list `soln` to make plots of the mass and the mass density as a function of radius. We use the function `take`, defined in `project2.mac`, to extract a list of just the radial positions `rL`, for example.

```

(%i5) rL : take(soln,1)$
(%i6) fll(rL);
(%o6) [0,2.49,35]
(%i7) mL : take(soln,2)$
(%i8) fll(mL);
(%o8) [0,0.70664,35]
(%i9) rhoL : take(soln,3)$
(%i10) fll(rhoL);
(%o10) [1,9.7524488E-5,35]
(%i11) plot2d([discrete,rL,mL],[style,[lines,3]],
 [xlabel,"r"], [ylabel,"m"],
 [gnuplot_preamble,"set grid"])$

```

which produces the plot of the scaled mass  $\bar{m}(\bar{r})$ :

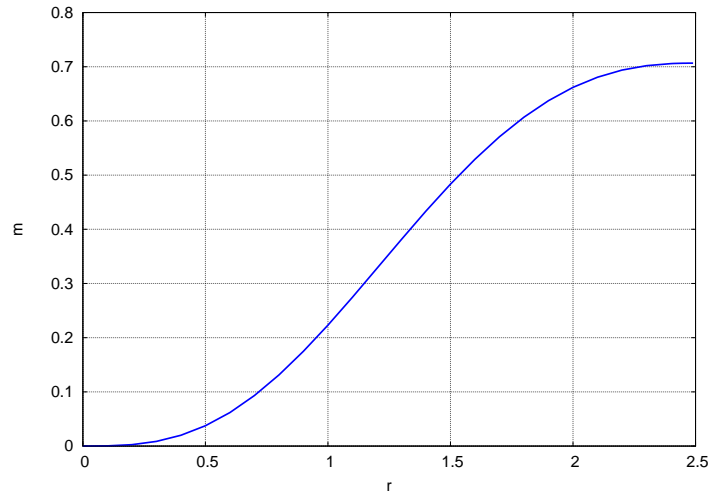


Figure 1: mass  $\bar{m}(\bar{r})$  for  $\bar{\rho}_c = 1$  and  $d\bar{r} = 0.1$

and then a plot of the scaled mass density:

```
(%i12) plot2d([discrete,rL,rhoL],[style,[lines,3]],
  [xlabel,"r"], [ylabel,"rho"],
  [gnuplot_preamble,"set grid"])$
```

which produces the plot of  $\bar{\rho}(\bar{r})$ :

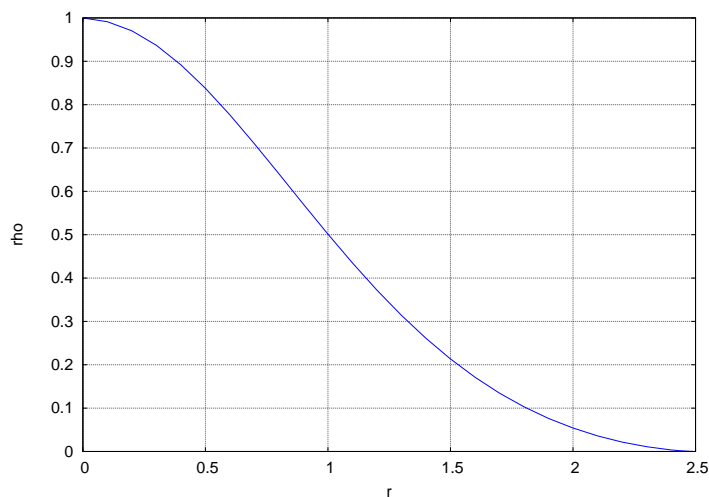


Figure 2: mass density  $\bar{\rho}(\bar{r})$  for  $\bar{\rho}_c = 1$  and  $d\bar{r} = 0.1$

If we decrease both **dr** and **rtol**, we get closer to the true solution (we need to make sure **nmax** is large enough to get to  $\bar{\rho} = 0$ ):

```
(%i13) soln2 : dwarf1(1,0.01,0.001)$
(%i14) f11(soln2);
(%o14) [[0,0,1],[2.497,0.70706,4.6097242E-6],258]
(%i15) rL2 : take(soln2,1)$
(%i16) f11(rL2);
(%o16) [0,2.497,258]
(%i17) mL2 : take(soln2,2)$
(%i18) f11(mL2);
(%o18) [0,0.70706,258]
(%i19) rhoL2 : take(soln2,3)$
(%i20) f11(rhoL2);
(%o20) [1,4.6097242E-6,258]
```

```
(%i21) plot2d([ [discrete,rL,mL], [discrete,rL2,mL2] ],
  [style,[lines,1]], [color,blue,red],
  [xlabel,"r"], [ylabel,"m"],
  [gnuplot_preamble,"set key bottom right; set grid"],
  [legend, "dr = 0.1", "dr = 0.01"])$
```

which shows the curves right on top of each other:

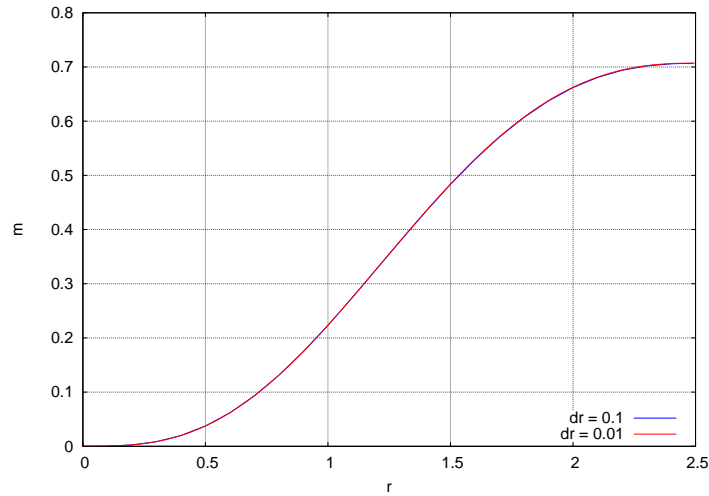


Figure 3: mass  $\bar{m}(\bar{r})$  for  $\bar{\rho}_c = 1$  and two values of  $d\bar{r}$

and now we compare the plots of  $\bar{\rho}(\bar{r})$  for these two different values of  $d\bar{r}$ :

```
(%i22) plot2d([ [discrete,rL,rhoL], [discrete,rL2,rhoL2] ],
  [style,[lines,1]], [color,blue,red],
  [xlabel,"r"], [ylabel,"rho"],
  [gnuplot_preamble,"set grid"],
  [legend, "dr = 0.1", "dr = 0.01"])$
```

which produces again curves right on top of each other:

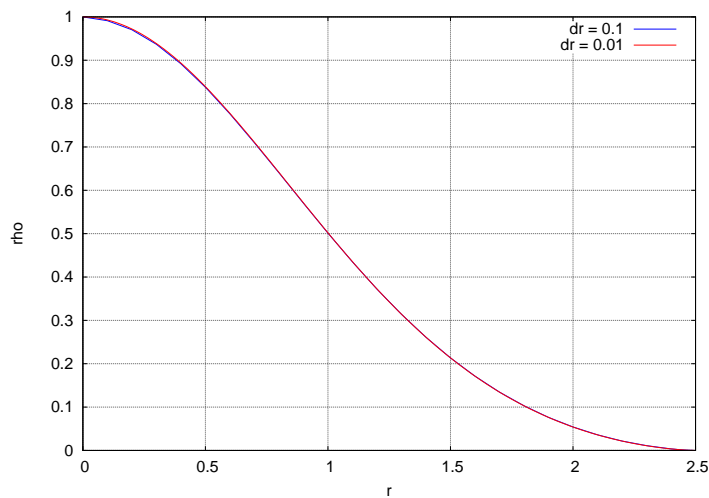


Figure 4: mass density  $\bar{\rho}(\bar{r})$  for  $\bar{\rho}_c = 1$  and two values of  $d\bar{r}$

## 8 R Code dwarf1(rhoc, dr, rtol, nmax)

R code for the function `dwarf1(rhoc,dt,rtol,nmax)` is in the file `project2.R`.

```
## dwarf1
## With rbar -> r, rho_bar -> rho, mbar -> m, rho_bar_c -> rhoc, drbar -> dr
## dwarf1(rhoc,dr,rtol,nmax) returns a R list with the vector elements rL,mL,and rhoL,
## given the central density rhoc at r = 0, the step size dr, and the rtol value and the
## maximum number of elements per vector, nmax,
## (using the externally defined derivs function when calling rk4_step)
## after taylor expanding the dependent variables out to r1=small, and then watching
## for rho to pass from positive to negative value.
## If rho is found to be negative, we back up and let rtol = smaller be the
## step size, integrating forward again until rho is found to be negative, then
## taking the last good values as the final values of r,m and rho
## and using that last good value of r to define the radius R and that
## last good value of m(r) to define M.
## Inside dwarf1, rk4_step returns the vector c(m,rho).

dwarf1 = function(rhoc, dr, rtol, nmax) {

## we don't know how long these vectors need to be

  rL = vector(length = nmax) # radius values
  mL = vector(length = nmax) # mass values
  rhoL = vector(length = nmax) # mass density values

  rL[1] = 0
  mL[1] = 0
  rhoL[1] = rhoc

## taylor expand away from r = 0 to avoid division by zero problems
  rh23 = rhoc^(2/3)
  gamc = rh23/3/sqrt(1+rh23)
  r1 = 1e-12
  m1 = rhoc*r1^3/3
  rho1 = rhoc - r1^2*rhoc^2/gamc/6
  rL[2] = r1
  mL[2] = m1
  rhoL[2] = rho1

## first loop: search for rho = 0 position using step dr
  num = 2 # number of vector elements defined so far
  repeat {
    rkstep = rk4_step( c(m1, rho1), r1, dr, derivs)
## watch for rho becoming negative
    if (is.nan(rkstep[2])) {
      cat(" new rho is negative-- go back and try again with smaller step \n")
      break }

    if (num == nmax) {
      cat(" num = ",num," = nmax; abort integration \n")
      return() }

    r1 = r1 + dr
    m1 = rkstep[1]
    rho1 = rkstep[2]
    num = num + 1
    rL[num] = r1
    mL[num] = m1
    rhoL[num] = rho1}
}
```

```

##      second loop: search for rho = 0 location using smaller step size  rtol
repeat {
  rkstep = rk4_step( c(m1, rho1), r1, rtol, derivs)
##      watch for rho becoming negative
  if (is.nan(rkstep[2])) {
##          cat(" rho is negative -- use last good value as surface \n")
          break }

  if (num == nmax) {
##          cat(" num = nmax; abort integration \n")
          return() }

  r1 = r1 + rtol
  m1 = rkstep[1]
  rho1 = rkstep[2]
  num = num + 1
  rL[num] = r1
  mL[num] = m1
  rhoL[num] = rho1}

##      construct vectors of length num for return
rrL = vector (length = num)
rmL = vector (length = num)
rrhoL = vector (length = num)
for (kk in 1:num) {
  rrL[kk] = rL[kk]
  rmL[kk] = mL[kk]
  rrhoL[kk] = rhoL[kk] }

list(rrL, rmL, rrhoL) }

```

The R code `dwarf1` makes use of the single step Runge-Kutta R code `rk4_step` (also in `project2.R`):

```

##      rk4_step(init,t,dt,func) returns y(t+dt) if one dependent variable
##      or returns the vector c(x(t+dt), vx(t+dt)) if simple harmonic oscillator (two dependent
##      variables), etc.

rk4_step = function(init, t, dt, func) {
  num.var = length(init)
  h = dt # step size
  yL = init
  k1 = func(t, yL) # vector of derivatives
  k2 = func(t + h/2, yL + h*k1/2) # vector of derivatives
  k3 = func(t + h/2, yL + h*k2/2) # vector of derivatives
  k4 = func(t + h, yL + h*k3) # vector of derivatives
  for (k in 1:num.var) yL[k] = yL[k] + h*(k1[k] + 2*k2[k] + 2*k3[k] + k4[k])/6
  if (num.var==1) yL[1] else yL}

```

and when called inside `dwarf1`, uses the derivatives function `derivs`, also defined in `project2.R`.

```

##      derivatives function 'derivs' for dm/dr = rho*r^2, drho/dr = -m*rho/gamma/r^2
##      y[1] = m, y[2] = rho
##      gamma = rho^(2/3)/3/sqrt(1+rho^(2/3))

derivs = function(r,y) {
  with( as.list(y), {
    y23 = y[2]^(2/3)
    gamma = y23/3/sqrt(1+y23)
    dm = r^2*y[2]
    drho = -y[1]*y[2]/gamma/r^2
    c(dm, drho)}}}

```



### 8.1 R Plots of $\bar{m}(\bar{r})$ and $\bar{\rho}(\bar{r})$

Here is an example of use of this R code, using the same central density as in the Maxima example. The function `mygrid` is also in `project2.R`.

```
> source("c:/k2/project2.R")
> soln = dwarf1(1,0.1,0.01,100)
> rL = soln[[1]]
> f11(rL)
  0  2.49  35
> mL = soln[[2]]
> f11(mL)
  0  0.70664406  35
> rhoL = soln[[3]]
> f11(rhoL)
  1  9.7524488e-05  35
> plot(rL,mL,type="l",lwd=3,xlab="r",ylab="m",col="blue")
> mygrid()
```

which produces the plot of  $\bar{m}(\bar{r})$  for  $\bar{\rho}_c = 1$ :

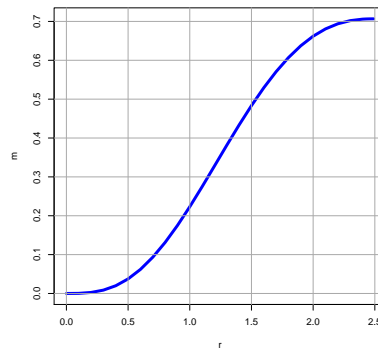


Figure 5: mass  $\bar{m}(\bar{r})$  for  $\bar{\rho}_c = 1$

and then for the mass density plot,

```
> plot(rL,rhoL,type="l",lwd=3,xlab="r",ylab="rho",col="blue")
> mygrid()
```

which produces the plot:

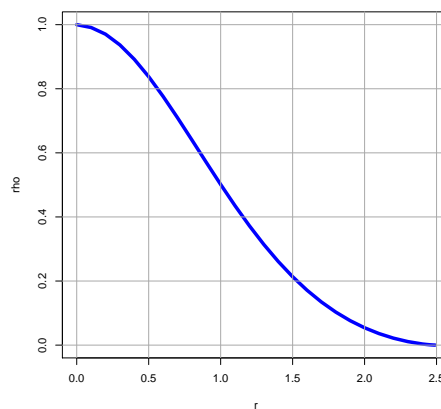


Figure 6: mass density  $\bar{\rho}(\bar{r})$  for  $\bar{\rho}_c = 1$

## 9 Energy Plots Using Maxima

Using the dimensionless kinetic energy Eqs. (3.4, 3.5), let's first show the shape of the kinetic energy integrand  $\bar{k}(\bar{r})$  and then find the numerical value of the integral  $\bar{K}$  which gives the internal kinetic energy of the star as a whole.

We use list arithmetic methods in Maxima to generate the list of needed ordinates from the list of  $\bar{r}$ -values **rL** and the list of  $\bar{\rho}$ -values **rhoL**. Recall that **dwarf1** returns a list of  $[\bar{r}, \bar{m}, \bar{\rho}]$  sub-lists.

```
(%i1) load(project2);
(%o1) "c:/k2/project2.mac"
(%i2) soln : dwarf1(1,0.1,0.01)$
(%i3) fill(soln);
(%o3) [[0,0,1],[2.49,0.70664,9.7524488E-5],35]
(%i4) rL : take(soln,1)$
(%i5) fill(rL);
(%o5) [0,2.49,35]
(%i6) mL : take(soln,2)$
(%i7) fill(mL);
(%o7) [0,0.70664,35]
(%i8) rhoL : take(soln,3)$
(%i9) fill(rhoL);
(%o9) [1,9.7524488E-5,35]
(%i10) eps(rh) := (3/8/rh)*(rh^(1/3)*(1+2*rh^(2/3))*sqrt(1+rh^(2/3)) -
log(rh^(1/3) + sqrt(1+rh^(2/3))))$
(%i11) epsml(z) := eps(z) - 1$
(%i12) kL : rL^2*rhoL*map('epsml, rhoL);
(%o12) [0,2.6047516E-25,0.0025683,0.0099318,0.02113,0.034741,0.049113,
0.062613,0.073844,0.081802,0.085953,0.086227,0.082951,0.076743,
0.068388,0.058726,0.048559,0.038578,0.029329,0.021197,0.014409,
0.0090516,0.0050974,0.0024274,8.5586504E-4,1.4609872E-4,1.1171861E-4,
8.2775005E-5,5.8920496E-5,3.9791237E-5,2.5002888E-5,1.4144578E-5,
6.7694126E-6,2.3779311E-6,3.8418924E-7]
(%i13) plot2d([discrete,rL,kL],[xlabel,"r"],[ylabel,"k(r)"],
[style,[lines,3]],[y,0,0.1],[gnuplot_preamble,"set grid"])$
```

which produces the plot of the kinetic energy integrand:

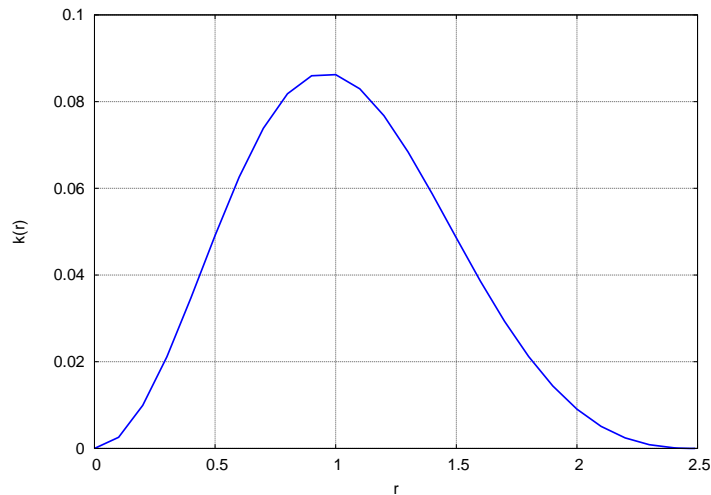


Figure 7: kinetic energy integrand  $\bar{k}(\bar{r})$

We then use the function **trapz(xL,yL)** (in the file **project2.mac**), to find the “area under the curve”, which represents the value of the integral  $\int_0^R d\bar{r} \bar{k}(\bar{r})$ . The code **trapz(xL,yL)** uses the trapezoidal rule and accommodates the non-uniform grid associated with the output of **dwarf1**.

```
(%i14) area : trapz(rL,kL);
(%o14) 0.096435
```

Using the dimensionless gravitational energy Eqs. (3.7, 3.8), we now plot the absolute value of the dimensionless integrand  $\bar{u}(\bar{r})$  of  $\bar{U}$  and then compute the numerical value of  $|\bar{U}|$

```
(%i15) uL : rL*mL*rhoL;
(%o15) [0,3.3333333E-49,3.2688898E-5,5.0722051E-4,0.0024291,0.0071014,
0.015691,0.028828,0.046346,0.067236,0.089792,0.11191,0.13142,0.14648,
0.15572,0.15848,0.15474,0.14509,0.13057,0.11251,0.092366,0.071579,
0.051472,0.0332,0.017759,0.006113,0.0051998,0.0043399,0.003536,
0.0027914,0.0021102,0.0014978,9.6163089E-4,5.1280807E-4,1.715986E-4]
(%i16) plot2d([discrete,rL,uL],[xlabel,"r"],[ylabel,"u(r)"],
[style,[lines,3]],[y,0,0.25],[gnuplot_preamble,"set grid"])$
(%i17) area : trapz(rL,uL);
(%o17) 0.17767
```

The plot of  $|\bar{u}(\bar{r})|$  is

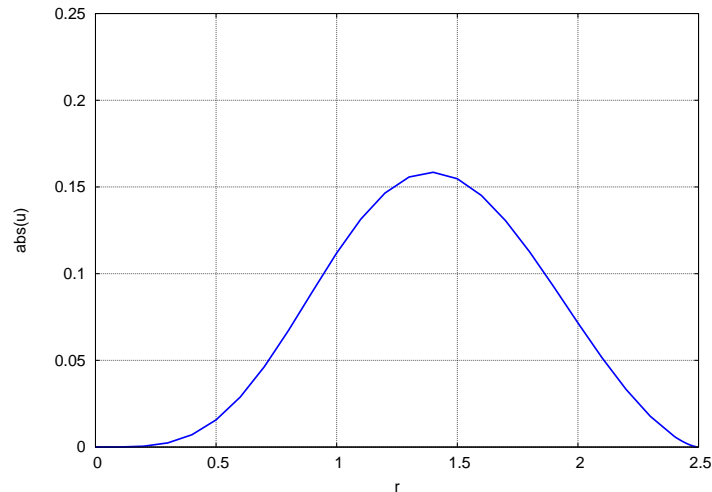


Figure 8:  $|\bar{u}(\bar{r})|$

For this case, the sum  $\bar{K} + \bar{U} = -0.0812$ , a negative number, as is needed for the existence of a bound state.

## 10 Energy Plots Using R

Using the dimensionless kinetic energy Eqs. (3.4, 3.5), we compute values of the kinetic energy at the  $r$  values used in the integration of the differential equations, then find the numerical value of the integral  $\bar{K}$  which gives the internal kinetic energy of the star as a whole, and finally plot the shape of the kinetic energy integrand  $\bar{k}(\bar{r})$ .

We use vector methods in R to generate the vector of needed ordinates from the vector of  $\bar{r}$  values  $\mathbf{rL}$  and the vector of  $\bar{\rho}$ -values  $\mathbf{rhoL}$ .

We use the function `trapz(xL,yL)` (in the file `project2.R`), to find the “area under the curve”, which represents the value of the integral  $\int_0^{\bar{R}} d\bar{r} \bar{k}(\bar{r})$ . The R code `trapz(xL,yL)` uses the trapezoidal rule and accommodates the non-uniform grid associated with the output of `dwarf1`.

```
> source("project2.R")
> soln = dwarf1(1,0.1,0.01,100)
> rL = soln[[1]]
> mL = soln[[2]]
> rhoL = soln[[3]]
> eps = function(rh) {
+   3*(rh^(1/3) * (1+2*rh^(2/3)) * sqrt(1+rh^(2/3)) - log(rh^(1/3)
+     + sqrt(1+rh^(2/3))))/rh }
> epsml = function(z) { eps(z) -1}
> kL = rL^2*rhoL*sapply(rhoL,epsml)
> kL
[1] 0.000000e+00 2.604752e-25 2.568310e-03 9.931753e-03 2.112956e-02
[6] 3.474073e-02 4.911342e-02 6.261336e-02 7.384384e-02 8.180181e-02
[11] 8.595289e-02 8.622710e-02 8.295137e-02 7.674282e-02 6.838762e-02
```

```
[16] 5.872617e-02 4.855862e-02 3.857759e-02 2.932885e-02 2.119674e-02
[21] 1.440851e-02 9.051573e-03 5.097404e-03 2.427364e-03 8.558650e-04
[26] 1.460987e-04 1.117186e-04 8.277500e-05 5.892050e-05 3.979124e-05
[31] 2.500289e-05 1.414458e-05 6.769413e-06 2.377931e-06 3.841892e-07
> trapz(rL,kL)
[1] 0.09643478
> plot(rL,kL,type="l",lwd=3,col="blue",xlab = "r",ylab = "k")
> mygrid()
```

which produces the plot of  $\bar{k}(\bar{r})$  for  $\bar{\rho}_c = 1$ :

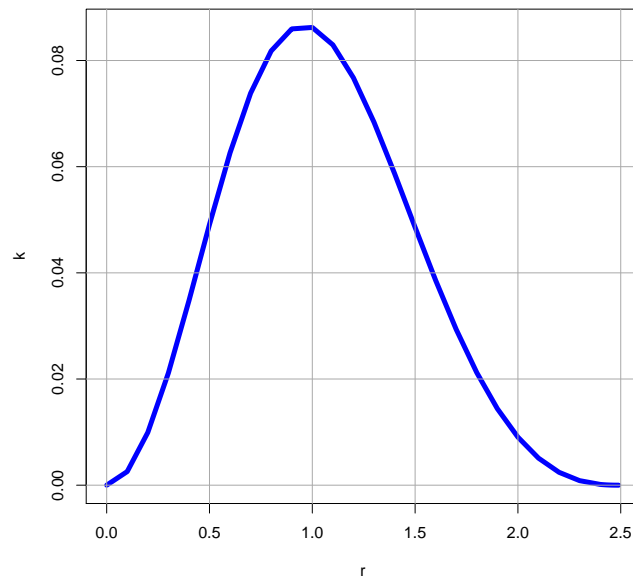


Figure 9: kinetic energy integrand  $\bar{k}(\bar{r})$

Using the dimensionless gravitational energy Eqs. (3.7, 3.8), we again use vector methods in R to find the values of the absolute value of  $\bar{u}(\bar{r})$  at the  $\bar{r}$  values used in the integration of the white dwarf differential equations, use `trapz` to find the “area under the curve”, which is the numerical value of  $|\bar{U}|$ , and then plot the absolute value of the dimensionless integrand  $\bar{u}(\bar{r})$  of  $\bar{U}$ .

```
> uL = rL*mL*rhoL
> uL
 [1] 0.000000e+00 3.333333e-49 3.268890e-05 5.072205e-04 2.429142e-03
 [6] 7.101385e-03 1.569108e-02 2.882771e-02 4.634633e-02 6.723638e-02
[11] 8.979184e-02 1.119057e-01 1.314240e-01 1.464765e-01 1.557222e-01
[16] 1.584790e-01 1.547387e-01 1.450879e-01 1.305683e-01 1.125095e-01
[21] 9.236633e-02 7.157889e-02 5.147179e-02 3.320043e-02 1.775878e-02
[26] 6.112973e-03 5.199767e-03 4.339866e-03 3.535981e-03 2.791400e-03
[31] 2.110234e-03 1.497842e-03 9.616309e-04 5.128081e-04 1.715986e-04
> trapz(rL,uL)
[1] 0.1776717
> plot(rL,uL,type="l",lwd=3,col="blue",xlab = "r",ylab = "abs(u)")
> mygrid()
```

which produces the plot

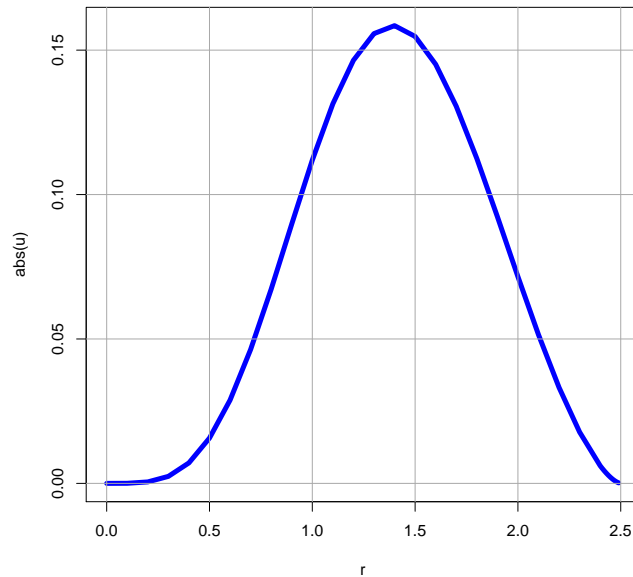


Figure 10:  $|\bar{u}(\bar{r})|$

For this case, the sum  $\bar{K} + \bar{U} = -0.0812$ , a negative number, as is needed for the existence of a bound state.

## 11 Plot of $\bar{R}$ Versus $\bar{M}$ Using Maxima

A Maxima function `dwarf3` (in `project2.mac`) converts a given central density into the list  $[\bar{M}, \bar{R}]$ .

```
/* dwarf3(rhoc, dr, rtol) calls dwarf1 and returns [Mbar, Rbar] */
dwarf3(rhoc, dr, rtol) :=
block([soln,surf,numer:true],
  soln : dwarf1(rhoc, dr, rtol),
  surf : last(soln),
  [surf[2], surf[1]])$
```

A Maxima function `dwarf4` converts a list of central mass densities into a list with elements  $[\bar{M}, \bar{R}]$  and returns that list, as well as printing a table of values.

```
dwarf4() :=
block([rhoL,drv:0.01,rtolv:0.001,mrL:[],rd3,numer:true],
  rhoL : [0.1,0.5,1,5,10,100,1e3,1e4,1e5, 2e5,3e5,4e5],
  print("  rhoc          M          R"),
  for vv in rhoL do (
    rd3 : dwarf3(vv,drv,rtolv),
    mrL : cons(rd3, mrL),
    printf(true,"~& ~8f ~10f ~10f ", vv, rd3[1], rd3[2])),
  reverse(mrL))$
```

```
(%i1) load(project2);
(%o1) "c:/k2/project2.mac"
(%i2) mRL : dwarf4();
      rhoc          M          R
      0.1    0.2798235    3.759
      0.5    0.5498911    2.834
      1.0    0.7070609    2.497
      5.0    1.12278661   1.833
```

```

10.0    1.29799605    1.591
100.0   1.73549443    0.953
1000.0  1.93281009    0.533
10000.0 1.99719275    0.279
100000.0 2.02782531    0.14
200000.0 2.05241163    0.114
300000.0 2.08105322    0.1
400000.0 2.11459555    0.092
(%o2) [[0.27982,3.759],[0.54989,2.834],[0.70706,2.497],[1.122787,1.833],
      [1.297996,1.591],[1.735494,0.953],[1.93281,0.533],[1.997193,0.279],
      [2.027825,0.14],[2.052412,0.114],[2.081053,0.1],[2.114596,0.092]]
(%i3) plot2d([discrete, mRL],[xlabel,"M"],[ylabel,"R"],
      [style,[lines,3]])$

```

which produces the plot

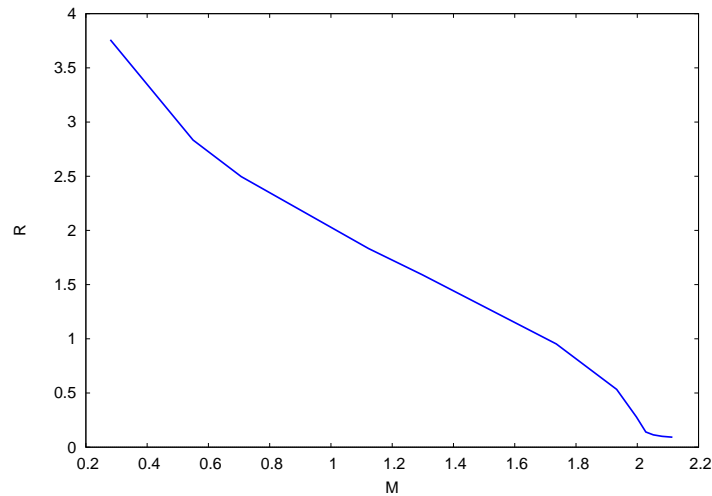


Figure 11:  $\bar{R}$  vs.  $\bar{M}$  for  $\bar{\rho}_c = 1$

## 12 Plot of $\bar{R}$ Versus $\bar{M}$ Using R

A R function `dwarf3` (in `project2.R`) converts a given central density into a vector holding the values  $(\bar{M}, \bar{R})$ .

```

## dwarf3(rhoc, dr, rtol) calls dwarf1 and returns c(Mbar, Rbar)

dwarf3 = function(rhoc, dr, rtol) {
  nmax = 500
  soln = dwarf1(rhoc, dr, rtol, nmax) # list(rL,mL,rhoL)
  surfL = get_last(soln)
  c(surfL[2], surfL[1]) }

## > dwarf3(1,0.1,0.01)
## [1] 0.7066441 2.4900000

```

A R function `dwarf4` converts a list of central mass densities into a list of two vectors, `mv` and `rv`, which hold corresponding values of the scaled masses and radii for the different central mass densities, by calling `dwarf3` with each central density.

```

## dwarf4() calls dwarf3 to accumulate vectors of values of M and R
## returns list(ML, RL)

dwarf4 = function(rho_vals) {
  drv = 0.1
  rtolv = 0.01
  num = length(rho_vals)

```

```

mv = vector(length=num)
rv = vector(length=num)
k = 1
for (rhoc in rho_vals) {
  rd3 = dwarf3(rhoc,drv,rtolv)
  mv[k] = rd3[1]
  rv[k] = rd3[2]
  k = k + 1}
list(mv, rv) }

```

Here we use the R function `dwarf4` to make the plot.

```

> source("project2.R")
> rhocL = c(0.1, 0.5, 1, 5, 10,100,1e3,1e4,1e5, 2e5, 3e5, 4e5)
> mRL = dwarf4(rhocL)
> ML = mRL[[1]]
> f11(ML)
0.2797391 2.096836 12
> RL = mRL[[2]]
> f11(RL)
3.75 0.07 12
> plot( ML, RL, type = "l", lwd = 3, col = "blue", xlab = "M", ylab = "R")
> mygrid()

```

which produces the plot

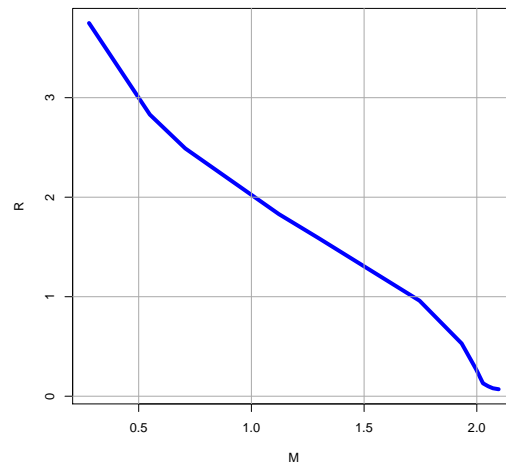


Figure 12:  $\bar{R}$  vs.  $\bar{M}$  for  $\bar{\rho}_c = 1$

We can then make a table of values using `data.frame`.

```

> rho_cases = data.frame(rhoc = rhocL, M = ML, R = RL)
> rho_cases
  rhoc      M      R
1 1e-01 0.2797391 3.75
2 5e-01 0.5496256 2.83
3 1e+00 0.7066441 2.49
4 5e+00 1.1219404 1.83
5 1e+01 1.2970923 1.59
6 1e+02 1.7459976 0.96
7 1e+03 1.9328042 0.53
8 1e+04 1.9970700 0.27
9 1e+05 2.0273174 0.13
10 2e+05 2.0504694 0.10
11 3e+05 2.0713817 0.08
12 4e+05 2.0968360 0.07

```

### 13 Energy Components as a Function of $\bar{\rho}_c$ and $\bar{M}$ Using Maxima

The Maxima function `dwarf2` (in `project2.mac`) takes as input the output of `dwarf1` and calculates the internal (dimensionless) kinetic energy of the star **Ek**, the internal (dimensionless) gravitational energy of the star **Eg**, and the internal (dimensionless) kinetic and rest mass energy of the star **Ek<sub>m</sub>**.

```
/* dwarf2(sL) evaluates the dimensionless kinetic (Ek), gravitational energy (Eg),
   and kinetic plus rest mass energy (Ekm) of the white dwarf using the
   output of dwarf1 as input, and returns the list [Ek, Eg, Ekm] */

dwarf2(sL) :=
block([rL,mL,rhoL,kL,uL,Ek, Eg,Ekm,tempL, numer:true],local(eps,epsm1),
  eps(rh) := (3/8/rh)*(rh^(1/3))*(1+2*rh^(2/3))*sqrt(1+rh^(2/3)) -
            log(rh^(1/3) + sqrt(1+rh^(2/3))),
  epsm1(z) := eps(z) - 1,
  rL : take(sL,1),
  mL : take(sL,2),
  rhoL : take(sL,3),
  kL : rL^2*rhoL*map('epsm1, rhoL),
  Ek : trapz(rL,kL),
  uL : rL*mL*rhoL,
  Eg : -trapz(rL,uL),
  tempL : float(map('eps, rhoL)),
  kmL : rL^2*rhoL*tempL,
  Ekm : trapz(rL,kmL),
  [Ek, Eg , Ekm ])$
```

We define `EL(rhoc)` to use the output of `dwarf1` for a given central mass density as the input to `dwarf2`, returning the list `[Ek, Eg, Ekm]`

```
(%i1) load(project2);
(%o1) "c:/k2/project2.mac"
(%i2) EL(rhvc) := dwarf2(dwarf1(rhvc,0.01,0.001))$
(%i3) EL(1);
(%o3) [0.096598,-0.17803,0.80366]
(%i4) EL(10);
(%o4) [0.62231,-1.008495,1.920303]
(%i5) EL(1e2);
(%o5) [2.435038,-3.355281,4.170529]
(%i6) EL(1e3);
(%o6) [6.974003,-8.383828,8.906817]
(%i7) EL(1e4);
(%o7) [17.09202,-18.80989,19.08944]
(%i8) EL(1e5);
(%o8) [39.16687,-41.08195,41.20077]
(%i9) EL(1e6);
(%o9) [86.5433,-88.49602,88.56048]
```

Evidently the sum (**Ek<sub>m</sub> + Eg**) approaches 0 at high central densities.

To see the behavior of the total energy as a function of the mass of the star, we use `dwarf5`, which returns a list with elements `[M, Et]`.

```
/* dwarf5 calls dwarf1 and dwarf2 to accumulate a list
   with elements [M,Ekm+Eg] */

dwarf5() :=
block([rhoL,drv:0.01,rtolv:0.001,MEL:[],rd1,rd2,
  M,Et, numer:true],
  rhoL : [0.1,0.5,1,5,10,100,1e3,1e4,1e5, 2e5,3e5,4e5],
  print("      rhoc      M      Et "),
  for vv in rhoL do (
    rd1 : dwarf1(vv,drv,rtolv),
    M : last(rd1)[2],
    rd2 : dwarf2(rd1),
```



```

Et : rd2[2] + rd2[3],
MEL : cons([M, Et], MEL),
printf(true,"~& ~8e ~10f ~10f ", vv, M, Et)),
reverse(MEL))$

```

and produces the table

```

(%i10) mel : dwarf5()$
      rhoc      M      Et
1.0E-1  0.2798235  0.2710139
5.0E-1  0.5498911  0.5057441
1.0E+0  0.7070609  0.6256286
5.0E+0  1.12278661  0.8608924
1.0E+1  1.29799605  0.9118089
1.0E+2  1.73549443  0.8152481
1.0E+3  1.93281009  0.5229894
1.0E+4  1.99719275  0.2795461
1.0E+5  2.02782531  0.1188195
2.0E+5  2.05241163  0.0464371
3.0E+5  2.08105322  0.0018255
4.0E+5  2.11459555  -0.045787

```

and the output allows us to plot the total dimensionless scaled energy of the star (including rest mass energy) versus the dimensionless scaled mass of the star.

```

(%i11) mv : take(mel,1)$
(%i12) f11(mv);
(%o12) [0.27982,2.114596,12]
(%i13) Ev : take(mel,2)$
(%i14) f11(Ev);
(%o14) [0.27101,-0.045787,12]
(%i15) plot2d( [discrete, mv, Ev] ,[xlabel,"M"],[ylabel," "],[style,[lines,3]] )$

```

which produces the plot

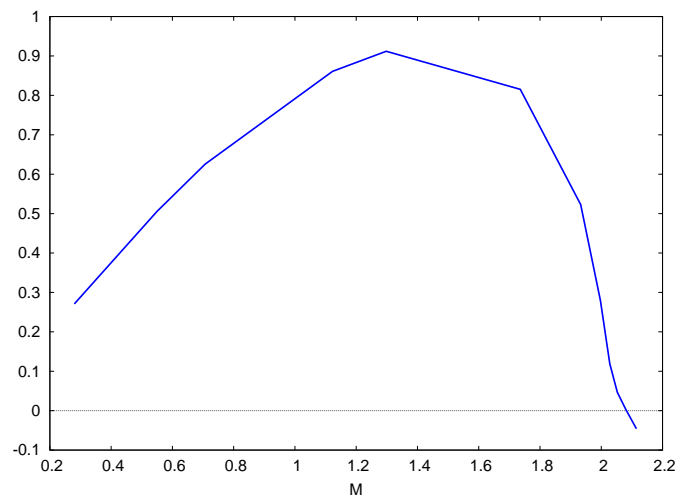


Figure 13: Total Energy  $\bar{E}_t$  vs. Mass  $\bar{M}$

We see that the total scaled energy becomes negative at a critical value of the scaled mass of the star, corresponding to the scaled radius of the star going to zero. Although our model is not detailed enough (especially needed is a better treatment of the physics near the surface of the star), the present results point to the existence of an upper limit to the value of the mass of a white dwarf.

## 14 Comparison of Non-Relativistic Approximation with Relativistic Solutions

In this section we drop the over-bars on the symbols for the scaled dimensionless variables. In the low density non-relativistic approximation,  $x \ll 1$ , and  $\gamma(x) \approx x^2/3 = \rho^{2/3}/3$ , and the pair of ordinary differential equations to be integrated from  $r = 0$  to the value of  $r$  at which  $\rho = 0$  are then:

$$\frac{dm}{dr} = \rho r^2 \quad (14.1)$$

$$\frac{d\rho}{dr} = -\frac{3m(r)\rho^{1/3}}{r^2} \quad (14.2)$$

And for  $r \ll 1$  and  $r \rightarrow 0$ ,

$$m(r) \approx \frac{1}{3} \rho_c r^3 \quad (14.3)$$

$$\rho(r) \approx \rho_c - \frac{1}{2} r^2 \rho_c^{4/3} \quad (14.4)$$

where  $\rho_c = \rho(r = 0)$ .

### 14.1 Plot comparisons

The Maxima code **dwarf6**, following the same logic as **dwarf1**, integrates the non-relativistic case.

```
(%i1) load(project2);
(%o1) "c:/k2/project2.mac"
(%i2) soln1 : dwarf1(1,0.1,0.01)$
(%i3) soln6 : dwarf6(1,0.1,0.01)$
(%i4) rL1 : take(soln1,1)$
(%i5) rL6 : take(soln6,1)$
(%i6) mL1 : take(soln1,2)$
(%i7) mL6 : take(soln6,2)$
(%i8) rhoL1 : take(soln1,3)$
(%i9) rhoL6 : take(soln6,3)$
(%i10) plot2d( [ [discrete,rL1,mL1],[discrete,rL6,mL6] ] ,[style,[lines,3]],
  [xlabel,"r"], [ylabel,"m"],[legend,"Relativistic","NR"],
  [gnuplot_preamble,"set key bottom right;set grid"])$
```

which produces the comparisons of  $\bar{m}(\bar{r})$  :

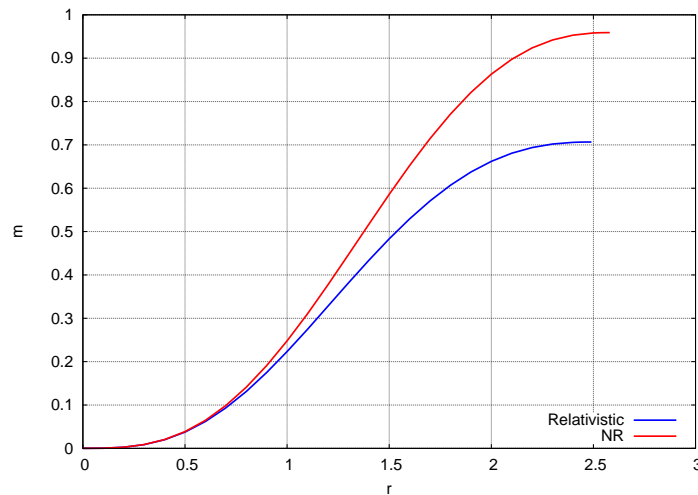


Figure 14:  $\bar{m}(\bar{r})$  for  $\bar{\rho}_c = 1$

and the comparisons for the mass density profile,

```
(%i11) plot2d( [ [discrete,rL1,rhoL1],[discrete,rL6,rhoL6] ] ,[style,[lines,3]],
  [xlabel,"r"], [ylabel,"rho"],[legend,"Relativistic","NR"],
  [gnuplot_preamble,"set grid"])$
```

which produces the comparisons of  $\bar{\rho}(\bar{r})$  :

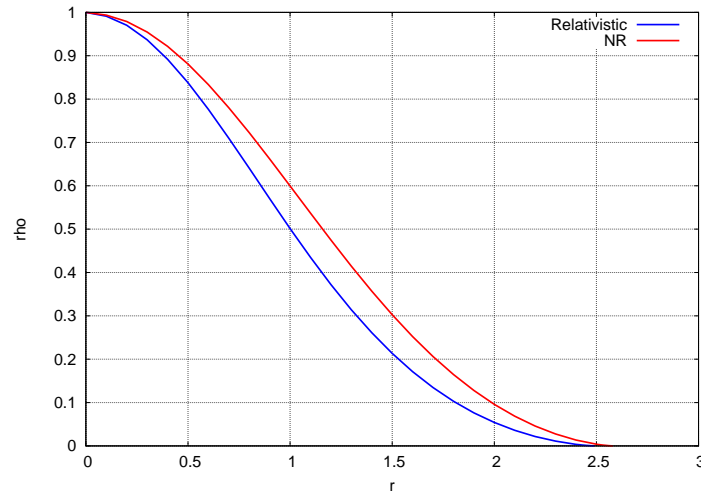


Figure 15:  $\bar{\rho}(\bar{r})$  for  $\bar{\rho}_c = 1$

## 15 Suggested Further Explorations

1. Compare the ultra-relativistic case  $x \gg 1$  with the cases we have considered.
2. Write R code for functions which will allow exploration of the variations of the energy components with changing central density and stellar mass, and allow a plot of the total star energy (including rest mass energy) versus the star mass.
3. Calculate the c.g.s. values of the mass and radius of a white dwarf star, assuming iron nuclei, for scaled central densities in the range we have used, and express the c.g.s. central densities in terms of the central density of the Sun, and express the masses in terms of the mass of the Sun, and express the radii in terms of the radius of the Sun. Write some code to do these calculations and make a table.
4. Find a theoretical discussion of the physics of the surface region of a white dwarf star, and incorporate that physics into our model.

## 16 Astrophysics References

1. **An Invitation to Astrophysics**, Thanu Padmanabhan (World Scientific, 2006), Chapter 3, Matter, Chapter 4, Stars and Stellar Evolution.
2. **Theoretical Astrophysics**, Volume 1, Astrophysical Processes, Thanu Padmanabhan (Cambridge Univ. Press, 2000), Chapter 1, Order of Magnitude Astrophysics, Chapter 5, Statistical Mechanics.
3. **Theoretical Astrophysics**, Volume 2, Stars and Stellar Systems, Thanu Padmanabhan (Cambridge Univ. Press, 2001), Chapter 5, White Dwarfs, Neutron Stars, and Black Holes.
4. **The Early Universe**, Edward W. Kolb and Michael S. Turner (Westview Press, 1994), Chapter 3, Standard Cosmology, Sec. 3, Equilibrium Thermodynamics.
5. **An Introduction to the Study of Stellar Structure**, S. Chandrasekhar, (Dover, 1958), Chapter 10, The Quantum Statistics, Chapter 11, Degenerate Stellar Configurations and the Theory of White Dwarfs.