University of Oslo

AST3310

Term Project 2

Modelling a Stellar Core

Written by:
Bendik Samseth

April 3, 2016

Contents

Abstract

In this report, a simple numerical model of a stellar core is developed. The governing equations are presented and discussed, and a numerical method is deployed in order to solve them. The Forward Euler method is chosen for simplicity, and the effects of using different step sizes are reviewed. The mathematics of using a dynamic step size is explained, and this is used in the final version of the model. A comprehensive analysis is then applied to the effects of varying the boundary conditions. This is done in order to find a set of initial parameters that give the most realistic model of a star. The result is a model where the radius, mass and luminosity all go to zero (within a small percentage), while still having a non-negligible core size of about 20% of the total star. All source material related to this report, which is cited at the relevant points, can be found at the projects GitHub repository [1].

1 Introduction

When trying to understand the mechanics of a star, the most logical place to start is with what we can observe. Let's take a look at the properties of a star that we can measure with relative ease.

Firstly, we can look at how much radiation energy hits one particular area, i.e. a solar panel on earth. By assuming that the star radiates equally in all directions we can calculate the total amount of energy sent out by the star, the so-called luminosity (denoted as L).

In addition, we can find the surface temperature (denoted T) by looking at the color of the light emitted from the star and using Wien's displacements law.

Lastly, we might be able to look at the orbits of all the planets and do some number crunching and figure out the total mass of the star (denoted M). This last part may be easier said than done, for instance in the case of a far away solar system where we can't even see the planets orbiting, just their minuscule effect on the stars orbit. However, for the purposes of this report, let's assume that we are able to do so.

But that's about it for what we can figure out by just looking at the outside effects. There are many other properties we would like to know about, e.g. the radius (R), density (ρ) and pressure (P) of the star. In particular, we would like to have all of these properties as a function of the radius, so that we can get an idea of what the cross section of the star looks like. In order to achieve this, we need to deploy many different branches of physics, including thermodynamics, hydrostatics and nuclear physics. And of course, a good number of assumptions to simplify things for us.

2 The Governing Equations

In this project, we will only be looking at the radiative core of the star. This means that we will not look at the outer convection layer, and assume that all energy transport is done through photon radiation. For reference, this corresponds to the inner $\sim 70\,\%$ of the sun. Furthermore, we will not consider time evolution; we look at one particular moment in time.

The following four differential equations govern the internal structure of radiative core of our star[1]:

$$\frac{\partial r}{\partial m} = \frac{1}{4\pi r^2 \rho} \tag{1}$$

$$\frac{\partial P}{\partial m} = -\frac{Gm}{4\pi r^4} \tag{2}$$

$$\frac{\partial L}{\partial m} = \epsilon \tag{3}$$

$$\frac{\partial T}{\partial m} = -\frac{3\kappa L}{256\pi^2 \sigma r^4 T^3} \tag{4}$$

The first thing to notice about these equations are that they are differentials with respect to mass, and not radius. Remember that what we want in the end is all the properties of the star as a function of radius, so it would make sense to write the equations with respect to r instead[2]. It turns out that it is more numerically stable to use dm compared to dr, so we are going to treat the radius as $r = r(m)$. When we plot the data later, however, we will return to using r as the variable. Eq. (1) is simply stating the relation between taking an infinitesimal step in r, compared to an infinitesimal step in m.

Eq. (2) is the assumption of hydrostatic equilibrium, and states that if the gas in the star is to be at rest, then the outward pressure must exactly balance the force of gravity acting on the gas.

The ϵ in Eq. (3) represents the amount of energy produced by nuclear fusion per time and mass. This quantity will be treated more in section 2.2.

Eq. (4) is the temperature gradient need to transport all the produced energy out of the star. I should be noted that this is only the correct temperature gradient when all the energy is transported by radiation.

[1] Derivations of the equations are not given, as they are shown in the lecture notes [2]. A qualitative description of their meaning is given instead.

[2] I will use lower case r and m when referring to the radius and mass as variables, and upper case R and M for the total radius and mass.

2.1 Equation of State

In the equations (1-4) we use T, ρ and P. If we take a look at how many unknowns we have, we find that we need one more equation. To this end, we are going to assume that we can treat the gas in the star as an ideal gas, thus getting an equation for the gas pressure through the ideal gas law:

$$P_g V = N k_B T$$
$$\Rightarrow P_g = \frac{N}{V} k_B T = \frac{\rho}{\mu\, m_u}, \tag{5}$$

where P_g is the gas pressure, V is the volume, N is the number of particles and T is the temperature of the gas. The quantity $\mu\, m_u$ is the average mass of all particles in the gas. m_u is just a constant giving the average mass of a nucleon. μ is slightly more complicated, and represents the mass of the average particle in units of m_u (μ is unit less). It can be calculated a couple of ways. From the above equation we have

$$\mu = \frac{\rho}{m_u} \frac{V}{N} = \frac{\rho}{m_u n_{\text{tot}}} \tag{6}$$

where n_{tot} is the total number density of particles in the gas. n_{tot} can be taken as the sum of n_i for all particles i,

$$n_{\text{tot}} = \sum_i n_i = n_e + \sum_i \frac{X_i \rho}{C_i m_u} \tag{7}$$

where X_i and C_i is the mass fraction, and number of core elements of particle i, respectively. For instance, ${}^4_2\text{He}$ gives $n_{{}^4_2\text{He}} = Y\rho/4m_u$, when Y is the mass fraction of Helium with four nucleons. The number density of the electron, n_e is written separate and should only consider the number of *free* electrons in the gas. In the case of only fully ionized ${}^4_2\text{He}$, $n_e = 2n_{{}^4_2\text{He}}$ because each ionized Helium core gives two electrons. A similar consideration should be taken for each species present in the gas.

Calculating n_{tot} is not necessarily so straight forward, but if we combine all the steps we have to take to compute μ, we can end up with a convenient formula:

$$\mu = \frac{1}{\sum p_i X_i / C_i}. \tag{8}$$

where p_i is the number of particles provided per nucleus (i.e. the core plus the number of ionized electrons).

This is the method used in the code, as a framework for summing over all particles is central to the structure of the code base.

Let's now return to the original problem of finding an extra equation. Eq. (5) gives us an additional equation, but it also introduces a new parameter, P_g. In addition to the gas pressure, we have radiation pressure P_{rad}, so that the total pressure is $P = P_g + P_{\text{rad}}$. Luckily, the expression for radiation pressure is quite simple, and depends only on temperature,

$$P_{\text{rad}} = \frac{a}{3} T^4 \tag{9}$$

where $a = 4\sigma/c$ is a constant. This gives us finally the additional equation of state we need;

$$P = \frac{\rho}{\mu\, m_u} k_B T + \frac{a}{3} T^4 \tag{10}$$
$$\Leftrightarrow \rho = (P - \frac{a}{3} T^4) \frac{\mu\, m_u}{k_B T}. \tag{11}$$

I have listed the equation with respect to density as well; this relation will be used to find P given ρ and T, and vice versa.

2.2 Energy Production

In Eq. (3), the quantity ϵ represents the amount of energy produced per time and mass. We derive the value of ϵ by looking at the reactions that produce energy inside a star. It is given by

$$\epsilon = \sum_{ij} r_{ij} Q_{ij}, \tag{12}$$

where r_{ij} is the reaction rate for particles i and j (per time and mass), and Q_{ij} is the energy produced for each such reaction. The exact values of Q_{ij} can be found in the lecture notes [2], and also listed in the source code [1, `reaction_energies.h`]. More interesting is the rates, which are given as

$$r_{ij} = \frac{n_i n_j}{\rho(1+\delta_{ij})}\lambda_{ij} \tag{13}$$

where n_i is the number density of particle i, and δ_{ij} is the Kronecker delta. λ_{ij} is a complicated function of temperature, and we will use tabulated expressions for these functions, found once again in the lecture notes [2, Table 3.1].

There are a few different ways that fusion can happen, depending on the conditions of the star. In our case we will simplify things a bit and only consider the PPI and PPII chains (Proton-Proton based fusion).

One important thing to mention when calculating ϵ is that no reaction should happen more often (have larger value for r_{ij}) than the reaction(s) that produced the reactant(s) of the first reaction. For instance, in the PPII we have the two steps

$$^3_2\text{He} + ^4_2\text{He} \rightarrow ^7_4\text{Be} \tag{14}$$
$$^7_4\text{Be} + e^- \rightarrow ^7_3\text{Li} + \nu_e. \tag{15}$$

Here, the second step can happen no more often than the first step; if this was not the case, the Beryllium would quickly disappear and we end up using Beryllium that does not exist. A similar considerations should be made to all dependencies of each step in the cycles.

The last assumption we add is that all Deuterium produced by Proton-Proton fusion, immediately fuses to $^3_2$He, thus effectively merging the two first steps in the PP chains into one reaction.

3 Assumptions

Up until this point, we've made several assumptions. In order to remember the limitations of our results, I have summarized them in a list:

- No time evolution, we only look at one moment in time.
- The gas of the star behaves as an ideal gas.
- All energy transport is done through radiation.
- Energy production:
  - Uniform and constant mass fractions. We use fixed values for all the mass fractions, independent of radius, and assume that they do not change.
  - Metals are not ionized at all, and all non-metals are fully ionized.
  - Only the PPI and PPII reaction chains produce energy.
  - Deuterium is produced and consumed at the same time, combining the first two steps of the PP chains into a single step.

4 Solving the Equations

At this point we are ready to start solving the governing equations. For simplicity, I have chosen to deploy the standard Forward Euler method. One should then note that the simplicity comes at the cost of an global error proportional to the step size, dm. We will try to be smart about this step size in order to minimize the effects of this.

As a reminder, Forward Euler solves a differential equation $du/dx = f(u,x)$ by the explicit scheme:

$$u_{i+i} = u_i + du = u_i + f(u_i, x_i)\,dx.$$

In our case, with four differential equations, this corresponds to the following:

$$\begin{aligned}
r_{i+1} &= r_i + \frac{1}{4\pi r_i^2 \rho_i}\,dm \\
P_{i+1} &= P_i - \frac{Gm_i}{4\pi r_i^4}\,dm \\
L_{i+1} &= L_i + \epsilon\,dm \\
T_{i+1} &= T_i - \frac{3\kappa L_i}{256\pi^2\sigma r_i^4 T_i^3}\,dm \\
m_{i+1} &= m_i + dm
\end{aligned}$$

I added the updating of the mass $m(r_i) = m_i$ for clarity.

4.1 Choosing a Suitable Step Size

Whenever a numerical solution to a differential equation is attempted, one has to think about what step size is suitable. In a perfect world, we would like to set it to an infinitesimal number, but the constraints of reality forces us to make a compromise between accuracy and speed. We need to find a value for dm that gives an accurate enough result, but does so in a reasonable time.

In order to evaluate the effect of different step sizes, figure 1 shows $r(m)$ calculated for a number of different values of dm[3]. From the figure we can see a large variation in $r(m)$ for the lower values of dm. The solution seems to converge when $dm \geq M_0 \times 10^{-3}$, but even there it is not perfectly aligned. It turns out that the code runs very fast anyway (in large part because it is written in C++), so it is no problem to crank the step size down to $dm = M_0 \times 10^{-5}$, and still finish in under a second.

This is quite neat, but we could still try to be a bit smarter about this than just using brute force. The downside to setting a fixed value for dm is that we might spend a lot of computational time on a part of our function(s) that is very stable, and on the other side, we might not have enough precision for where the function(s) is very steep. The alternative is *dynamic step size* (DSS), which involves defining the step size in such a way that the value of the function(s) does not change by more than a fixed percentage. In the case of a diff.eq. $du/dx = f(u, x)$, we define the step size in the following way:

$$\frac{du}{u} \leq p \Leftrightarrow \frac{f\,dx}{u} \leq p \tag{16}$$

$$\Rightarrow dx = \frac{pu}{f}, \tag{17}$$

where p is the fraction that the solution is allowed to change by after one step. In our case we have more than one equation, so we impose a similar requirement to all equations and then simply choose the smallest required step size.

The only thing we then have to define is what the value of p should be. As previously stated, the code is quite fast so we can afford to be quite strict. The code used in all subsequent figures is set to allow a maximum change of 5‰. Figure 2 shows the same plot as figure 1, but this time including dynamic step size as well. The figure shows that DSS gives good results.

5 Changing the Parameters

In this numerical model, we have the option to change the initial parameters to our liking. In particular, we would like to find the set of initial parameters that give the most realistic model of an actual star. To this end we are now going to look at how the solution is affected by changing R_0, M_0, P_0 etc. (In the following, $u_{0,i}$ will denote the default initial condition of quantity u, as defined in the exercise text. The defaults correspond to the values for the Sun at the bottom of the convection layer.)

5.1 Changing R_0

In order to see the effects of changing the initial radius, figure 3 has been produced. From the figure we can conclude with the following for increasing initial radius:

- A change in m corresponds to a larger fractional step in r, meaning that the mass falls of earlier compared to the maximum radius[4].
- P and ρ becomes *much* smaller[5].
- L goes to zero closer to the center.
- T becomes smaller.

[3] The shape and values of $r(m)$ in figure 1 are not important. The graph is only meant to illustrate the different step sizes. Analysis of the shapes and input parameters will be discussed in detail later.

[4] Note that many of these observations relate to changes relative to the initial value. Therefore, saying that the mass falls of earlier doesn't mean that the mass is less than it would be at the same *absolute* distance in a smaller star. The same thing is true for many of the other effects listed.

[5] It turns out that P and ρ behave very much the same in all of the plots. This is due to the linear connection between them shown in Eq. (10)

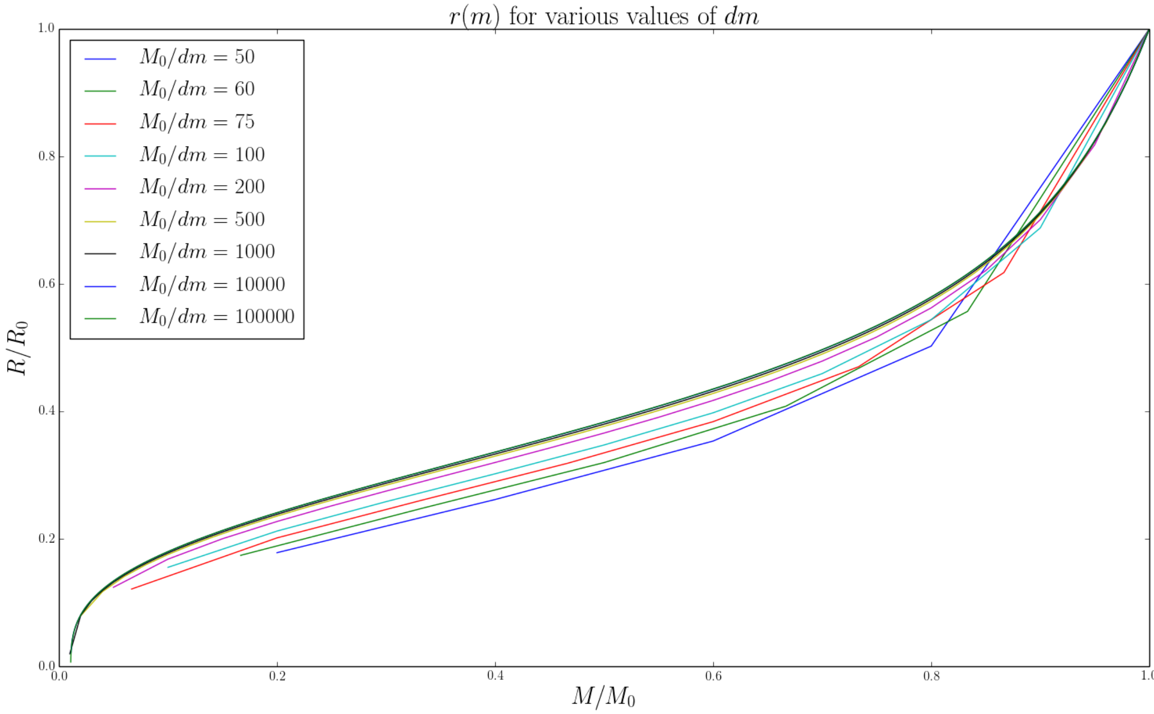

Figure 1: Plots of $r(m)$ using a range of different values for dm. We see that we need quite a large number of steps in order for the solution to converge.

These effects are all reasonable given an increase in the size of the star. We can compare this to what we know happens to a star towards the end of its life cycle, where it becomes a red giant. In that case, the radius increases, causing the surface temperature to drop (thus glowing more red). The core has no reason to expand or contract, so it becomes a smaller fraction of the total radius. Finally, with a constant total mass, it seems reasonable that the density would drop significantly, along with the pressure (all in accordance with Eq. (10)).

5.2 Changing T_0

Figure 4 shows a similar comparison for a change in the initial temperature T_0. We conclude that for increasing initial temperature we have that:

- L falls of more suddenly.
- m falls off more towards the center.
- T is smaller relative to T_0, which hints at a more or less unchanged temperature graph.
- ρ is smaller further out, but bigger in the center.

All these changes can be explained by the star becoming more centrally dense (as seen in the plot for ρ). This would cause m to fall of more in the center. It then seems reasonable that fusion would start more sudden, compared to a star where the center becomes only gradually dense enough to start fusion.

5.3 Changing ρ_0

Figure 5 shows what happens to the solutions when we vary the initial density. We conclude that for increasing initial density we have that:

- m falls off more rapidly

The absolute main thing we gather from this figure is that the mass drops of more rapidly, which of course makes sense given a larger density further out in the star.

The other graphs are quite hard to extract anything else from given that the solution becomes unphysical very fast for large ρ_0's. I will therefore not give much time to interpret the

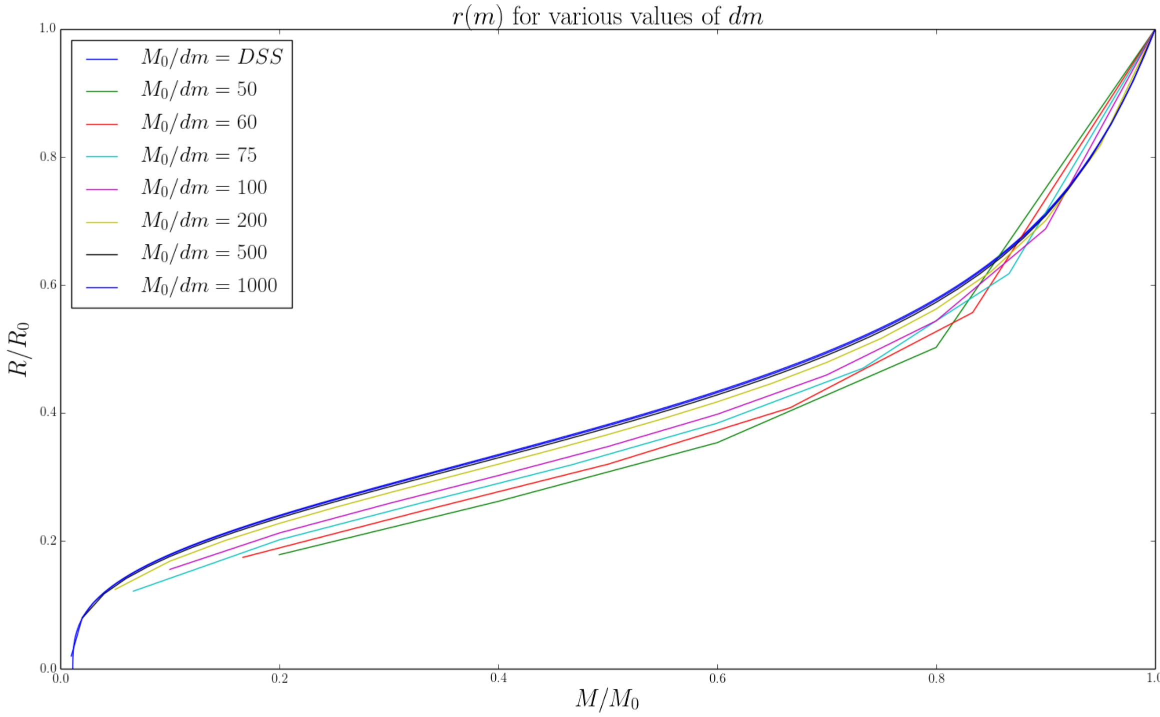

Figure 2: Plots of $r(m)$ using a range of different values for dm, including the solution obtained when using dynamic step size.

result for i.e. P, because they might as well be caused by very strange behaviours of other parameters. However, it seems like the other parameters remain fairly constant, if we take into account that ρ (and P) is scaled according to its initial value.

5.4 Changing P_0

Due to the connection between P, ρ and T shown in Eq. (10), the effects of changing P_0 is included in the above sections. In fact, the code sets the initial value for P based on the values for ρ and T. Producing a similar plot for variations in P_0 would be equivalent of changing ρ_0 or T_0, or some arbitrary combination of the two according to Eq. (10).

5.5 Changing M_0

Figure 6 shows the solutions for a range of total masses. For increasing total mass we conclude that:

- Both ρ and P becomes *much* bigger in the outer parts, but get caught up by the smaller M_0's towards the core
- L starts to decline earlier (core becomes bigger), but it's not clear how increasing M_0 effects where $L \to 0$
- Low M_0's causes m to fall of faster in the beginning and slower towards the center, and the opposite is true for higher M_0's

When we make the mass larger, it seems that we obtain a more uniformly dense star. We can see that the density goes up fast in the outer parts of the star, but then plateaus and remains stable for large parts of the star. This makes for a much less well defined core, as we can see from the luminosity plot (L declines earlier, but less suddenly).

6 Finding the Most Realistic Model

Our end goal with this project was to determine how the parameters of the star evolves as a function of radius, given a set of initial conditions. As seen, what we chose to set as these conditions can have a great impact on the final solution. In particular, many of the condi-

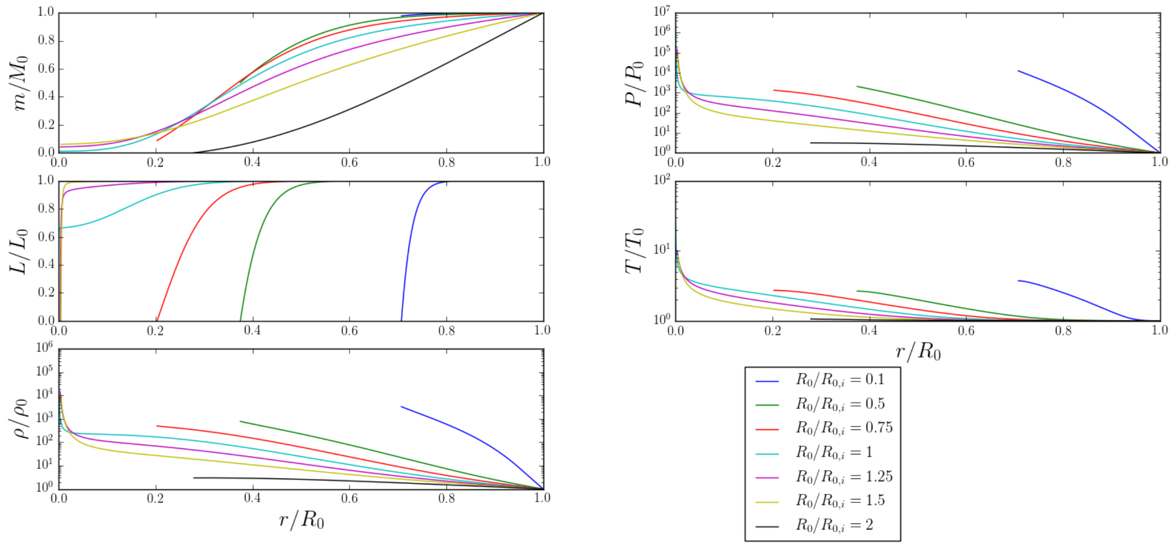

Figure 3: Plots for all parameters of the star showing the effects of changing the initial radius R_0.

tions we have tried yielded very unphysical results. There are certain things we know need to happen at the center of the star, namely that the mass and luminosity both need to be zero (for $r \to 0$). We can therefore set out to find a combination of parameters that best fulfills this requirement, using what we learned from the previous section.

By using the default parameters (which correspond to the values at the bottom of the convection layer in the Sun), we get the results in figure 7. As we can see, with these initial conditions the mass goes to zero way to fast, in comparison with the luminosity and radius. Because of this, the graphs don't seem to look the way we would expect them to. We do however, see that the mass and luminosity goes down with radius, while density, pressure and temperature goes up towards the center. These are all characteristics that we expect from the governing equations.

These parameters obviously don't fit the model very well[6], so we would like to find the parameters that represent the most realistic model of a star. First of we see that the mass needs to fall off slower than it does in figure 7. The simplest way to do this, according the the results from section 5, is to reduce the initial density. I actually need to reduce this quite a bit in order for the other parameters to have the time to go to zero. Then it's a matter of getting r, m and L synced up as much as possible. The one additional criteria that we have is that the core should not be to small, and we would like it to be at least $\sim 10\%$ of the total radius.

Through a large amount of trial and error, both manually and programmatically varying the inputs, I have landed on the following values:

$$\begin{aligned} R_0/R_{0,i} &= 1.3, \quad M_0/M_{0,i} = 0.95 \\ \rho_0/\rho_{0,i} &= 0.01, \quad T_0/T_{0,i} = 0.8 \end{aligned} \tag{18}$$

Once again, $u_{0,i}$ refers to the value of quantity u at the bottom of the convection zone in the Sun.

It must be said that these values have not been found in a very analytical matter, and so they are not the "perfect" values. It is also quite likely that there exists many combinations of parameters that all yield realis-

[6] One might think that parameters taken from the Sun should fit quite well, because after all they are real values for a real star. However, we must to remember that we have made many assumptions and simplifications, and so it would actually be quite remarkable if they were to fit. We expect our model to fit a star that behaves in the way we have assumed, and the differences we find to our Sun are related to the missing features (and/or lack of understanding) of the Sun.

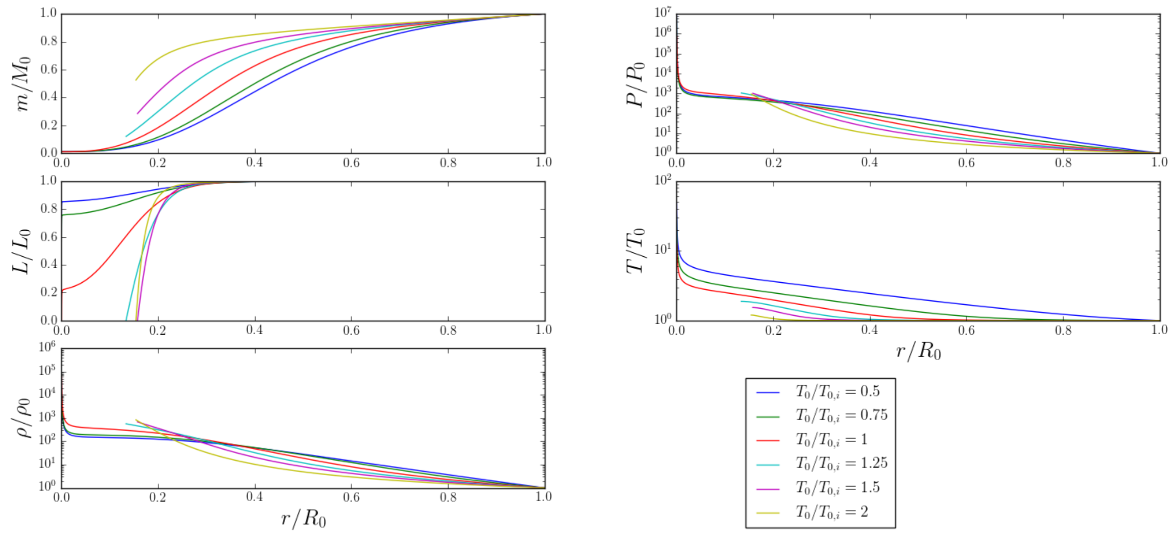

Figure 4: Plots for all parameters of the star showing the effects of changing the initial temperature T_0.

tic models. As seen in the previous sections, some parameters causes i.e. the core to become bigger, while others make it smaller, and so some combination of changes could yield similar results. But, I have not found a way to do this more rigorously than by trial and error, so a single, approximate solution must be sufficient.

With that said, the inputs produce the plots in figure 8. We can see the following characteristics from the figure:

- L is constant at max for $r > 0.2 \cdot R_0$, and goes rapidly to zero for $r < 0.2 \cdot R_0$, meaning that we have a core reaching out to about 20% of R_0.
- Both ρ, P, T and ϵ increases significantly as we move towards the center.
- The mass changes very little in the beginning, due to the low density, and falls of much faster towards the core.

7 Remarks on the Development of the Model

In large part the project has been enjoyable to work on, and I feel quite satisfied with how it has turned out. I have also taken this project as an opportunity to improve my knowledge of the C++ language, and to this end I have at certain points done what might be considered unnecessary work. By this I mean using GitHub and related services (Travis CI, Code coverage reports, README file for the landing page etc.), using the documentation tool Doxygen and setting up (unit) tests with Google Test.

Much of the extra work related to this project arose when trying to debug various problems. Things that have caused more than the normal amount of headache are:

- Implementing bilinear interpolation of the opacity table.
- Getting the energy production correct.
- Converting units from SI→CGS and vice versa (my self-esteem as a physics student took a big hit after messing this up more than I care to say).
- Reproducing the results for the default parameters.
- Many other minor things...

All of this accumulated in a suite of ~ 20 (unit) tests [1, tests/]. In addition, after several days trying to reproduce the plots in the exercise, I rewrote the entire code into Python,

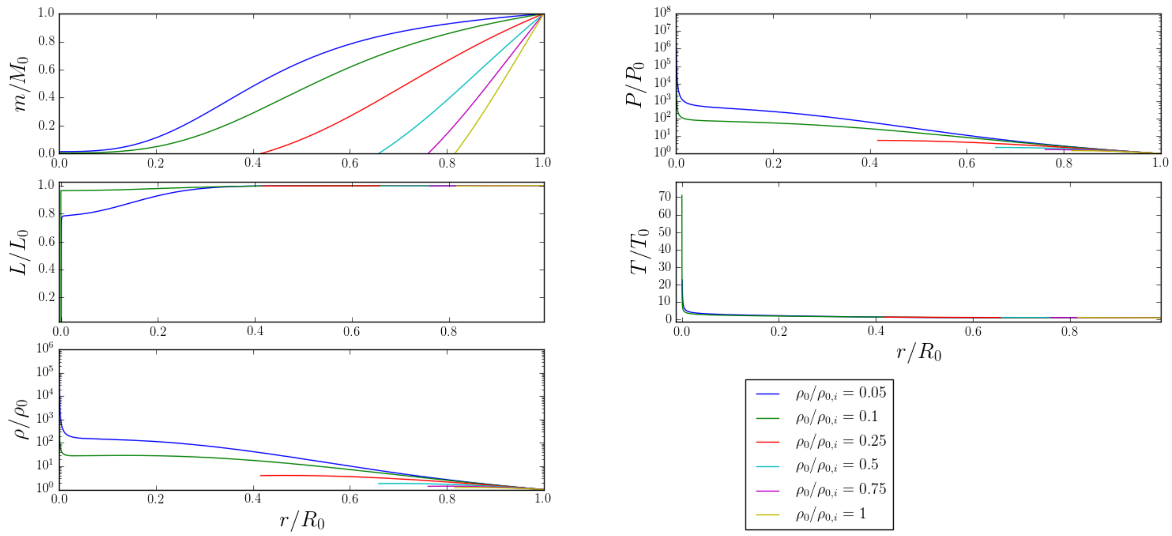

Figure 5: Plots for all parameters of the star showing the effects of changing the initial density ρ_0.

in the hopes of finding the reasons why I couldn't. To my despair, this yielded the exact same result, so I was very much relived when the plots in the exercise was discovered to be incorrect and updated with results that I was able to reproduce.

References

[1] B. Samseth. GitHub repository. URL: `https://github.com/bsamseth/ast3310`.

[2] B. V. Gudiksen. *AST3310: Astrophysical plasma and stellar interiors*. 2016. URL: `http://www.uio.no/studier/emner/matnat/astro/AST3310/v16/noter/appd29-03-16.pdf` (visited on 03/31/2016).

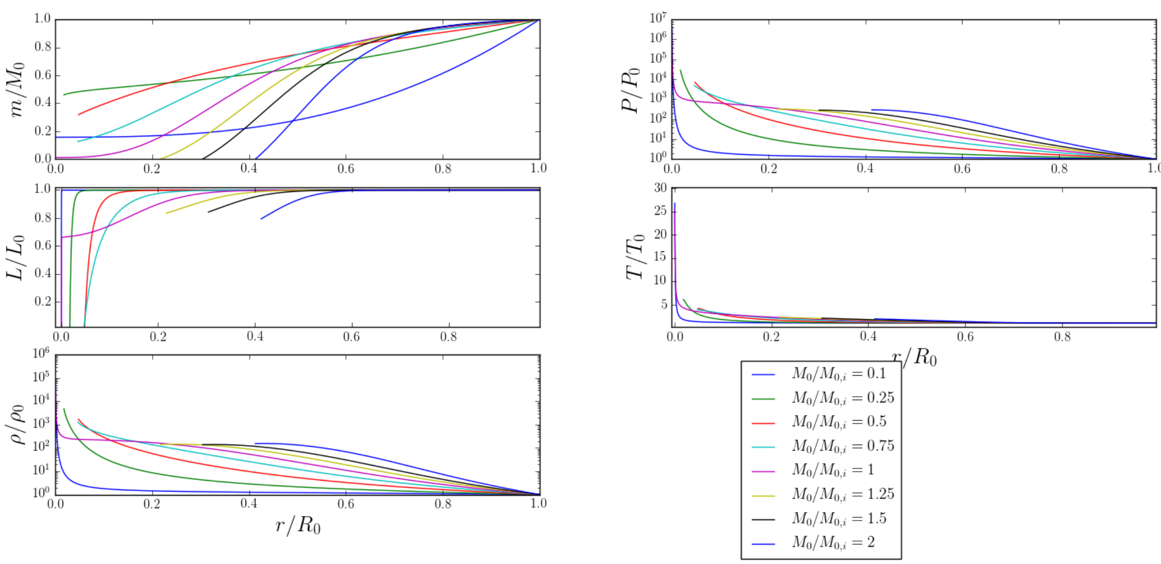

Figure 6: Plots for all parameters of the star showing the effects of changing the total mass M_0.

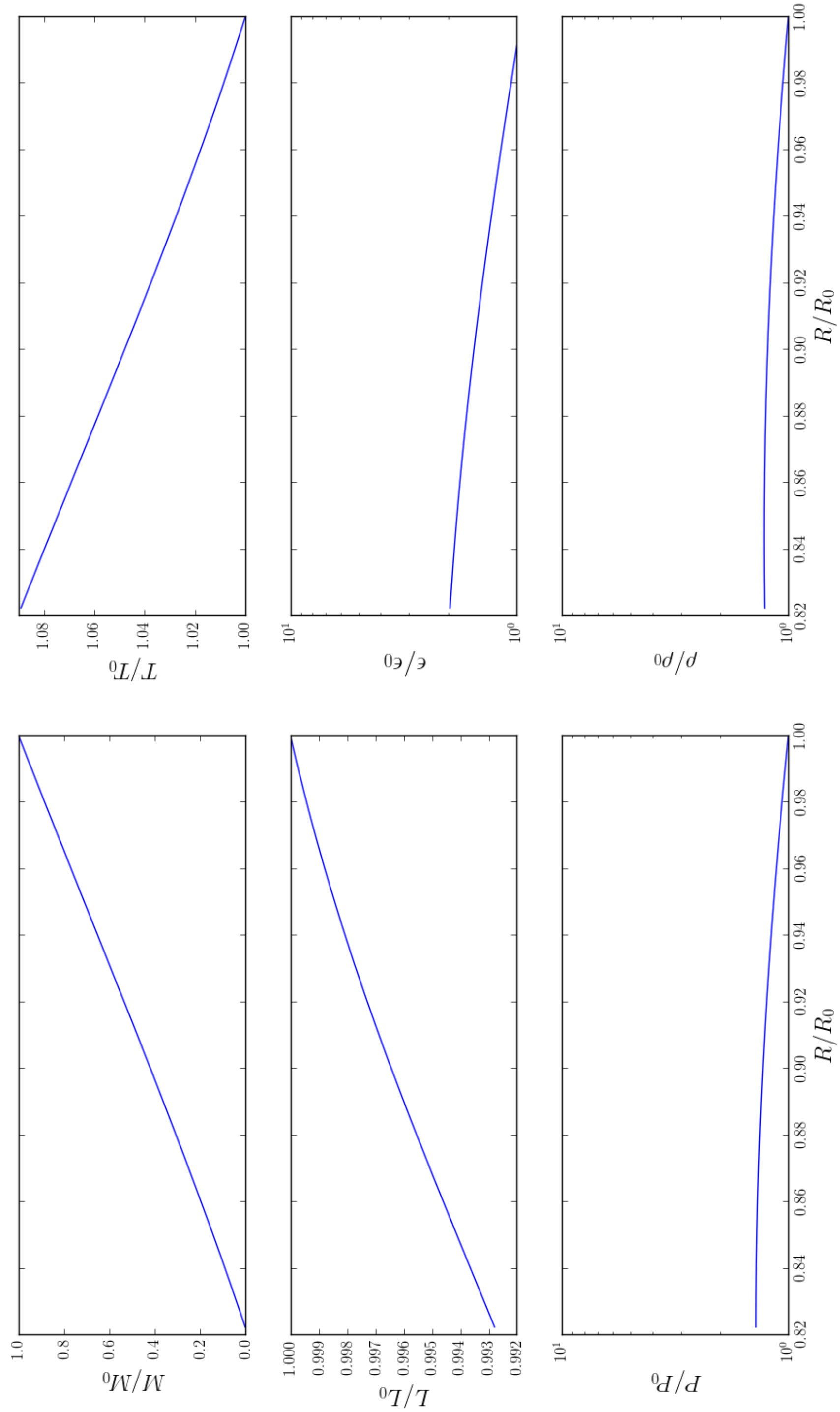

Figure 7: The parameters of the star, all as a function of radius, using the default parameters (corresponding to the bottom of the convection layer in the Sun).

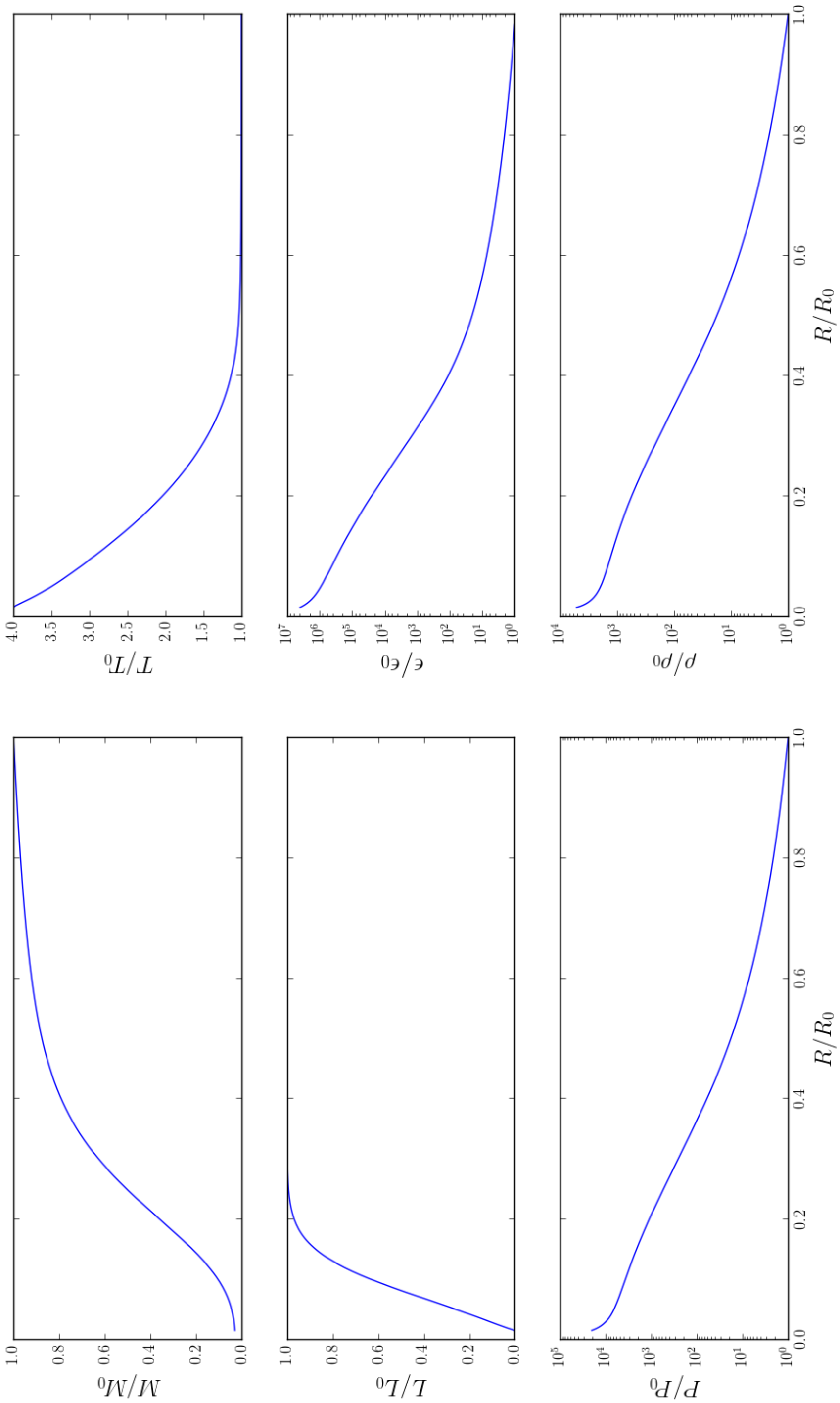

Figure 8: The parameters of the star, all as a function of radius, using the parameters found to give the best result.