Draft version May 15, 2023
Typeset using LaTeX **twocolumn** style in AASTeX631

A Simple Numerical Solution to the Stellar Structure Equations for a Zero-Age Main Sequence Star

Patrick M. McCreery[1]

[1] *Johns Hopkins University*
3400 N. Charles Street
Baltimore, MD 21218, USA

ABSTRACT

In this project, we present a simple Zero-Age Main Sequence (ZAMS) stellar model based on fundamental physical principles, utilizing the stellar structure equations to capture the essential physics of ZAMS stars while avoiding unnecessary complexities. We compare our model to Modules for Experiments in Stellar Astrophysics's (MESA's) predictions, demonstrating our model's accuracy in reproducing key stellar properties. This project highlights the key aspects of more accurate stellar models, as well as the improvements and shortfalls of simple models.

1. INTRODUCTION

Understanding the structure and behavior of stars is a fundamental topic in astrophysics, with implications ranging from the evolution of the Universe, and all of the visible matter within it, to the search for habitable exoplanets. Due to the difficulties in studying stellar interiors (distances, opacities, etc.), the use of mathematical models that describe their internal structure and physical properties are critical to understand their interiors. At the most basic level, these models use the stellar structure equations, providing a mathematical framework for describing the equilibrium and energy transport mechanisms within stars.

In this project, we focus on creating a ZAMS stellar model using the stellar structure equations. A ZAMS star is an early stage of a star's evolution when it has just begun its main sequence phase, characterized by the primary energy generation source being nuclear fusion (compared to gravitational contraction, for example). Studying ZAMS stars provides insights into the initial properties and evolution of stars, serving as a reference point for understanding the physical processes that govern stellar evolution.

We will provide an overview of the key equations involved and the physical principles they represent. We will also discuss the primary numerical method used for solving this coupled system of equations – the shooting method. At the end of this report, we compare specific properties to MESA's predictions and provide a table of interior structure. Additionally, we will explore the challenges and limitations of our specific stellar model, such as the uncertainties in input physics, boundary conditions, and other assumptions.

2. METHODS

The stellar structure equations are a set of differential equations that describe the physical properties of stars, including their temperature, pressure, density, and energy production mechanisms. These four nonlinear, coupled differential equations are based on hydrostatic equilibrium, conservation laws, and energy transport through radiation and convection. By solving these equations, we can create models that demonstrate the behavior of stars for different cases, such as different masses and compositions.

2.1. *Assumptions*

Before getting into the mathematical methods, we begin by outlining the assumptions we make in our model:

- There are only fusion energy sources – ignore other energy sources.
- The ideal gas law applies everywhere.
- Only gas and radiation pressure are present.
- We neglect rotation.
- Assume full ionization of the star.

Now we have a discussion of the fundamental equations we use.

2.2. *Stellar Structure Equations*

The stellar structure equations are four, coupled differential equations that model a star using energy generation, hydrostatic equilibrium, mass conservation, and energy transport using the parameters $\mathcal{L}$ for luminosity, P for pressure, r for distance, and T for temperature.

In their Lagrangian form (using the mass enclosed as the independent variable), the stellar structure equations we wish to solve are outlined below (Kippenhahn et al. 2012).

$$\boxed{\begin{array}{l}\frac{d\mathcal{L}}{d\mathcal{M}_r} = \varepsilon, \\ \frac{dP}{d\mathcal{M}_r} = -\frac{G\mathcal{M}_r}{4\pi r^4}, \\ \frac{dr}{d\mathcal{M}_r} = \frac{1}{4\pi r^2 \rho}, \\ \frac{dT}{d\mathcal{M}_r} = -\frac{G\mathcal{M}_r T}{4\pi r^4 P}\nabla. \end{array}} \tag{1}$$

ε is defined as the total energy generation rate, G is the gravitational constant, $\nabla = \frac{\partial \ln T}{\partial \ln P}$, and ρ is the density at the specific mass coordinate.

We define an equation of state that includes both ideal gas pressure and radiation pressure:

$$P = P_{\text{ideal}} + P_{\text{radiation}} = \frac{\rho k T}{\mu} + \frac{1}{3}aT^3, \tag{2}$$

k and a are constants, while μ is dependent on the mass fraction of hydrogen X: $\mu = \frac{4}{3+5X}$. Equation 2 can be solved for the density as a function of pressure, temperature, and composition, which aids in the calculation of these derivatives.

The total energy generation rate, ε, can be defined in a variety of ways, so we now discuss our treatment of the variable.

2.3. *Energy Generation*

Stars, by definition, fuse elements in their cores, which generates energy that is eventually liberated at the stellar surface. For the purposes of this project, we assume that fusion is the dominant process and is the primary source of energy generation in a star that has just reached the ZAMS. Gravitational contraction is a significant energy source and will cause discrepancies between our simple model and more complex models like MESA's. We wouldn't expect a ZAMS star to have significant energy loss due to neutrino loss, and we thus ignore it. The pp-chains and the CNO cycle are the primary fusion reactions we will implement in this model (Kippenhahn et al. 2012),

$$\frac{d\mathcal{L}}{d\mathcal{M}_r} = \varepsilon = \varepsilon_{pp} + \varepsilon_{CNO}. \tag{3}$$

The individual energy generation rates are described analytically by:

$$\varepsilon_{pp} = 2.57 \times 10^4 \psi f_{11} g_{11} \rho X^2 T_9^{-2/3} e^{-3.381/T_9^{1/3}}, \tag{4}$$

$$g_{11} \equiv 1 + 3.82T_9 + 1.51T_9^2 + 0.144T_9^3 - 0.0114T_9^4. \tag{5}$$

We assume weak screening:

$$f_{11} = e^{E_D/kT}, \tag{6}$$

where $\frac{E_D}{kT} = 5.92 \times 10^{-3} \left(\frac{\rho}{kT}\right)^{1/2}$ as this is the pp-chain and we assume $\zeta \approx 1$. Thus:

$$f_{11} = e^{5.92\times10^{-3} Z_1 Z_2 \left(\frac{\zeta\rho}{T_7^3}\right)^{1/2}}. \tag{7}$$

We also use the Figure 18.7 from Stellar Structure and Evolution to determine the value of ψ:

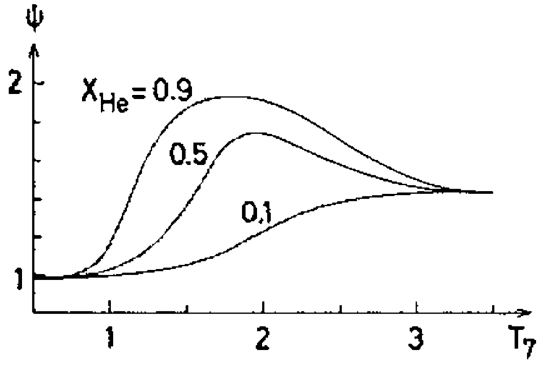

Figure 1. Figure 18.7 from Stellar Structure and Evolution plotting the parameter ψ used in calculating the pp-chain's energy generation (Kippenhahn et al. 2012).

For our purposes, we use some linear relation from a value of 1 for $T_7 < 1$ and 1.4 for $T_7 > 3.5$:

$$\psi \approx \left\{ \begin{array}{ll} 1, & \text{if } T_7 < 1 \\ .15T_7 + .85, & \text{if } 1 < T_7 < 3.5 \\ 1.4, & \text{if } T_7 > 3.5 \end{array} \right\}, \tag{8}$$

$$\begin{aligned} \varepsilon_{CNO} = 8.24 \times 10^4 \psi g_{14,1} X_{CNO} X \rho \\ \times T_9^{-2/3} e^{(-15.231 T_9^{-1/3}) - \frac{T_9}{0.8}^2}, \end{aligned} \tag{9}$$

$$g_{14,1} = 1 - 2.00T_9 + 3.41T_9^2 - 2.43T_9^3, \tag{10}$$

where $X_{CNO} = X_C + X_N + X_O \approx .71Z$ using approximately solar composition of C, N, O, and metals (Z) (Iglesias & Rogers 1996).

With energy generation covered, we investigate how energy transport occurs.

2.4. *Energy Transport*

After energy generation, we focus on the transport of energy, $\frac{dT}{d\mathcal{M}_r}$. The factor ∇ is an important parameter because it separates radiative from convective energy transport – two very different processes that impact stellar structure of a star.

When we calculate the two values ∇_{ad} and ∇_{rad}, we assign ∇ to be the smaller of the two values, meaning the more efficient energy transport mechanism in a specific portion of the star.

2.4.1. *Radiative Energy Transport*

With convection neglected and local thermodynamic equilibrium (LTE) we make the assertion (based on radiative transport):

$$\nabla = \nabla_{rad} = \left(\frac{d\ln T}{d\ln P}\right)_{rad} = \frac{3}{16\pi ac}\frac{P\kappa}{T^4}\frac{\mathcal{L}_r}{G\mathcal{M}_r}. \quad (11)$$

The presence of convection in a stellar core (c) impacts the temperature close to the core ($\mathcal{M}_r \approx .001\mathcal{M}$):

$$\ln T_r = \ln T_c - \left(\frac{\pi}{6}\right)^{1/3} G\frac{\nabla_c \rho_c^{4/3}}{P_c}\mathcal{M}_r^{2/3}. \quad (12)$$

2.4.2. *Convective Energy Transport*

For convective energy transport, assuming that the transport is adiabatic:

$$\nabla = \nabla_{ad} = \left(\frac{d\ln T}{d\ln P}\right)_{ad} = \frac{\gamma - 1}{\gamma}. \quad (13)$$

Assuming complete ionization (no partial ionization of H or He), $\gamma = \frac{5}{3}$, making $\nabla_{ad} = 0.4$.

In the stellar core, we can analytically express the temperature by:

$$T_r^4 = T_c^4 - \frac{1}{2ac}\left(\frac{3}{4\pi}\right)^{2/3} \kappa_c \varepsilon_c \rho_c^{4/3} \mathcal{M}_r^{2/3}. \quad (14)$$

In summary:

$$\nabla = \left\{ \begin{array}{r} .4, \text{ if } \nabla_{rad} > \nabla_{ad} \\ \frac{3}{16\pi ac}\frac{P\kappa}{T^4}\frac{\mathcal{L}_r}{G\mathcal{M}_r}, \text{ if } \nabla_{rad} < \nabla_{ad} \end{array} \right\}. \quad (15)$$

2.5. *Choice of Star*

Given the assumptions/nuances above, we can make a general estimate of the type of ZAMS star that will be best modeled by our simple model. Neglecting gravitational contraction, we must use a star that is not so massive as to leverage our neglecting of this term. However, we must use a star that is massive enough such that the assumption of full ionization is true – otherwise we will have significant deviations in our convective energy transport analysis. Stars between 1 and 1.5 solar masses ($M_\odot$) have the largest ratios between their core and average densities, meaning, numerically, we would expect poor convergence around a guess that assumes constant density. **For the above reasons, we believe the use of a 2 solar mass ZAMS star is an ideal mass to model.** While the specific number is arbitrary, anything larger or smaller than this could break our model.

For simplicity, we make assumptions of solar composition C, N, and O mass fractions, as well as hydrogen, helium, and metal mass fractions of X=0.70, Y=0.28, and Z=0.02, which are consistent with the present day sun. More importantly, these values are consistent with the opacity tables we use.

Now that we have a working understanding of the differential equations we must solve and the details of each, we can now proceed to actually solving these equations with the given conditions explained above.

3. NUMERICAL METHODS

Of equal importance to our analysis of the actual differential equations is the solving of the set of differential equations. Checking our solution, treating boundary conditions, and deciding when we have a converged solution are just some of the intricacies we must address in our path to modeling a ZAMS star.

One of the most simple ways to numerically solve a boundary value problem: shooting for a solution (Press et al. 2007).

3.1. *The Shooting Method*

The shooting method is a robust numerical technique that is widely used to solve boundary value problems (BVPs) that lack analytical solutions or are challenging to solve using other methods. It involves transforming the BVPs into a set of initial value problems (IVPs) and solving them in both forward and backward directions from the interval endpoints (inward and outward from the surface and core for our use) and then adjusting the initial conditions iteratively to satisfy both sets of boundary conditions.

For our purposes, we will define boundary values at the surface and the core, numerically integrate inwards from the surface, outwards from the core, then iteratively change the boundary conditions individually until the two integrations meet somewhere in the middle of the interval.

We can define a vector function for the differences at the meeting point:

$$\mathbf{f}(L, P, R, T) = \{L_i - L_o, P_i - P_o, R_i, R_o, T_i - T_o\}_{\xi_1} \quad (16)$$

In plain words, this function is the difference between the inner (i) and outer (o) shots at some intermediate point ξ_1. If we minimize this function to within some threshold, we can make the claim that our solution has

converged, as the inner and outer shots agree with each other (again, within some threshold).

This begs the question of what this intermediate point is and where is will be placed. This point is some fraction of the mass of the star where we decide would be best for the two IVPs to meet; for this application we choose about a fifth of the total star's mass as the cutoff point. As most of the mass is concentrated in the core, we picked a fifth ($\frac{1}{5}\mathcal{M}_*$) to provide the surface-based IVP enough room to actually shoot, while not trying to fit too close to the core where there are rapid changes in ZAMS structure.

3.2. *Opacities*

Preemptively, for the following sections, we discuss how we will find opacities for each portion of the star.

To do this, we take opacities from the Lawrence Livermore National Laboratory's OPAL database (Iglesias & Rogers 1996), which provides us opacity values for solar composition stars from log R values of -8 to 1, and log T values from 3.75 to 8.70 (Grevesse et al. 1996). We also concatenate opacity values for log T below 3.75, just in case they are necessary for surface calculations where temperatures are low (Asplund et al. 2009), (Alexander & Ferguson 1994).

We took these 2D tables and created an interpolator function that uses density and temperature to return an interpolated opacity value. We use linear interpolation (`scipy.interpolate.griddate`) for ease, and we have no reason to expect any more complicated interpolation methods to be significantly more accurate.

Note:

$$R = \frac{\rho}{T_6^3} \tag{17}$$

3.3. *Boundary Conditions/Initial Values*

Now, we must define the boundary conditions that we will implement as the initial values for the IVPs we will shoot solutions for.

3.3.1. *Surface Boundary*

The surface boundary conditions are relatively straight forward. We assume some radius and luminosity of the total star that we obtain from low-mass homology relations (Hansen et al. 2004):

$$\boxed{\begin{aligned} \frac{\mathcal{L}}{\mathcal{L}_\odot} &\approx \left(\frac{\mathcal{M}}{\mathcal{M}_\odot}\right)^{3.9}, \\ \frac{\mathcal{R}}{\mathcal{R}_\odot} &\approx \left(\frac{\mathcal{M}}{\mathcal{M}_\odot}\right)^{0.2}. \end{aligned}} \tag{18}$$

We can then make pressure and temperature estimates based on these values. For the pressure, we can use the surface gravity to state:

$$P = \frac{2g}{3\kappa} = \frac{G\mathcal{M}}{R^2}\frac{2}{3\kappa}. \tag{19}$$

For temperature, we use the effective surface temperature:

$$\mathcal{L} = 4\pi R^2 \sigma T_{\text{eff}}^4 \rightarrow T_{\text{eff}} = \left(\frac{\mathcal{L}}{4\pi R^2 \sigma}\right)^{1/4}. \tag{20}$$

The largest complexity here is to figure out the value of the surface opacity. To do this, we note that our pressure relation must obey our equation of state, thus:

$$\frac{2g}{3\kappa} = \frac{\rho k T}{\mu} + \frac{1}{3}aT^3. \tag{21}$$

Now we have an issue knowing the surface density. However, using our interpolation routine, we can find the opacity value given a density, as we have a temperature estimate already. Thus, we really need to fit for some density that makes Equation 21 true, then we can make a claim about our estimated surface opacity. This is what we do to estimate κ and thus define the surface pressure.

In total, our surface boundary values/initial conditions can be described by:

$$\boxed{\begin{aligned} &\mathcal{L}_{\mathcal{M}_r=\mathcal{M}} = \mathcal{L}, \\ &P_{\mathcal{M}_r=\mathcal{M}} = \tfrac{G\mathcal{M}}{R^2}\tfrac{2}{3\kappa}, \\ &r_{\mathcal{M}_r=\mathcal{M}} = R, \\ &T_{\mathcal{M}_r=\mathcal{M}} = T_{\text{eff}} = \left(\tfrac{\mathcal{L}}{4\pi R^2\sigma}\right)^{1/4}. \end{aligned}} \tag{22}$$

3.3.2. *Core Boundary*

In the core, we have issues with singularities at the exact center, where $\mathcal{M}_{r=0} = 0$. It is clear that three of our equations will be indeterminant or non-existent, which is obviously an issue for integration when the initial values do not exist. Thus, we will actually start our integration at a somewhat-arbitrary point away from the star's core, at say $\mathcal{M} = 1 \times 10^{-8}\mathcal{M}_*$.

We must provide guesses for the star's central pressure and temperature and then we must correct for our small perturbation from the core.

We outlined in Section 2.4 the temperature corrections for temperatures close to the core given radiative or convective energy transport. Now, we must do the same for pressure corrections:

$$P_{\mathcal{M}_r} = P_c - \frac{3G}{8\pi}\left(\frac{4\pi}{3}\rho_c\right)^{4/3}\mathcal{M}_r^{1/3}. \tag{23}$$

Close to the core we can assume the density is approximately constant, meaning for corrections to our radial coordinate, we can assume:

$$r_{\mathcal{M}_r} = \left(\frac{3}{4\pi\rho_c}\right)^{1/3} \mathcal{M}_r^{1/3}. \tag{24}$$

In total, our core's initial values will take the form:

$$\boxed{\begin{aligned} &\mathcal{L}_{\mathcal{M}_r=d\mathcal{M}} = \varepsilon\mathcal{M}_r, \\ &P_{\mathcal{M}_r=d\mathcal{M}} = P_c - \tfrac{3G}{8\pi}\left(\tfrac{4\pi}{3}\rho_c\right)^{4/3}\mathcal{M}_r^{1/3}, \\ &r_{\mathcal{M}_r=d\mathcal{M}} = \left(\tfrac{3}{4\pi\rho_c}\right)^{1/3}\mathcal{M}_r^{1/3}, \\ &T_{\mathcal{M}_r=d\mathcal{M}} = T_r. \end{aligned}} \tag{25}$$

All we must do is provide an initial pressure and temperature guess, which will be iteratively changed towards a convergent value, along with the surface boundary conditions. These initial guesses for pressure and temperature will be derived from the constant density model (Hansen et al. 2004).

$$\boxed{\begin{aligned} P_c &= \tfrac{3}{8\pi}\tfrac{G\mathcal{M}^2}{\mathcal{R}^4}, \\ T_c &= \tfrac{1}{2}\tfrac{G\mathcal{M}}{R}\tfrac{\mu}{N_A k}. \end{aligned}} \tag{26}$$

Now, we must investigate how we're iterating, changing, and evaluating our parameters in the model.

3.4. *Iteration*

Before minimizing the difference between our two initial value problems at our cutoff point, we must integrate the initial value problems. The package `Scipy` has an initial value problem solver, `optimize.solve_ivp` that will integrate the differential equations for us (given the initial values and the mass interval over which we are integrating). For the solving, we specifically made use of Runge-Kutta, or RK45, that the integrator already had available to use (Virtanen et al. 2020).

To speed up the iterations, we implemented multiprocessing via pooling using the package `Ray` (Moritz et al. 2017). The two inner and outer integrations being parallelized saves plenty of time.

3.5. *Newton-Raphson*

To minimize the difference between the inner and outer integrations, as shown by the function in Equation 16, we use the Newton-Raphson (NR) method of convergence. If we are able to minimize Equation 16 to some threshold, then we can be confident that our code has found a converged solution. However, the NR method implemented by standard packages, like `Scipy` will take too large of steps, so we must build our own NR minimizer that will take more careful step sizes as to not shoot into unstable/undesireable regions.

The Newton-Raphson root-finding method for a nonlinear set of equations is:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{J}^{-1}(\mathbf{x}_n)\mathbf{f}(\mathbf{x}_n)\,. \tag{27}$$

We do not have the derivatives of the function we are minimizing, thus:

$$J_{i,j}(\mathbf{x}_n) = \frac{\partial f_i}{\partial x_j} \approx \frac{\Delta f_i}{\Delta x_i} = \frac{f_i(\mathbf{x}+h_j) - f_i(\mathbf{x})}{h_j}. \tag{28}$$

Note that h_j is applied to the jth element of $\mathbf{x}$. We use an array of $\mathbf{h} = [1\times10^{33}, 1\times10^{16}, 1\times10^{8}, 1\times10^{4}]$ that are appropriate to the level of accuracy we require and the relative magnitudes of the parameters themselves.

To prevent overshooting, we can introduce some overshooting parameter (δ_o) that lessens the step that the NR algorithm takes so our updated guesses will stay in a stable region:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{J}^{-1}(\mathbf{x}_n)\mathbf{f}(\mathbf{x}_n)\,\delta_o. \tag{29}$$

To determine when the model has converged, we use relative values of how much the the two integrations agree, compared to the values we input. We set our threshold at .01, so that when the two integrations differ by less than .01, relative to the parameters' original value, we can assume that the model is well converged. Simply, we require no more than 1% difference between the inner and outer integrations. The values will not all be the same, so we require that the maximum difference across all parameters be 1%.

3.6. *Constant Density Models*

The constant density model is not a very good estimation of a true star, especially for the core's conditions, which estimates pressures too low by a few orders of magnitude. Newton-Raphson is sensitive to our initial guess, so we will numerically find a better estimate before using our minimizer. To do this, we take the pressure and temperature, increase them (holding the surface parameters constant), then update them for 2 or 3 more steps using the differences in the inner and outer integrations. While not ideal, as this does not simultaneously change all parameters, this will speed up the Newton-Raphson method and ensure convergence. Without this step, the minimizer shoots into unstable regions for certain masses.

Now, given the numerical techniques described above, we can implement these and obtain a converged solution for a 2.0 solar mass star.

4. RESULTS

The code used for this project can be found in the project's Github repository. A machine readable table with various parameters at select mass coordinates can be found in Table 2 on page 8.

The results of our converged solution, along with our initial guess and MESA's solution are outlined below (Paxton et al. 2011).

Model	$\mathcal{L}/\mathcal{L}_\odot$	log P	$\mathcal{R}/\mathcal{R}_\odot$	log T
Initial Guess	14.9285	15.4900	1.1487	7.0896
Converged Solution	15.3300	17.2191	1.6691	7.3140
MESA Solution	16.1325	17.2638	1.6714	7.3203
Percent Errors (%)	4.97	9.79	.13	1.46
Final Diff. Errors	.0097	.0046	.0006	.0012

Table 1. Table of parameter values containing the initial guess, converged solution, MESA's solution, percent errors between MESA and the converged solution, as well as the fractional differences between the inward and outward integrations (normalized to the parameter values). Note: $\mathcal{L}_\odot = 3.839 \times 10^{33}$ erg/s, $\mathcal{R}_\odot = 6.957 \times 10^{10}$ cm

Our solutions were cut off when the difference between the two integrations at the meeting point was less than 1% for the parameter with the largest difference. This took 39 integrations.

Paper I from the Mesa Isochrones and Stellar Tracks (MIST) models defines a star reaching the ZAMS when $\mathcal{L}_{nuc}/\mathcal{L} \geq 0.9$ in MESA (Choi et al. 2016). We follow this guideline for the values in Table 1. We use the same stellar mass to run this calculation and used solar composition.

All of the values in Table 1 are within 10% of the expected MESA values, but there is some difference. This is to be as expected, given the poor physics that we have implemented in our model. As a star enters the main sequence, gravitational contraction plays a significant role, meaning our luminosity values are understandably different. Given the coupling of the temperature, pressure, and energy generation rate, the issue with the physics propagates to other parameters. All things given, however, our converged solution is in agreement with MESA's values.

Other mass values were tested above and below 2 solar masses, but the 2 solar mass model seemed to run the most accurate in a small number of iterations.

4.1. *Stellar Structure*

The star's structure (cumulative energy generation, pressure, radius, and temperature vs mass) can be visualized in the Figure 4.1:

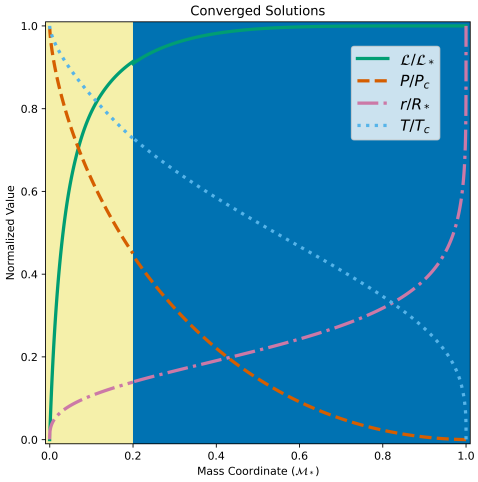

Figure 2. Diagram showing ZAMS star's interior structure via luminosity, pressure, radius, and temperature. Note that the values are normalized. The left color denotes the outward integration region, while the right color denotes the inward integration region. Solid-green line indicated energy generation, dashed-red indicates pressure structure, dot-dash-purple indicates radius, dotted-blue indicates temperature structure

We normalized the values to their maximum values in an effort to make a more clear plot. We see at the boundary between the inward and outward integration that there is a slight discrepancy in the luminosity especially, which is expected given the fractional differences outlined above. Otherwise, the behavior appears appropriate and consistent with typical ZAMS stellar structure behavior.

5. CONCLUSION

In this project, we built a basic ZAMS stellar structure model created by numerically solving the coupled stellar structure equations outlined in box 1. This model, obtained using the shooting method, provides a simplified, yet accurate, understanding of the internal structure of ZAMS stars. Beginning with a rudimentary, constant-density stellar model, we were able to build a model that agrees with more complex models (like MESA), validating its reliability and usefulness in explaining stellar structure. While we are ignoring important physics like rotation, inhomogeneous composition, gravitational contraction, etc., we still show that the basic modeling of a ZAMS star is reasonably achievable. The ability for our model to converge is due to the simplified assumptions (some listed above), as computational time is relatively low and the complexity of the model is min-

imal, meaning we do not need perfect physics and is not highly sensitive to our computational techniques.

Stars are inherently three dimensional objects and require modeling in three dimensions, but computational power is expensive, meaning simple models, like the one we have constructed here, are an important and valuable exercise.

REFERENCES

Alexander, D. R., & Ferguson, J. W. 1994, ApJ, 437, 879, doi: 10.1086/175039

Asplund, M., Grevesse, N., Sauval, A. J., & Scott, P. 2009, ARA&A, 47, 481, doi: 10.1146/annurev.astro.46.060407.145222

Choi, J., Dotter, A., Conroy, C., et al. 2016, The Astrophysical Journal, 823, 102, doi: 10.3847/0004-637x/823/2/102

Grevesse, N., Noels, A., & Sauval, A. J. 1996, in Astronomical Society of the Pacific Conference Series, Vol. 99, Cosmic Abundances, ed. S. S. Holt & G. Sonneborn, 117

Hansen, C. J., Kawaler, S. D., & Trimble, V. 2004, Stellar Interiors (Springer-Verlag)

Iglesias, C. A., & Rogers, F. J. 1996, ApJ, 464, 943, doi: 10.1086/177381

Kippenhahn, R., Weiss, A., & Weigert, A. 2012, Stellar Structure and Evolution (Springer-Verlag)

Moritz, P., Nishihara, R., Wang, S., et al. 2017, CoRR, abs/1712.05889

Paxton, B., Bildsten, L., Dotter, A., et al. 2011, ApJS, 192, 3, doi: 10.1088/0067-0049/192/1/3

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 2007, Numerical Recipes: The Art of Scientific Computing (Cambridge University Press)

Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, Nature Methods, 17, 261, doi: 10.1038/s41592-019-0686-2

Table 2. Machine readable table containing select interior parameter values at various Lagrangian mass coordinates.

$\mathcal{M}/\mathcal{M}_*$	$\mathcal{L}/\mathcal{L}_*$	log P	$\mathcal{R}/\mathcal{R}_*$	log T	log ρ	ε	∇_{ad}	∇	Transport
		dyne/cm^2		K	g/cm^3	erg/g/s	K/cm	K/cm	Nature
1.000000e-08	3.332010e-07	17.21910	0.000466	7.31397	1.776360	4.929170e+02	0.4	0.400000	Convective
4.790100e-04	1.519530e-02	17.21400	0.016951	7.31195	1.773330	4.539800e+02	0.4	0.400000	Convective
3.777510e-03	1.038070e-01	17.19880	0.033899	7.30587	1.764230	3.551490e+02	0.4	0.400000	Convective
1.248450e-02	2.736750e-01	17.17340	0.050847	7.29570	1.749020	2.376890e+02	0.4	0.400000	Convective
2.873600e-02	4.726440e-01	17.13770	0.067796	7.28142	1.727670	1.386400e+02	0.4	0.400000	Convective
5.406300e-02	6.454100e-01	17.09160	0.084746	7.26297	1.700070	7.329760e+01	0.4	0.400000	Convective
8.922210e-02	7.706090e-01	17.03490	0.101695	7.24030	1.666140	3.763960e+01	0.4	0.400000	Convective
1.344020e-01	8.537740e-01	16.96740	0.118644	7.21328	1.625760	2.020780e+01	0.4	0.400000	Convective
1.886410e-01	9.099450e-01	16.88870	0.135593	7.18225	1.578110	1.164540e+01	0.4	0.379794	Radiative
3.142300e-01	9.626190e-01	16.69950	0.169492	7.12084	1.450500	4.429730e+00	0.4	0.308750	Radiative
3.829160e-01	9.788020e-01	16.59140	0.186441	7.08873	1.374460	2.672060e+00	0.4	0.286178	Radiative
4.515430e-01	9.885270e-01	16.47560	0.203390	7.05679	1.290640	1.580140e+00	0.4	0.267926	Radiative
5.178250e-01	9.939920e-01	16.35280	0.220339	7.02500	1.199610	9.143090e-01	0.4	0.250523	Radiative
5.799760e-01	9.969180e-01	16.22390	0.237288	6.99362	1.102120	5.204690e-01	0.4	0.236395	Radiative
6.370010e-01	9.984650e-01	16.09160	0.254237	6.96312	1.000280	2.961110e-01	0.4	0.227089	Radiative
6.883590e-01	9.992500e-01	15.95610	0.271186	6.93293	0.894985	1.662610e-01	0.4	0.219167	Radiative
7.338460e-01	9.996370e-01	15.81830	0.288136	6.90324	0.786852	9.244860e-02	0.4	0.211094	Radiative
7.736680e-01	9.998280e-01	15.68000	0.305085	6.87458	0.677141	5.140100e-02	0.4	0.206299	Radiative
8.379310e-01	9.999620e-01	15.40200	0.338983	6.81825	0.455405	1.560470e-02	0.4	0.199136	Radiative
8.850370e-01	9.999920e-01	15.12630	0.372882	6.76449	0.233366	4.733190e-03	0.4	0.194017	Radiative
9.034260e-01	9.999960e-01	14.98990	0.389830	6.73821	0.123163	2.601750e-03	0.4	0.193118	Radiative
9.190140e-01	9.999980e-01	14.85470	0.406780	6.71244	0.013614	1.431180e-03	0.4	0.191858	Radiative
9.321910e-01	9.999990e-01	14.72040	0.423729	6.68654	-0.094828	7.819060e-04	0.4	0.192785	Radiative
9.527150e-01	1.000000e+00	14.45540	0.457627	6.63570	-0.309187	2.324540e-04	0.4	0.195414	Radiative
9.606380e-01	1.000000e+00	14.32410	0.474576	6.60969	-0.414607	1.247170e-04	0.4	0.198508	Radiative
9.729450e-01	1.000000e+00	14.06310	0.508475	6.55778	-0.623842	3.518400e-05	0.4	0.201735	Radiative
9.776760e-01	1.000000e+00	13.93290	0.525425	6.53134	-0.727654	1.834480e-05	0.4	0.205630	Radiative
9.877890e-01	1.000000e+00	13.54000	0.576272	6.44680	-1.036320	2.265410e-06	0.4	0.226194	Radiative
9.901340e-01	1.000000e+00	13.40620	0.593220	6.41637	-1.139660	1.060350e-06	0.4	0.231324	Radiative
9.920880e-01	1.000000e+00	13.27040	0.610168	6.38454	-1.243760	4.768620e-07	0.4	0.235580	Radiative
9.950470e-01	1.000000e+00	12.99000	0.644067	6.31949	-1.459210	8.832460e-08	0.4	0.231474	Radiative
9.970260e-01	1.000000e+00	12.69510	0.677967	6.25112	-1.685780	1.397550e-08	0.4	0.233612	Radiative
9.977380e-01	1.000000e+00	12.54080	0.694922	6.21564	-1.804730	5.193790e-09	0.4	0.229541	Radiative
9.983040e-01	1.000000e+00	12.38100	0.711870	6.17932	-1.928230	1.836990e-09	0.4	0.227074	Radiative
9.990920e-01	1.000000e+00	12.04250	0.745753	6.10308	-2.190660	1.909120e-10	0.4	0.224120	Radiative
9.993550e-01	1.000000e+00	11.86180	0.762701	6.06351	-2.332000	5.587790e-11	0.4	0.221145	Radiative
9.996960e-01	1.000000e+00	11.47130	0.796599	5.97791	-2.637210	3.539230e-12	0.4	0.216217	Radiative
9.998000e-01	1.000000e+00	11.25840	0.813551	5.93308	-2.805510	7.714620e-13	0.4	0.208235	Radiative
9.998730e-01	1.000000e+00	11.03230	0.830464	5.88666	-2.985540	1.504000e-13	0.4	0.205371	Radiative
9.999220e-01	1.000000e+00	10.78820	0.847499	5.83676	-3.180100	2.456040e-14	0.4	0.204163	Radiative
9.999540e-01	1.000000e+00	10.52840	0.864287	5.78406	-3.387670	3.391860e-15	0.4	0.203482	Radiative
9.999870e-01	1.000000e+00	9.92915	0.897929	5.65970	-3.863740	2.517070e-17	0.4	0.211358	Radiative
9.999940e-01	1.000000e+00	9.56627	0.915018	5.58240	-4.150070	1.001050e-18	0.4	0.215140	Radiative
9.999970e-01	1.000000e+00	9.16065	0.931455	5.49517	-4.469360	2.178200e-20	0.4	0.216297	Radiative
9.999990e-01	1.000000e+00	8.74122	0.945781	5.40226	-4.796740	2.974680e-22	0.4	0.231281	Radiative
9.999990e-01	1.000000e+00	8.42696	0.954853	5.32661	-5.035570	7.871260e-24	0.4	0.253772	Radiative
1.000000e+00	1.000000e+00	3.27834	1.000000	3.94678	-8.799520	4.750320e-73	0.4	0.125224	Radiative

NOTE—$\mathcal{M}_* = 2\mathcal{M}_\odot$, $\mathcal{L}_* = 15.33\mathcal{L}_\odot$, $\mathcal{R}_* = 1.67\mathcal{R}_\odot$