Draft version December 9, 2014
Preprint typeset using LaTeX style emulateapj v. 12/16/11

STELLAR STRUCTURE MODEL

Eric Gentry
Astro 220A
(Dated: December 9, 2014)
Draft version December 9, 2014

ABSTRACT

We modeled a *Zero-Age Main Sequence* star of mass $M_* = 2M_\odot$, assuming a uniform solar-like composition. Using appropriate initial boundary conditions chosen from Hansen *et al.* (2004), we employed a *shoot to a fit* method, until our boundary conditions resulted in a self-consistent model. These boundary conditions are compared to the initial boundary conditions, and agree to within about 15%. When compared to more sophisticated models, our results agree within 5-10%. We discuss the effects of observational and numeric systematics which might contribute to those errors.

1. INTRODUCTION

While the properties and evolution of stars are an ongoing topic of interest, the typical densities and opacities of stars make it difficult to directly probe much more than their outermost layers. One indirect way we can learn about the inner regions of stars is through modeling. By assuming a set of physical processes at play, and boundary parameters regulating those processes, we can model what the resulting star would look like, and compare that to the limited observation data we do have. Through this approach, particularly when paired with Bayesian statistics, we can determine the likelihood of any given set of physical assumptions and regulating parameters.

The scope of this work will be to simply create a single, self-consistent model of stellar structure. We will use observation and empirical data to create an initial model, then we will slowly vary those input parameters until our model is self-consistent. Then, we will compare our structure model to models produced by different approaches, including different physics, to gain an understanding of the robustness of our results.

2. THEORY

This model followed the outline of Kippenhahn *et al.* (2012, Chapter 10). Here the main equations are summarized, with the details left to later sections.

Using mass enclosed, m, as the independent variable, mass conservation and hydrostatic equilibrium (respectively) first give us:

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

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

where r is the radius a given shell, ρ is the density, P is the pressure, and G is the gravitational constant.

Similarly, the conservation of energy in a static star gives us the definition of the energy generation rate, ϵ:

$$\frac{\partial \ell}{\partial m} \equiv \epsilon \tag{3}$$

egentry@ucsc.edu
https://github.com/egentry/stellarstructure

where ℓ is the local luminosity, such that $\ell(M_*) \equiv L_*$. For details on the energy generation rate, ϵ, see Section 2.1.

The energy being generated in the core, while energy is being radiated at the surface; in order to remain in local thermodynamic equilibrium there needs to be transport of energy within the star. This transport is

$$\begin{aligned}\frac{\partial T}{\partial m} &= \frac{\partial P}{\partial m}\frac{\partial T}{\partial P} = \frac{\partial P}{\partial m}\frac{\partial T}{\partial P} = \frac{\partial P}{\partial m}\frac{T}{P}\frac{\partial \ln T}{\partial \ln P} \\ &= -\frac{Gm}{4\pi r^4}\frac{T}{P}\frac{\partial \ln T}{\partial \ln P} \\ &\equiv -\frac{Gm}{4\pi r^4}\frac{T}{P}\nabla \end{aligned} \tag{4}$$

where T is the local temperature, and the last line simply defines $\nabla \equiv \frac{\partial \ln T}{\partial \ln P}$, which is discussed in Section 2.2.

Finally we also include an equation of state:

$$P = P_{\rm gas} + P_{\rm rad} = \frac{\rho}{\mu}\mathcal{R}T + \frac{aT^4}{3} \tag{5}$$

where μ is the mean molecular weight, $\mathcal{R}$ is the ideal gas constant, and a is the radiation constant.

Given the system of Equations 1 - 4, we can integrate to find the physical parameters $\{r, \ell, P, T\}$ at any point within the star. The numerical techniques for doing so will be covered in Section 3.

2.1. *Energy Generation*

Equation 3 above gave us:

$$\frac{\partial \ell}{\partial m} \equiv \epsilon$$

but now we must determine ϵ. In principle, ϵ will be a combination of the rates of nuclear energy generation, neutrino losses and gravitational contraction. For a star on the main sequence though, we only expect nuclear energy generation to be significant. Neutrino losses only become more significant at later stellar stages, and we do not expect main sequence stars to be contracting.

For a star on the main sequence, we expect hydrogen burning to be the main source of energy (rather than the burning of larger elements). Within hydrogen burning, we have two primary branches, the *pp chain* and the

CNO cycle. The total energy generation will simply be the sum of the energy generated by those two cycles:

$$\epsilon = \epsilon_{pp} + \epsilon_{\rm CNO} \tag{6}$$

2.1.1. pp *Chain*

Determining the respective energy generation rates for those cycles at a given temperature, density and composition is a question of nuclear physics. For simplicity we adopt the approximations of Angulo *et al.* (1999):

$$\epsilon_{pp} = 2.57 \cdot 10^4 \psi f_{11} g_{11} \rho X_1^2 T_9^{-2/3} {\rm e}^{-3.381/T_9^{1/3}} \tag{7}$$

$$g_{11} \equiv 1 + 3.82 T_9 + 1.51 T_9^2 + 0.144 T_9^3 - 0.0114 T_9^4 \tag{8}$$

where ψ accounts for the relative strength of the *pp*-subchains at different conditions, f_{11} is the electron shielding factor (we assume weak shielding: $f_{11} \approx 1$), X_1 is the hydrogen mass fraction, X, and T_9 follows the standard convention $T_9 \equiv T/10^9 K$. While ψ can be calculated through nuclear physics (cf. Parker *et al.* 1964), we once again approximate it, using a ramp approximation valid for a helium mass fraction $Y \approx .2$:

$$\psi = \begin{cases} 1 & T_7 < 1 \\ 1 + \frac{1}{2}(T_7 - 1) & 1 < T_7 < 3 \\ 2 & 3 < T_7 \end{cases} \tag{9}$$

2.1.2. *CNO Cycle*

For the CNO-cycle, we have a set of equations similar to Equation 7, also following the work of Angulo *et al.* (1999):

$$\epsilon_{\rm CNO} = 8.24 \cdot 10^2 5 g_{14,1} X_{\rm CNO} X_1 \rho \times T_9^{-2/3} {\rm e}^{\left(-15.231 T_9^{-1/3}\right) - (T_9/0.8)^2} \tag{10}$$

$$g_{14,1} \equiv 1 - 2.00 T_9 + 3.41 T_9^2 - 2.43 T_9^3 \tag{11}$$

with $X_{\rm CNO}$ being the combined mass fraction of carbon, nitrogen and oxygen:

$$X_{\rm CNO} \equiv X_{\rm C} + X_{\rm N} + X_{\rm O} \approx 0.7 Z \tag{12}$$

where Z is the mass fraction of all metals.

2.2. *Energy Transport*

Equation 4 left us with:

$$\frac{\partial T}{\partial m} = -\frac{Gm}{4\pi r^4} \frac{T}{P} \nabla$$

with ∇ defined as:

$$\nabla \equiv \frac{\partial \ln T}{\partial \ln P}$$

so in order to calculate $\partial T/\partial m$, we need to know ∇. As ∇ is being used as a proxy for the transfer of energy, we must know the method of energy transport.

Within a stellar interior the two main mechanisms of energy transport are radiative (diffusive) transport and adiabatic (convective) transport. Each mechanism has an associated ∇, and in principle ∇ will be a combination of each component ∇, but in practice, ∇ will be dominated by the contribution from the most efficient transport mechanism:

$$\nabla = \min\left(\nabla_{\rm rad}, \nabla_{\rm ad}\right) \tag{13}$$

2.2.1. *Adiabatic Transport*

For adiabatic (convective) transport, $\nabla_{\rm ad}$ can simply be calculated using the definition of an adiabat, $d(P\rho^{-\gamma})_{\rm ad} = 0$, and the equation of state for an ideal gas, $P = \frac{\rho}{\mu}\mathcal{R}T$:

$$\begin{aligned} \nabla_{\rm ad} &= \left(\frac{\partial \ln T}{\partial \ln P}\right)_{\rm ad} \\ &= \frac{\partial \ln P\rho^{-1}}{\partial \ln \rho^{\gamma}} \\ &= \frac{\partial \ln \rho^{\gamma-1}}{\partial \ln \rho^{\gamma}} \\ \nabla_{\rm ad} &= \frac{\gamma - 1}{\gamma} \end{aligned} \tag{14}$$

and for typical conditions of a star (neglecting hydrogen and helium partial ionization zones) we have $\gamma = 5/3$ yielding:

$$\nabla_{\rm ad} = 0.4 \tag{15}$$

For more details on the adiabatic transport of energy, see Kippenhahn *et al.* (2012, Section 4.3).

2.2.2. *Radiative Transport*

Given the condition of Local Thermodynamic Equilibrium in a spherically symmetric geometry, paired with the fundamental law of radiative transport we find:

$$\frac{\partial T}{\partial r} = -\frac{3}{16\pi ac} \frac{\kappa \rho \ell}{r^2 T^3} \tag{16}$$

where a is the radiation constant, c is the speed of light, and κ is the mean opacity. (While this also holds for the more general case including conduction: $\kappa^{-1} = \kappa_{\rm rad}^{-1} + \kappa_{\rm cond}^{-1}$, we typically find that $\kappa_{\rm rad}$ dominates.) Changing dependent variables from radius to mass, we get:

$$\frac{\partial T}{\partial m} = -\frac{3}{64\pi^2 ac} \frac{\kappa \ell}{r^4 T^3} \tag{17}$$

Combining Equations 4 and 17 results in:

$$\nabla_{\rm rad} = \frac{3}{16\pi acG} \frac{\kappa \ell P}{m T^4} \tag{18}$$

3. NUMERICAL METHODS

Equations 1 - 4 define the structure of a star, but are not generally analytically integrable. In order to find $\{r, \ell, P, T\}$ with a given mass enclosed, m, the system of differential equations need to be evaluated numerically. Broadly, this can be accomplished using a numerical integration algorithm (such as a Runge-Kutta method) and knowledge of the boundary conditions for $\{r, \ell, P, T\}$. Given that both the core and the stellar surface are numerically sensitive, and we can only guess the boundary conditions for a star of a given mass, we employed a *shooting to a fit* algorithm. This algorithm took guessed core and boundary conditions, and integrated to a set fitting point within a star. The guessed boundary conditions were then iteratively varied until the inward and outward integrations converged on the same values at the fitting point.

The chosen initial boundary conditions are laid out in Section 3.2. The integration method to get to the fitting

point is summarized in Section 3.3. Once at the fitting point, the inward and outward values will not agree; Section 3.4 will detail the algorithm to iterate the boundary conditions until convergence is achieve, and how the convergence criterion is defined.

3.1. *Opacity Interpolation*

Before being able to determine the boundary conditions or begin any integrations, it is first necessary to determine the local mean opacity, κ. Rather than use a semi-analytic model, we incorporated an empirical grid of opacities from the Opacity Project (Badnell *et al.* 2005), given as a function of temperature and density. From that grid of opacities, we constructed a 2d-interpolation function which was a cubic polynomial in $\{\log T, \log \rho\}$.

3.2. *Boundary Conditions*

3.2.1. *Core Boundary*

Given the geometry of spherical symmetry, $m = 0$ constitutes a singularity for many equations. Given the core temperature and pressure (initial estimates taken from Hansen *et al.* 2004), we can determine the central density using the equation of state (namely, an ideal gas with additional radiation pressure). Knowing central density, temperature and pressure, we can use Equations 1 - 4 to expand our dependent variables, $\{r, \ell, P, T\}$, to lowest order in m, for a small step outside $m = 0$:

$$r(m) \approx \left(\frac{3}{4\pi\rho_c}\right) m^{1/3} \tag{19}$$

$$\ell(m) \approx \epsilon_c m \tag{20}$$

$$P(m) \approx P_c - \frac{3G}{8\pi}\left(\frac{4\pi}{3}\rho_c\right)^{4/3} m^{2/3} \tag{21}$$

$$T(m) \approx \begin{cases} \left(T_c - \frac{1}{2ac}\left(\frac{3}{4\pi}\right)^{2/3}\kappa_c\epsilon_c\rho_c^{4/3}m^{2/3}\right)^{1/4} & \nabla_{\rm ad} > \nabla_{\rm rad} \\ T_c \exp\left[-\left(\frac{\pi}{6}\right)^{1/3} G\frac{\nabla_{{\rm ad},c}\rho_c^{4/3}}{P_c} m^{2/3}\right] & \nabla_{\rm ad} < \nabla_{\rm rad} \end{cases} \tag{22}$$

where the subscript c denotes values at the center. For a derivation of the above relations, see Kippenhahn *et al.* (2012, Chapter 11).

3.2.2. *Surface Boundary*

Similar to the approach taken in Section 3.2.1, we can use Hansen *et al.* (2004) to determine initial estimates of the stellar luminosity and radius, L_* and R_*. Using the Eddington graybody approximation, we can define the surface as the mass shell located at an optical depth of $\tau = 2/3$. Using that surface, and assuming that enclosed mass and luminosity vary slowly with position, we can find:

$$R(M_*) \equiv R_* \tag{23}$$

$$L(M_*) \equiv L_* \tag{24}$$

$$T(M_*) \approx T_{eff} \equiv \left(\frac{L*}{4\pi\sigma_{SB}R_*^2}\right)^{1/4} \tag{25}$$

$$P(M_*) \approx \frac{GM_*}{R_*^2}\frac{2}{3}\frac{1}{\kappa} \tag{26}$$

where σ_{SB} is the Stefan Boltzmann constant, $\sigma_{SB} \equiv \frac{ac}{4}$. Once again, a more detailed derivation and discussion can be found in Kippenhahn *et al.* (2012).

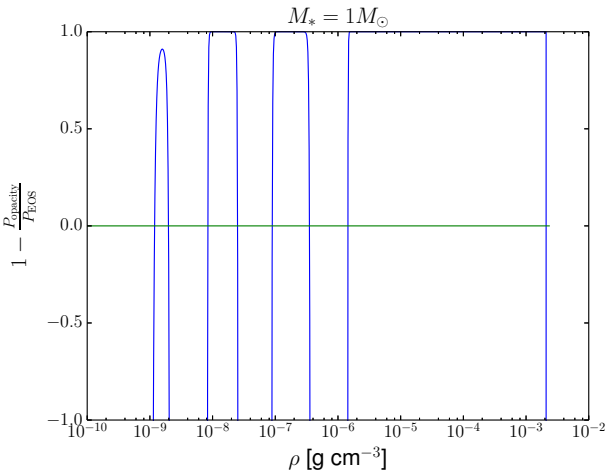

FIG. 1.— Visualization for determining the surface density through numerical root finding. The blue line is the difference between the pressure given by the equation of state (Equation 5), and the surface condition (Equation 26). A valid solution (*root*) is any ρ where the numerically calculated curve crosses the x-axis (the green horizontal line). Notice for solar values ($M_* = 1M_\odot$), there are multiple densities which satisfy the surface boundaries – the solution is non-unique.

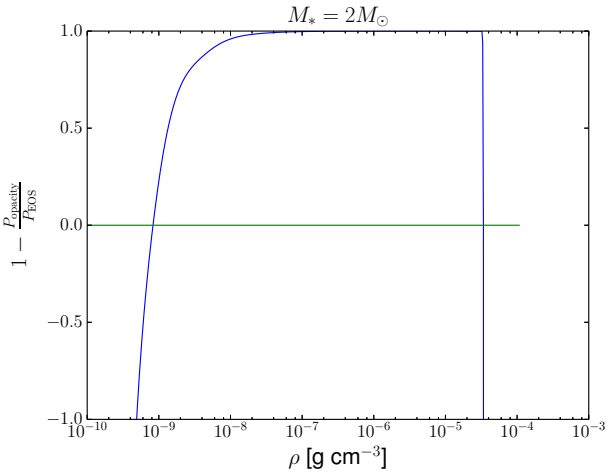

FIG. 2.— Same as in Figure 1, but now for $M_* = 2M_\odot$. Notice that the solution is significantly more unique.

While that process might seem straightforward, it is complicated because neither opacity, nor density are known. This wasn't a problem for the core, where pressure and temperature were known, which are needed for the equation of state (Equation 5). For this model, a numerical root finder was employed (`optimize.brentq` of the Python package `Scipy`) to find a density which produced equal pressures from the equation of state (Equation 5) as well as the surface pressure condition (Equation 26) for the corresponding opacity.

It should be noted that, depending on the choice of stellar mass and metallicity, the numerically determined surface density, opacity and pressure might not be unique. In fact the choice of a $M_* = 2M_\odot$ star was motivated by the uniqueness of the surface density (and therefore the uniqueness of the surface opacity and pressure). For a visualization of this non-uniqueness problem, see Figures 1 and 2.

3.3. *Numerical Integration*

Using Equations 1 - 4, we can integrate from the boundary conditions determined from Hansen *et al.* (2004) and Section 3.2, towards a central fitting point, m_f. Rather than write a custom integrator, I made use of the `optimize.odeint` function within the Python package `Scipy`. `odeint`, based on `lsoda` of the FORTRAN library `odepack` switches between solvers, depending on the stiffness of the system of differential equations. This ability, paired with built-in adaptive step sizing, makes it a good option for these integrations, where the system changes rapidly at the bounds, but then becomes significantly stable away from the boundaries. Extra computational power can be spent resolving boundary behaviors, which can be made up the time saved taking larger, simpler steps further from the bounds.

3.4. *Iterating to Convergence*

While the boundary conditions given by Hansen *et al.* (2004) for a star of a given mass are useful guesses, they do not generally produce self-consistent results using the physics included in Section 2. Typically, when integrating from the boundaries to a fitting point, the inward and outward integrations do not quite agree. Using the level of disagreement, we can determine how we should change our boundary conditions. This entire process can be iterated until the solution converges.

3.4.1. *Iteration Algorithm*

In order to iteratively improve the input boundary conditions, we applied an adapted multi-dimensional Newton-Raphson method. Treating our input boundary conditions as a 4-dimensional vector:

$$\mathbf{x} = \left\{ \frac{R_*}{R_{*,0}}, \frac{L_*}{L_{*,0}}, \frac{P_c}{P_{c,0}}, \frac{T_c}{T_{c,0}} \right\} \tag{27}$$

which has been scaled relative to the values suggested by Hansen *et al.* (2004), denoted with a 0 subscript.

Similarly, we can define the dependent variable as the (non-dimensionalized) fraction error at the fitting point between the inward and outward integrations:

$$\mathbf{y} = \left\{ \frac{r^+ - r^-}{r^-}, \frac{L^+ - L^-}{L^-}, \frac{P^+ - P^-}{P^-}, \frac{T^+ - T^-}{T^-} \right\} \tag{28}$$

where the $+$ superscript denotes the value at the fitting point for the integration from the surface to the fitting point, and the $-$ superscript likewise denotes the integrate from the center.

Given a set of boundary conditions, $\mathbf{x}_i$, we can determine $\mathbf{y}_j$. The challenge is to determine the $\mathbf{x}_i$ which will produce the minimal $\mathbf{y}_j$. The Newton-Raphson method provides an iterative approach:

$$x_i^{(n+1)} = x_i^{(n)} + \left(\frac{\partial y_j}{\partial x_i}^{(n)} \right)^{-1} y_j^{(n)} \tag{29}$$

for the conditions of the iteration n providing the boundary conditions for iteration $(n+1)$. While in the simple case of $\mathbf{y}$ linear in $\mathbf{x}$, this should converge within the first iteration, we generally find that y is non-linear, complicating the process.

Furthermore, given the numeric nature of the product, we don't typically have a closed form expression for the tensor $\partial_i \mathbf{y}_j$, let alone its inverse. With that constraint, we can use the *secant approximation* to the Newton-Raphson method:

$$\frac{\partial y_j}{\partial x_i} \approx \frac{y_j(\mathbf{x} + h\hat{\mathbf{e}}_i) - y_j(\mathbf{x})}{(x_i + h) - x_i} \qquad h = 0.04 \tag{30}$$

where h was arbitrarily chosen as .04 and could be chosen as any similar small number.

Iterative solutions were found until reaching a convergence criterion, such that there was not more than 1% error between inward and outward integrations between any integrated value:

$$\max\left(|y_i|\right) < 0.01\% \tag{31}$$

The only significant alteration to the Newton-Raphson method (with the secant approximation) is that the Newton-Raphson method tends to over-predict the stability of the system with respect to changes in the initial conditions (for instance, suggesting that luminosity be decreased by 80% within one integration). Many effects, such as the strong temperature dependence of the nuclear energy production rate ϵ, produce highly sensitive, non-linear behavior in our numerical integration. In order to increase the robustness of my algorithm, I choose relatively small step sizes ($h \approx 0.04$), and only changed my initial parameters 1/4 of the amount proposed by the Newton-Raphson method:

$$x_i^{(n+1)} - x_i^{(n)} = \frac{1}{4}\left(dx_{i,\mathrm{N-R}}\right)^{(n)} = \frac{1}{4}\left(\frac{\partial y_j}{\partial x_i}^{-1} y_j \right)^{(n)} \tag{32}$$

This ensured that the Newton-Raphson method would not over-shoot into an unstable region of parameter space, but did so at the cost of slower convergence (and thus increase runtime) for the same desired accuracy compared to the standard Newton-Raphson method.

3.4.2. *Parallelization*

During this convergence process, a majority of the runtime went into numerically integrating from the bounds to the fitting point. Preliminary tests showed that this was largely due to the overhead incurred by calling multiple nested functions to calculate the derivatives of Equations 1 - 4. As those multiple function calls were largely due to my modular package design, I chose not to significantly alter the structure of my package, and instead achieve time savings through parallelizing my code.

In particular, I choose run parallelize my numerical integrations, so that the inward and outward integrations would run concurrently, approximately cutting my runtime in half. Choosing the integrations as the part to parallelize was a natural choice; they were two integrations which were expect to take roughly equal time, could be run completely independently, and were the slowest element of my code.

The implementation of the parallel processing was done using the `multiprocessing` Python package. Specifically, the functions `Pool` and `apply_async` made things simple and painless.

	R_* $[R_\odot]$	L_* $[L_\odot]$	$\log P_c$ [dyne cm^{-2}]	$\log T_c$ [K]
Initial	1.48	18.3	17.2	7.32
Converged	1.70	15.2	17.2	7.31
`EZ-Web`	1.60	16.2	17.3	7.32

TABLE 1
BOUNDARY CONDITIONS. *Initial* REFERS TO THE VALUES SUGGESTED BY HANSEN *et al.* (2004); *Converged* REFERS TO THE VALUES FOUND TO PRODUCE CONSISTENT INTEGRATIONS TO FITTING POINT AT $m_f = 0.5M_*$. *`EZ-Web`* REFERS TO THE MODEL PRODUCED USING THE CODE OF (TOWNSEND, SEE SECTION 4.1).

	$\frac{\Delta r}{r}$	$\frac{\Delta \ell}{\ell}$	$\frac{\Delta P}{P}$	$\frac{\Delta T}{T}$
Initial	-0.118	-0.281	0.354	0.182
Converged	-0.001	-0.006	-0.0004	-0.003

TABLE 2
FRACTIONAL ERRORS AT THE FITTING POINT, $m_f = 0.5M_*$. CONVERGENCE WAS DEFINE BY THE BOUNDARY CONDITIONS THAT YIELDING LESS THAN 1% ERROR FOR ALL INTEGRATED VARIABLES. *Initial* REFERS TO THE ERRORS RESULTING FROM THE BOUNDARY VALUES SUGGESTED BY HANSEN *et al.* (2004); *Converged* REFERS TO THE ERRORS RESULTING FROM THE BOUNDARY VALUES WHICH SATISFIED THE CONVERGENCE CRITERION.

4. RESULTS

Following approximately 20 iterations (as laid out in Section 3.4), my model converged on initial conditions that resulted in errors smaller than 1% at the fitting point. The boundary conditions for this solution may be found in Table 1. The corresponding residual errors remaining after convergence may be found in Table 2. Graphical results of the integrations may be found in Figures 3 and 4.

4.1. *Comparison to other results*

The physics in our model necessarily neglected certain effects, in particular by using simplified expressions to encompass convective mixing, composition gradients in the star, and zones of partial ionization near the surface.

To compare the reliability of our model, with its assumed physics, we compared against multiple sources. As already discussed, the boundary conditions suggested by Kippenhahn *et al.* (2012) did not lead to a consistent structure model with our included physics.

We also compared our stellar structure model to the `EZ-Web` code of Townsend. The results of the `EZ-Web` code, for a star of the same mass and metallicity, can be seen in Figure 4. There is significant agreement between the `EZ-Web` structure model and my structure model. There is also significant agreement between the total scale of the boundary conditions (see Table 1).

5. CONCLUSION

Using Equations 1 - 4, we integrated the boundary conditions suggested by Hansen *et al.* (2004). Given the numerical instability at both boundaries, stellar core and surface, we simultaneously integrated from the surface to a midway fitting point and from the core to the same fitting point. Generally, the values of both integration do not agree at the fitting point. Using an adapted Newton-Raphson method, we varied the input boundary conditions, until both inward and outward integrations converged at the fitting point.

In order to implement this numerical process, we used an empirical opacity table. The range of this opacity table was limited; in order to avoid leaving the range of this table, we were restricted to stars $M_* \gtrsim 2M_\odot$.

In order to make this code run in a reasonable amount of time, some optimizations were required. The biggest time cost was due to the modular design of this package. A large number of nested function calls are taken within each step of the numerical integration, incurring a significant overhead cost. This was partially mitigated through harnessing multiple cores, using parallel programming. Access to more cores might allow further increases in speed, but the biggest improvement would probably come through refactoring the code entirely, to reduce function call overhead.

Overall, the integrations were successful, given the physics included. When compared to observational data, the model agreed within about 15%, but this model agreed to other models within 5-10%. The existence of these errors is probably a combination of numerical and physical systematics. For instance, we modeled a *Zero-Age Main Sequence* star, whereas real stars have a finite age, which can be difficult to constrain. As a star ages, we expect its boundary conditions and structure to change, providing a source of error relative to our model. And even at the *Zero-Age Main Sequence* our model is unable to resolve all of the physics. We could model the most significant effects, but we could not resolve composition inhomogeneities, or properly treat convection. Such physics is certainly present, but go beyond the scope of this work.

REFERENCES

C. J. Hansen, S. D. Kawaler, and V. Trimble, *Stellar interiors : physical principles, structure, and evolution, 2nd ed., New York: Springer-Verlag* (2004).

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

C. Angulo, M. Arnould, M. Rayet, P. Descouvemont, D. Baye, C. Leclercq-Willain, A. Coc, S. Barhoumi, P. Aguer, C. Rolfs, R. Kunz, J. W. Hammer, A. Mayer, T. Paradellis, S. Kossionides, C. Chronidou, K. Spyrou, S. degl'Innocenti, G. Fiorentini, B. Ricci, S. Zavatarelli, C. Providencia, H. Wolters, J. Soares, C. Grama, J. Rahighi, A. Shotter, and M. Lamehi Rachti, Nuclear Physics A **656**, 3 (1999).

P. D. Parker, J. N. Bahcall, and W. A. Fowler, ApJ **139**, 602 (1964).

N. R. Badnell, M. A. Bautista, K. Butler, F. Delahaye, C. Mendoza, P. Palmeri, C. J. Zeippen, and M. J. Seaton, MNRAS **360**, 458 (2005).

R. Townsend, "Ez-web," `http://www.astro.wisc.edu/~townsend/static.php?ref=ez-web`, accessed: 2014-12-09

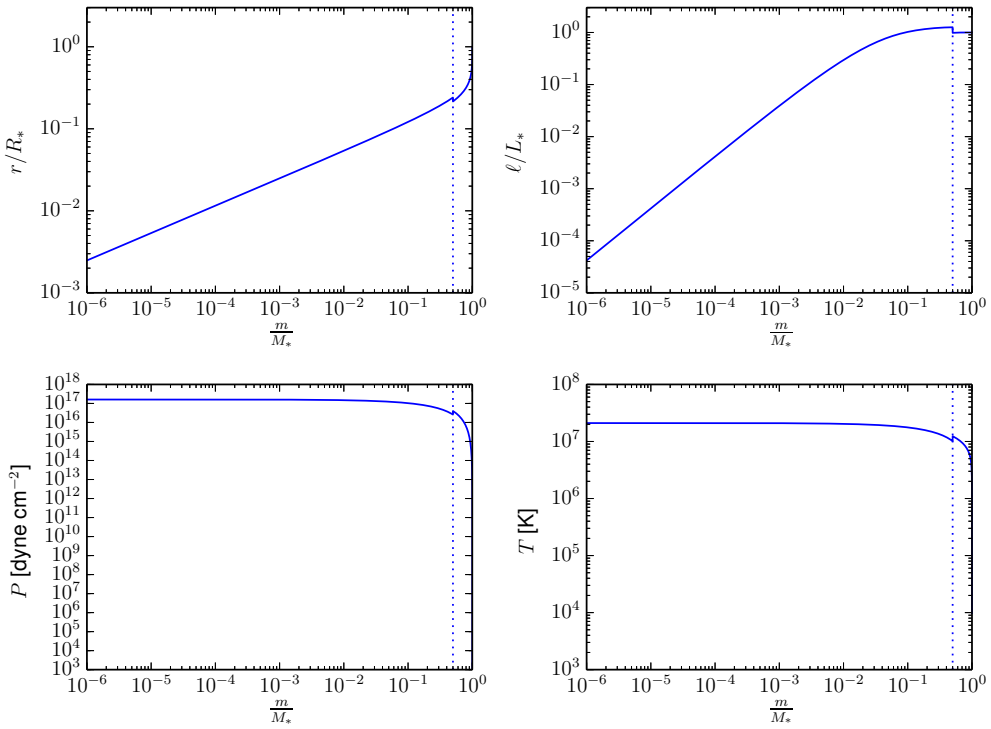

FIG. 3.— Results of numerically integrating Equations 1 - 4 with the initial boundary conditions in Table 1, as suggested by Hansen *et al.* (2004). Fitting point, $m_f = 0.5M_*$ noted by vertical dashed line. Note the discontinuities at the fitting point.

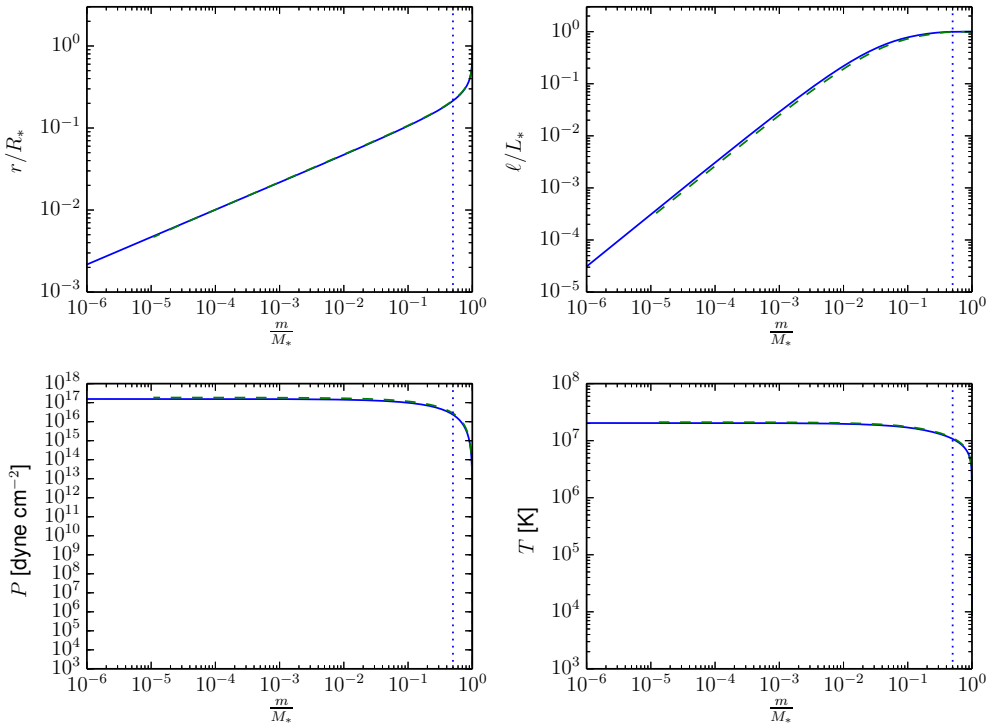

FIG. 4.— Same as in Figure 3, but now the converged boundary conditions listed in Table 1. Fitting point, $m_f = 0.5M_*$ noted by vertical dashed line. The dashed represents the results of the `EZ-Web` code (see Section 4.1 and Townsend). The disagreement between our model and the `EZ-Web` model is virtually imperceptible.