`StarStruc.jl`: Stellar structure in Julia

Asher Wasserman

1 Introduction

This document presents a description of a static stellar structure model implemented in `Julia`, a high-performance dynamical programming language. The accompanying code can be found at https://github.com/adwasser/StarStruc.jl. For carefully considered initial guesses as to the boundary values, the code should converge on a structure model on the order of a few minutes. For arbitrary initial guesses, the code will likely crash and burn.

2 Physics

Stars are fearsome beasts, so we tame them by introducing some simplifying assumptions. We model stars as non-rotating, time-independent, uniform composition, spherically symmetric cows. Such domesticated animals still can pose numerical dangers to reckless star-cow wranglers, however.

2.1 Structure equations

We can concisely write the equations of stellar structure as

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

$$\frac{\partial \ell}{\partial m} = \epsilon_n \tag{2}$$

$$\frac{\partial P}{\partial m} = -\frac{Gm}{4\pi r^4} \tag{3}$$

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

where $r(m)$ is the radius, $\ell(m)$ is the luminosity, $P(m)$ is the pressure, and $T(m)$ is the temperature. Our goal is to solve for these four structure function for a specified total mass, M, and composition, parameterized as hydrogen mass fraction, X, and helium mass fraction Y. What is then left is to specify the equation of state, $\rho(P, T)$, the energy generation per unit mass ϵ_n, the temperature gradient, $\nabla = \partial \log T / \partial \log P$, and the opacity, $\kappa(\rho, T)$.

We assume that the star is a fully ionized ideal gas, so its equation of state is

$$P(\rho, T) = \frac{\rho}{\mu m_H}kT + \frac{a_{\rm rad}}{3}T^4 \tag{5}$$

where

$$\mu = \frac{1}{1 + 3X + Y/2} \tag{6}$$

is the dimensionless mean molecular weight. The term on the right accounts for pressure due to radiation and is generally small compared to the gas pressure.

For simplicity, we model energy transport such that convection efficiently transports any energy beyond the adiabatic limit, and so

$$\nabla = \frac{T}{P}\frac{\partial T}{\partial P} = \begin{cases} \nabla_{\mathrm{ad}} & \nabla_{\mathrm{rad}} \geq \nabla_{\mathrm{ad}} \\ \nabla_{\mathrm{rad}} & \nabla_{\mathrm{rad}} < \nabla_{\mathrm{ad}} \end{cases} \tag{7}$$

For our assumed ideal gas, $\nabla_{\mathrm{ad}} = 0.4$, and for a diffusive radiative transport model,

$$\nabla_{\mathrm{rad}} = \frac{3}{16\pi acG}\frac{\kappa \ell P}{mT^4} \tag{8}$$

We only model hydrogen burning for energy generation. Hydrogen burning occurs through the proton-proton chain and the CNO cycle, both of which have their own temperature dependencies. In the weak electron shielding approximation, we have

$$\epsilon_{\mathrm{pp}} = 2.5 \times 10^4 \psi g_{11} \rho X_1^2 T_9^{-2/3} \exp\left(-\frac{3.381}{T_9^{1/3}}\right) \; \mathrm{erg}\;\mathrm{s}^{-1}\;\mathrm{g}^{-1} \tag{9}$$

$$\epsilon_{\mathrm{CNO}} = 8.24 \times 10^{25} g_{14,1} X_{\mathrm{CNO}} X_1 \rho T_9^{-2/3} \exp\left(-15.231 T_9^{-1/3} - \left(\frac{T_9}{0.8}\right)^2\right) \; \mathrm{erg}\;\mathrm{s}^{-1}\;\mathrm{g}^{-1}$$

where $T_9 = 10^{-9} \times (T/K)$. Here, ψ accounts for the different branches of the proton-proton chain, and is generally between 1 and 1.5. The polynomial terms g_{11} and $g_{14,1}$ are given by

$$g_{11} = 1 + 3.82 T_9 + 1.51 T_9^2 + 0.144 T_9^3 - 0.0114 T_9^4 \tag{10}$$
$$g_{14,1} = 1 - 2.00 T_9 + 3.41 T_9^2 - 2.43 T_9^3$$

For the CNO mass fraction, X_{CNO}, we assume that carbon, nitrogen, and oxygen make up 70% of the metal mass fraction.

The last missing piece for completely specifying the derivatives is a relevant opacity model for stellar matter. Here we use the OPAL opacity tables, taken from `http://opalopacity.llnl.gov/`, and interpolate between values of ρ and T.

2.2 Boundary conditions

With all of the above, we have four coupled first-order differential equations. To solve for our four structure profiles, we need to impose some physically meaningful boundary conditions. Starting from a mass shell very close to the center, m_c, we can use the linearizations of equations 1 and 2 to get

$$r_c = \left(\frac{3}{4\pi\rho_c}\right)^{1/3} m_c^{1/3} \tag{11}$$
$$\ell_c = \epsilon_n m_c$$

At the surface, we define our photosphere to be at the radius, R_s, where we have an optical depth of 2/3. Then from hydrostatic equilibrium and the definition of the effective temperature, we have

$$P_s = \frac{GM}{R_s^2}\frac{2}{3}\frac{1}{\kappa} \tag{12}$$
$$L_s = 4\pi R_s^2 \sigma T_s^4$$

Since we only have four boundary condition equations with eight boundary values, we will have to make educated guesses for the other four, and evaluate how well we do in creating the four profiles. For our purposes, we will propose the boundary values of R_s, L_s, P_c, and T_c, then use the relations above to create the full set of boundary values.

3 Numerics

We follow the strategy of shooting to a fixed point to solve the boundary value problem posed by the section above. We choose a fixed mass shell at some fraction of the total mass. We then iteratively propose a new set of boundary values, $\vec{y} = (R_s, L_s, P_c, T_c)$, and use these along with our boundary conditions to integrate the structure equations both outward from the center and inward from the surface, both ending at the fixed point. We implement a fourth-order Runge-Kutta method for performing these integrations.

We define the mismatch function as

$$\vec{f}(\vec{y}) = (r_{\mathrm{in}}(m_f) - r_{\mathrm{out}}(m_f),\ \ell_{\mathrm{in}}(m_f) - \ell_{\mathrm{out}}(m_f),\ P_{\mathrm{in}}(m_f) - P_{\mathrm{out}}(m_f),\ T_{\mathrm{in}}(m_f) - T_{\mathrm{out}}(m_f)) \quad (13)$$

The mismatch function thus maps the proposed boundary values, $\vec{y}$, to the set of difference between the inward and outward integrations at the fixed point, m_f. Finding a solution to the boundary value problem amounts to finding the zero of the mismatch function.

We approach this task by using a modified Newton-Raphson descent method. We look at the linearization of $\vec{f}$ about y,

$$\vec{f}(\vec{y} + \delta\vec{y}) \sim \vec{f}(\vec{y}) + J\delta\vec{y} \quad (14)$$

Here, J is the Jacobian of $\vec{f}$, defined as

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial y_1} & \cdots & \frac{\partial f_1}{\partial y_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial y_1} & \cdots & \frac{\partial f_n}{\partial y_n} \end{pmatrix} \quad (15)$$

We approximate the Jacobian through the finite difference method. For each proposed boundary value, y_i, we can perturb the initial proposed vector to $\vec{y}_{\Delta_i} = \vec{y} + \Delta y_i \hat{i}$, where Δ is some small fraction. The i-th column of J is then given by

$$\vec{J}_i = \frac{\vec{y}_{\Delta_i} - \vec{y}}{\Delta y_i} \quad (16)$$

If we wish to find the zero of $\vec{f}$, we need to choose the step $\delta\vec{y}$ such that $f(\vec{y} + \delta\vec{y}) \to 0$. This leads to the proposed step

$$\delta\vec{y} = -J^{-1}\vec{f}(\vec{y}) \quad (17)$$

Our next proposed set of boundary values is then given just by $\vec{y} + \delta\vec{y}$. However, given that we have used only the linear term in the Taylor expansion of $\vec{f}$, this method will only work if we start very close to the true value of $\vec{y}$. In practice, we will start with some poor guess for $\vec{y}$ for which the Newton-Raphson step will push us too far, leading to wild oscillations. In order to create a chance for convergence, we down-weight the step $\delta\vec{y}$ by an amount such that $\max_i |\delta y_i / y_i| < 0.1$, and only allow for the full step once all changes are within 10% of their previous value.

4 Results

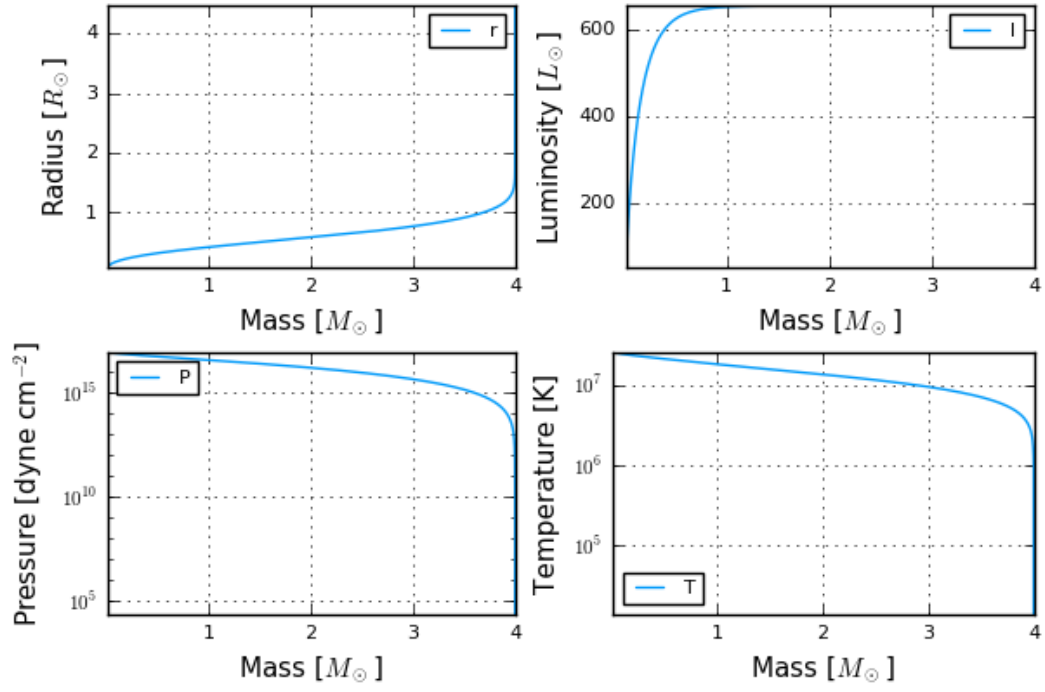| Value	R_s	L_s	$\log_{10} P_c$	$\log_{10} T_c$
Units	$R_\odot$	$L_\odot$	dyne cm^{-2}	K
StarStruc	4.47	657.43	16.94	7.41
MESA	3.39	318.24	16.86	7.43

Table 1: Comparison of model boundary values with inputs from MESA.

Figure 1: Convergent model for 4 $M_\odot$.

5 Conclusions