Stellar Structure Calculation Instructions

STELLAR STRUCTURE AND EVOLUTION (SPRING 2023)

Your task is to write your own program(s) in your language of choice to calculate the zero-age main sequence (ZAMS) structure of a star of your choosing. You are free to use existing code for the heavily numerical parts of your calculation (e.g., SciPy). You must then compare the structure predicted by your software to the ZAMS structure predicted by the state-of-the-art open source stellar structure and evolution software Modules for Experiments in Stellar Evolution (MESA).

1. Choose values for the star's mass M_*, hydrogen mass fraction X, helium mass fraction Y, and metal mass fraction $Z = 1 - X - Y$.

2. Obtain the appropriate opacity table for your selected composition from the OPAL or Opacity Project (OP) websites (possibly using the Internet Archive Wayback Machine for the former). Note that both OPAL and OP tables are indexed by the parameter R (and not ρ) and that the lowest temperatures T in both tables are $\log_{10}(T/\mathrm{K}) = 3.75$. Consult Alexander & Ferguson (1994) if your calculation requires opacities at lower temperatures.

3. Write your own function (possibly calling an existing interpolation routine) to interpolate the opacity table for specific values of density ρ and T in $\log_{10} \rho$ and $\log_{10} T$ space. Your function should work over the range $-9 < \log_{10}(\rho/\mathrm{g\ cm^{-3}}) < 3$ and $3.75 < \log_{10}(T/\mathrm{K}) < 7.5$.

4. Write your own function to calculate the rate of energy generation ϵ in erg $\mathrm{g^{-1}}$ $\mathrm{s^{-1}}$ from hydrogen fusion given density, temperature, and composition accounting for both the proton–proton chains and CNO cycles assuming the reactions have reached equilibrium. You should use the approximations given in Chapter 18 of Stellar Structure and Evolution (Second Edition). Be sure to use weak screening for f_{11}.

5. Write your own function to calculate density given pressure P, temperature, and composition (i.e., the equation of state) assuming complete ionization. Be sure to include the effect of radiation pressure.

6. Read Sections 18.0, 18.1, and 18.2 of Numerical Recipes, as it will be necessary to implement the method of shooting to a fitting point (their `shootf` function) to solve the boundary-value problem posed by the four coupled ordinary differential equations of stellar structure and evolution. You will also need a program similar to the `newt` function described in Section 9.7 of Numerical Recipes or in the SciPy documentation to repeatedly call a `shootf`-like function, calculate updated boundary values, and ultimately obtain a converged solution. Faster convergence can usually be obtained by using fractions of the updates suggested by a `newt`-like function. Ordinary differential equation (ODE) solvers like those described in Sections 17.0, 17.1, and 17.2 of Numerical Recipes or the SciPy documentation are also critical.

7. The calculation of stellar structure using the shooting method described in the references above requires outward integrations to start just outside the center of the star due to the singularities that occur at $m = r = 0$. Read Chapters 11.1 and 11.2 of Stellar Structure and Evolution (Second Edition) for instructions on how to handle the non-trivial inner and outer boundary conditions necessary for the use of the shooting method.

8. Write your own functions corresponding to `load1` and `load2` in the documentation for `shootf` described above. The function `load1` should give the values of the four dependent variables l, P, r, and T at a mass point m very slightly away from the center of the star for use as initial values of outward integrations. The function `load2` should give the same four dependent variables at the surface for use as initial values of inward integrations. Note that the central pressure $P_{\rm c}$, central temperature $T_{\rm c}$, total luminosity L_*, and total radius R_* are parameters for which you must provide an initial guess.

9. Write your own function corresponding to `derivs` in the documentation for `shootf` described above. The function `derivs` should take the independent variable m and the four dependent variables l, P, r, and T and return the derivatives dl/dm, dP/dm, dr/dm, dT/dm given by

$$\frac{dl}{dm} = \epsilon, \tag{1}$$

$$\frac{dP}{dm} = -\frac{Gm}{4\pi r^4}, \tag{2}$$

$$\frac{dr}{dm} = \frac{1}{4\pi r^2 \rho}, \tag{3}$$

$$\frac{dT}{dm} = -\frac{GmT}{4\pi r^4 P}\nabla. \tag{4}$$

10. Write a driver program/script analogous to `newt` to repeatedly call a `shootf`-like function, calculate updated boundary values, and ultimately obtain a converged solution.

11. Read the documentation for the latest version of MESA and install MESA after first installing its SDK. Modify the default stellar model located in `mesa/star/work/` so that it corresponds to the same star for which you have calculated its ZAMS structure. Use MESA to predict the ZAMS structure of that star for comparison with your own results.

12. The written report on the outcome of your calculation is **as important** as the accuracy of your calculation. It should include

    (a) a description of the physical problem;

    (b) a statement of parameters used;

    (c) a description of the numerical methods used and the quality of their convergence;

    (d) a machine-readable table with Lagrangian mass coordinate m, radius r, density ρ, temperature T, pressure P, luminosity l, nuclear energy generation rate ϵ, opacity κ, adiabatic temperature gradient $\nabla_{\rm ad}$, actual temperature gradient $\nabla = d\ln T/d\ln P$, and the convective/radiative nature of the shell;

    (e) an appropriate graphical presentation of the star's structure;

    (f) a link to a public Github repository with your code;

    (g) and a comparison with MESA's prediction for the stellar structure of star with identical properties.