Ay123 Stellar Model

Due Monday, December 4, 2017, 5pm

1. **Converging your model:** Read sections 17.0, 17.1, and 17.2 of the "old" Numerical Recipes (Press et al., 1992, 2nd edition). The PDF version of the book can be found online various places, including here:

   http://www.nr.com/oldverswitcher.html

   We will be using the method of shooting to a fitting point (Section 17.2). You can use the routine given there, `shootf` to perform one integration. You will need an overall control program (something like `newt`, Section 9.7) to do multiple calls to `shootf`, to calculate corrections to the assumed boundary values, and to eventually obtain a converged solution.

   Modify the programs for your own purposes. For example, it is generally helpful to take only a fraction of the correction to the unknown parameters that the program calculates.

2. **Assignment:**

   When you hand in your assignment, you should include the following items.

   (a) Statement of parameters used, including stellar mass, composition, and location of your fitting point.

   (b) Statement of the numerical method, how it works, and how well it converged.

   (c) Graphical presentation of results. Include

      i. Plots of mass, pressure, density, temperature, and luminosity as a function of radius. All quantities should be in cgs units. Axes should be scaled (e.g., use logarithmic axes when necessary) so that quantities are clearly visible.

   (d) *Well commented*, complete code (Python, C, IDL, Fortran, etc.)

   (e) Comparison of your model with a MESA model using the same parameters. Construct the same plots listed above for the MESA model. Briefly describe any differences and why they might have occurred. See the advice section below.

3. **Advice:**

   (a) To create a model of arbitrary composition, you can use the command

   ```
   create_pre_main_sequence = .true.
   ```

   in the job section of your in list. In the controls section, use the commands

   ```
   initial_y = ##
   ```

   and

   ```
   initial_z = ## .
   ```

   If MESA has trouble converging, you may want to try the `relax_Y` and `relax_Z` commands described in the file

   `star/defaults/star_job.defaults` .

(b) For best comparison, you'll want to run MESA using only electron scattering opacity like you have done for your model. This can be done by setting `use_simple_es_for_kap = .true.` in the controls section of your inlist.

(c) MESA may have trouble if you choose a model with high metal content. It might be easiest to choose $Z < 0.04$ for comparison with MESA.

(d) IDL users: There may be an issue with double precision in IDL not being precise enough, depending on the platform, for the Newton-Raphson matrix inversions. However, the built-in IDL routine `NEWTON` has worked for people. If you write your own Newton-Raphson routine in IDL you may run into problems, or the code may only work on some platforms. This has not been a problem with other languages, as far as I know.

A voice from the past: "What I found was that my code worked on some machines, but not on others. This was probably due to the presence/absence of extended double precision. It works on my Mac, but not on all Linux boxes."

(e) "I would say Newton-Raphson is extremely finicky (not just in its IDL incarnation, but in general). Nothing will defeat it more effectively than bad guesses for the star's central pressure and temperature. You don't even have to be way off, just slightly off. I bet some of the students have codes that work, but they haven't found the right initial conditions."

"I recall that the convergence was quite finicky and my initial 'guesses' were fine-tuned by hand to be numbers that did not loop infinitely, but would allow for convergence."

(f) Your fitting point can be a mass shell at $m = 0.5M$, but it doesn't have to be. It could be $0.2M$ or something else, if that helps.

(g) A few bits that are somewhat related:

i. The increments used to get the numerical derivatives used in the matrix should not be too small (say 0.1% of the variable being incremented).

ii. Once the corrections are derived, they should not be used "full force." It sometimes means taking only a very small fraction of the calculated corrections to get the convergence process started. Maybe as small as 1%.

iii. Be careful of the step size used in the numerical integrations of the ODEs. For the inward integrations the step size (in mass) has to start out being very small. At the center this is not a problem.

iv. Particularly for the surface to core integration, adaptive step sizes can significantly speed up execution speed, although they are not required. Using an adaptive step size allows one to resolve the steep temperature gradient near the surface while still taking relatively large mass steps in the interior.