Title: March 17, 2012 Research/Programming Notes & Progress Copy
Date: March 17, 2012 5:26 PM
Category: Work
Tags: python, research, Henyey code, Bodenheimer code, finding initial conditions, from campus

March 17, 2012 5:26 PM

Location: from home
Computing context: Macho-Mac2

Continuing from last time:

Figuring out why the models under 0.5Msun never converge, regardless of timestep size, when fusion is turned off:

Try running the initial (polytrope-->model) single timestep for these lower mass models **with** fusion turned on. Then, see if you can take that initially converged model (which is no longer in any way, shape or form the initial condition that so set these models off the rails) and successfully evolve it forward in time **without fusion**.

(1) Proceedure for doing this

(2) (1) For the 0.3Msun case:

- (a) Run polytr.f on polytr03.inp (saving the result as polyout)
- (b) Edit pmsstar03.start, setting
  - The 'write to file' filename to... 0.3MsunWithFusion.mod. Or something like that.
  - EFRC = 1.0 (i.e., turning fusion back on)
  - NTES = 5 (if it isn't already, just so we can compare the convergence behavior of the successful case to the ones above)
  - dTIM (and everything else on that line) to 1.00E+11
- (c) Run thecode.f on pmsstar03.start. If it fails to converge, try using a smaller timestep.

Don't go any father along this checklist until you get it set up so that it *does* converge.

(d) Run ./parse_output.pl on the (ascii) results. Also, copy over the .mod file to the results directory.

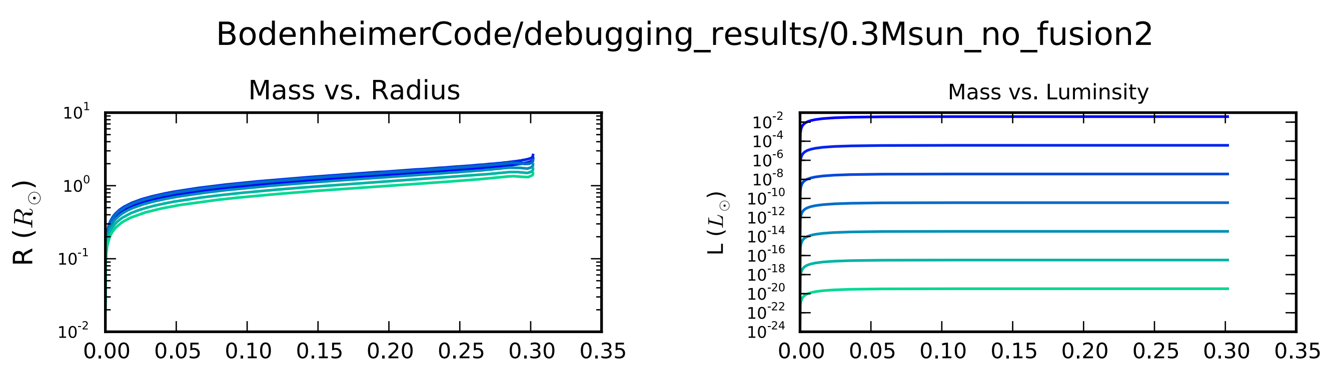
(e) Edit pmsstar03.start, setting

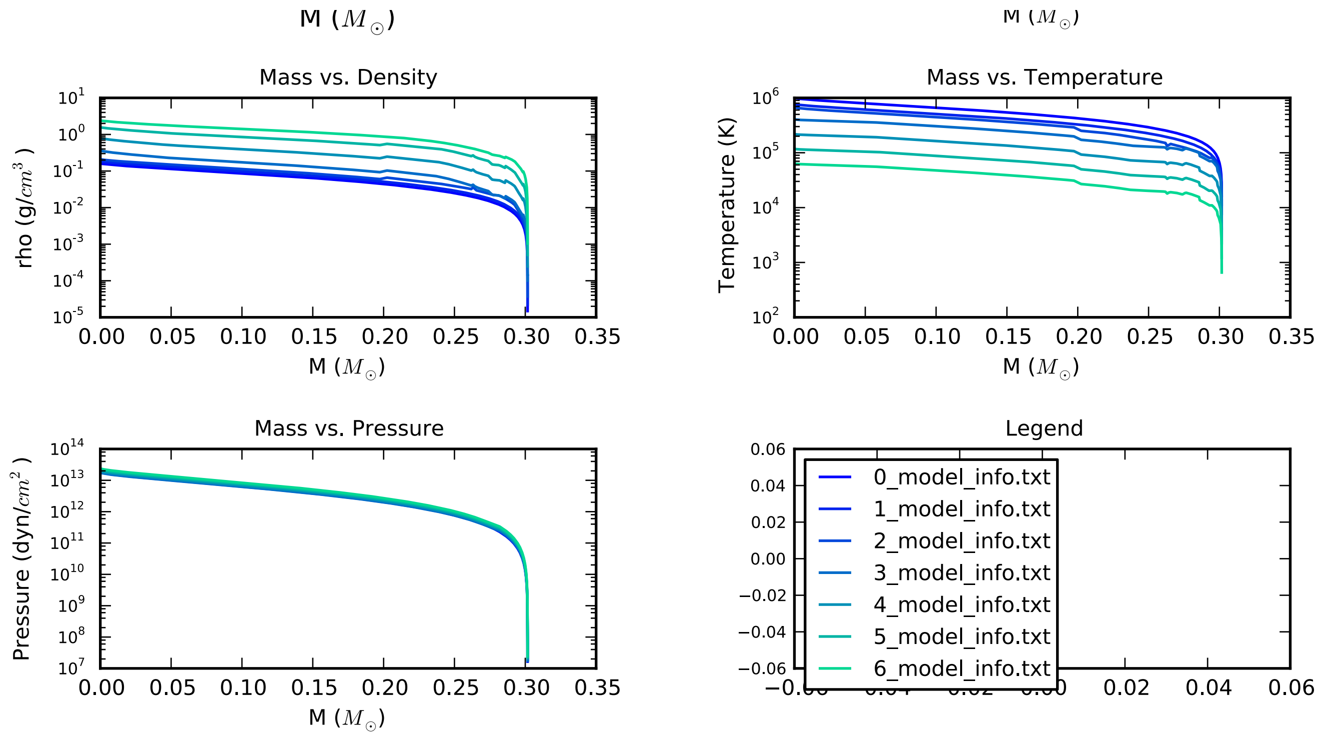
- The 'input file' to binary, name = 0.3MsunWithFusion.mod
- The 'write to file' filename to... 0.3MsunNoFusion.mod.
- EFRC = 0.0 (i.e., turning fusion off)
- NTES = 5 (if it isn't already, just so we can compare the convergence behavior of the successful case to the ones above)
- dTIM (and everything else on that line) to 1.00E+11

(f) Run thecode.f on pmsstar03.start.

(g) See if the model still fails to converge.

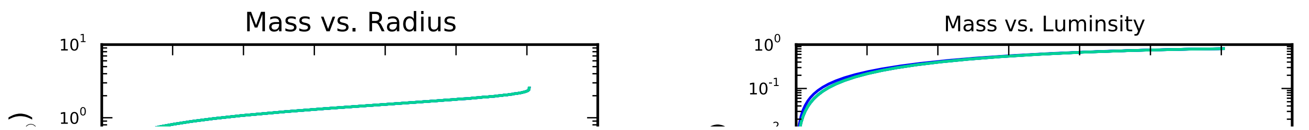
- Still fails to converge, but I discovered that Cwrk = 0.00 in pmsstar03NoFusion.start, so changed that to Cwrk = 1 (with Crad = 1, still), and am re-running the models. No wonder the luminosity profile was plummeting: no PdV work was being accounted for in the star's collapse.

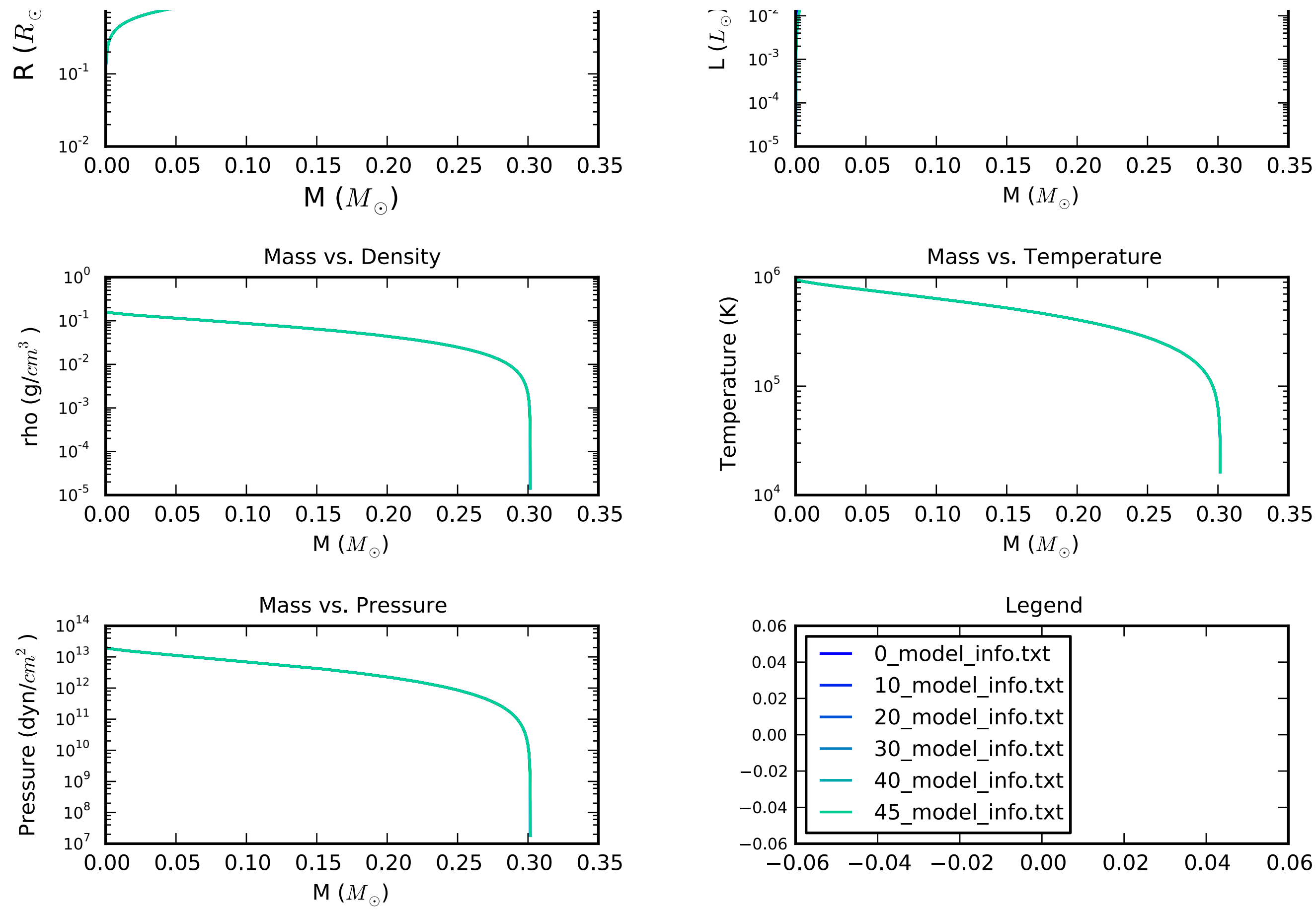




- Oh hey, look: it converges now that Cwrk = 1.00 ! (At least, when you start it from the output of the converged-single-timestep-with-fusion model.)

BodenheimerCode/debugging_results/0.3Msun_no_fusion3

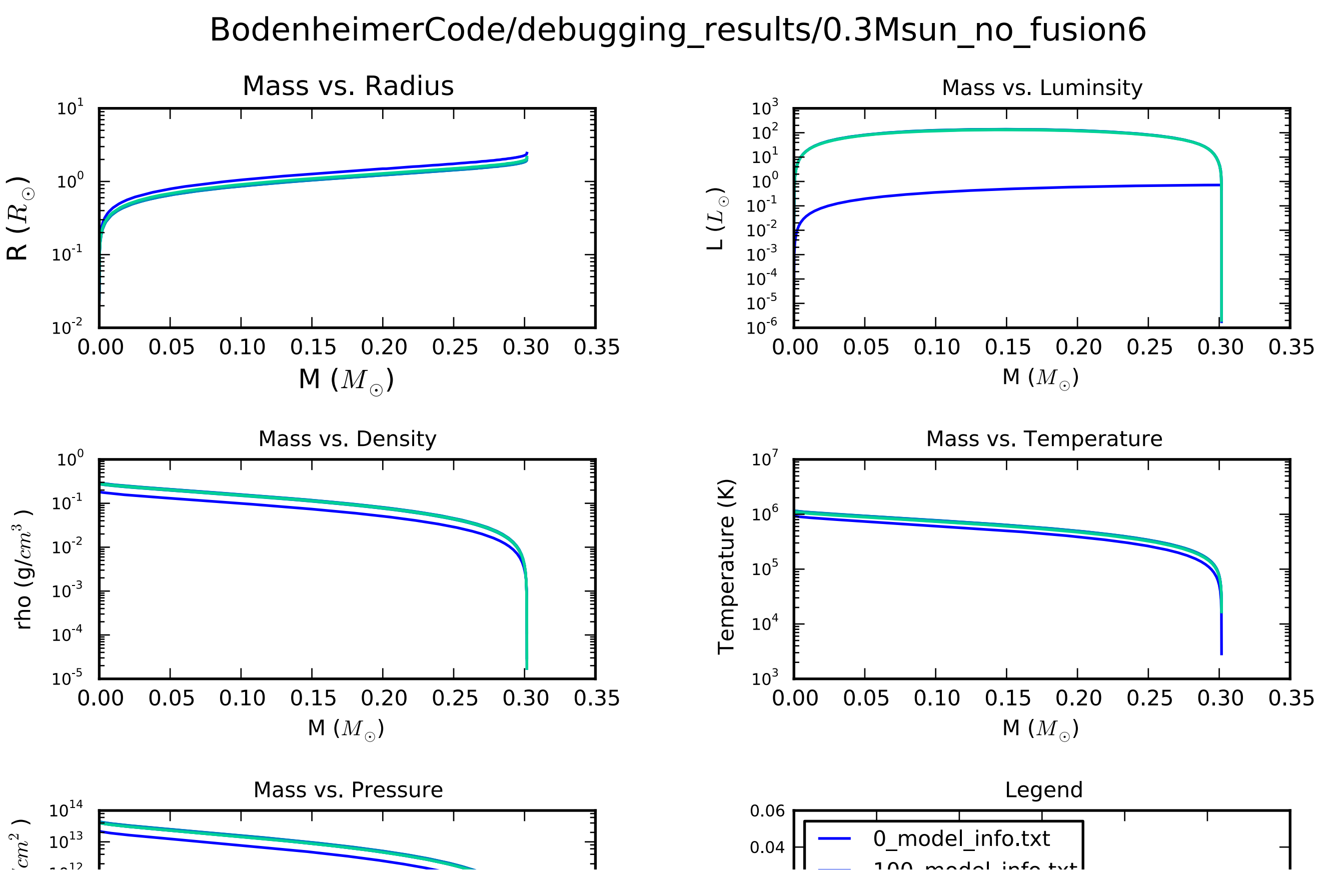


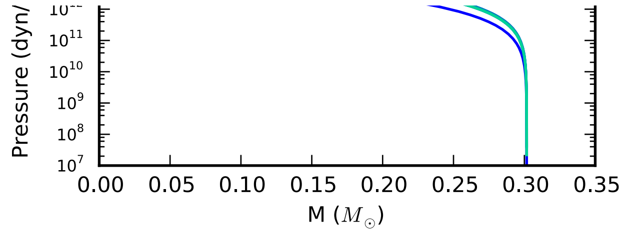


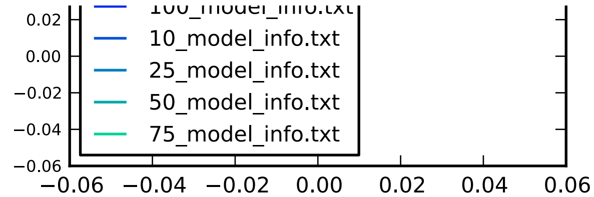
-
- - Doesn't converge when I start it from the n=3/2 polytrope initial conditions, with a timestep of 10^11 seconds. But, maybe it'll converge if I use a smaller timestep.

-

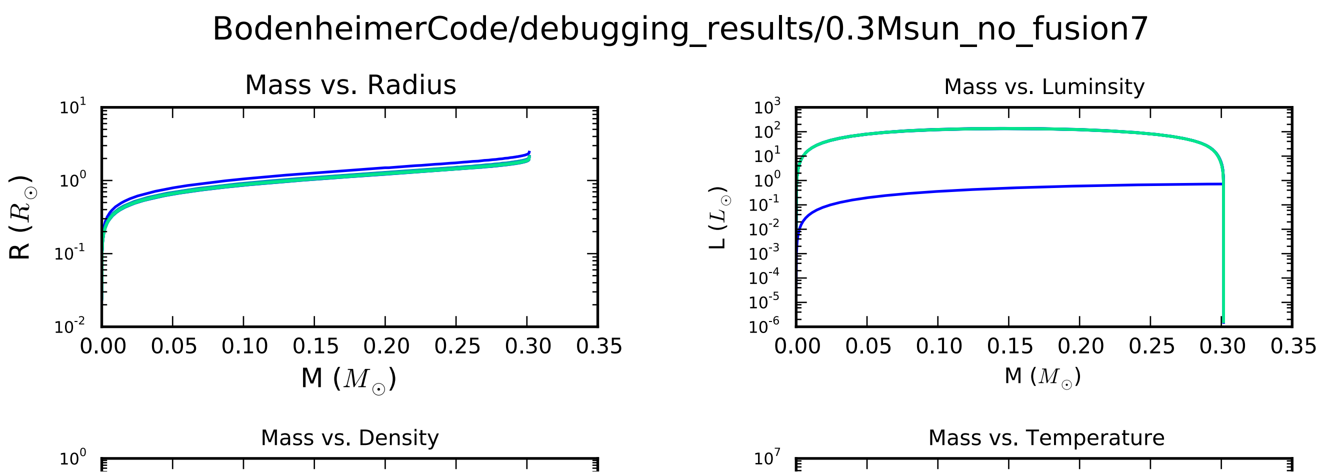
- - Nope, tried it with 10^9 second timestep, still didn't converge in the 1000 iteration loops available to it. (Would it converge if I let it go for 10000 loops, instead?)

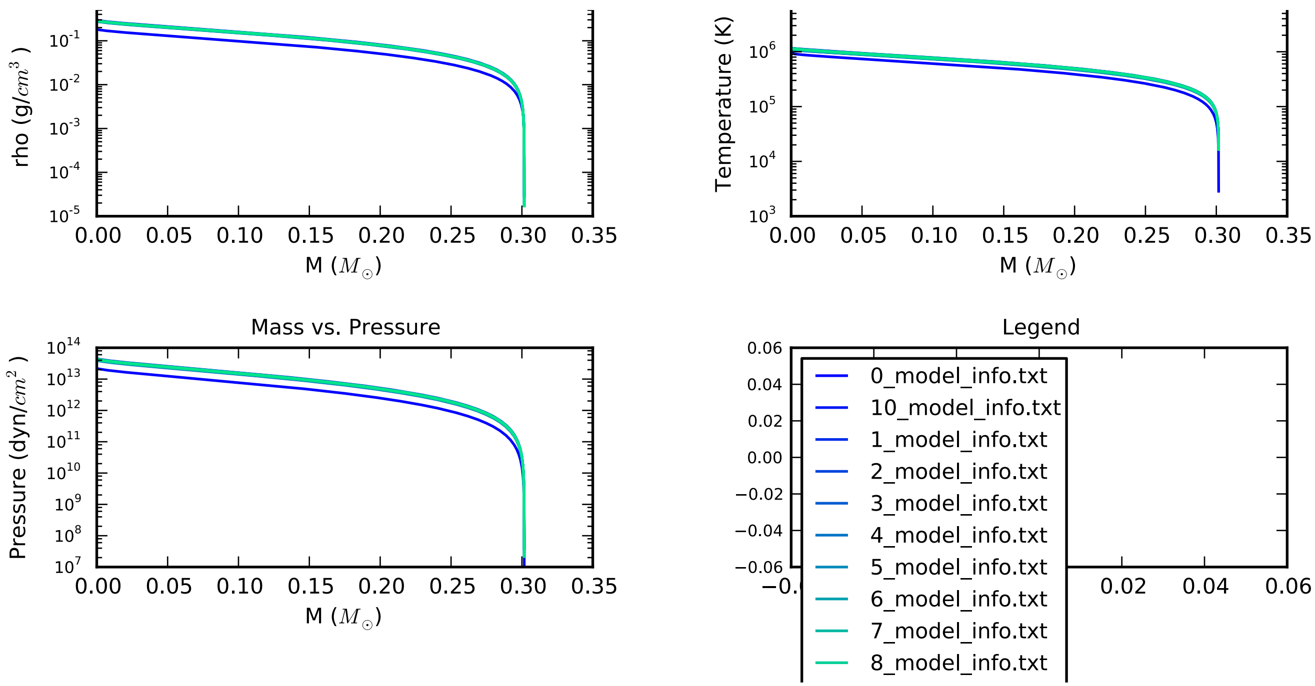






- Strange behavior where the luminosity falls off to ~zero at the outer boundary after ~500 to ~750 iterations.
- Maybe this behavior will change if I turn Crad completely off in the .start file?
    - Nope, still doesn't converge. Not surprising, really, since setting EFRC = 0 overrides any of the Crad settings, and effectively sets Crad = 0.





(4) (2) For the 0.4Msun case:

(a) Run polytr.f on polytr04.inp (saving the result as polyout)

(b) Edit pmsstar04.start, setting

- The 'write to file' filename to... 0.4MsunWithFusion.mod. Or something like that.
- EFRC = 1.0 (i.e., turning fusion back on)
- NTES = 5 (if it isn't already, just so we can compare the convergence behavior of the successful case to the ones above)
- dTIM (and everything else on that line) to 1.00E+11

(c) Run thecode.f on pmsstar03.start. If it fails to converge, try using a smaller timestep. Don't go any father along this checklist until you get it set up so that it *does* converge.

(d) Run ./parse_output.pl on the (ascii) results. Also, copy over the .mod file to the results directory.
(e) Edit pmsstar04.start, setting
    - The 'input file' to binary, name = 0.4MsunWithFusion.mod
    - The 'write to file' filename to... 0.4MsunNoFusion.mod.
    - EFRC = 0.0 (i.e., turning fusion off)
    - NTES = 5 (if it isn't already, just so we can compare the convergence behavior of the successful case to the ones above)
    - dTIM (and everything else on that line) to 1.00E+11
(f) Run thecode.f on pmsstar04.start.
(g) See if the model still fails to converge.

<u>After doing the stuff above, go back to the mass chain-down line of inquiry (from before) as listed below:</u>

- *See if using the 'mass chain-down' technique with the 0.5M$_{\odot}$ (no fusion) converged model as a starting point can produce converged models for lower mass (no fusion) balls of gas.*
    - *Implement a mass chain-down proceedure in thecode.f*
        - *Add a 'mass chaindown?' flag to the .start file, and modify thecode.f to be able to read it in*
        - *If the 'mass chaindown' = true,*
            - *read in a converged model*
            - *evolve it forward in time by 10(?) dTthresh steps*
            - *then decrease the mass of the system by some factor*
                - *(By what factor? How much or how little can you successfully decrease the mass at any given chain-down step? Need to think about this more once I get to this point...)*