Date & Time: Nov. 13, 2012

Location: Campus

Computing context: /Desktop/Research/CppHenyeyCode, /Desktop/Research/BodenheimerCode/UnalteredCode/

Continuing from last time:

I (eventually) figured out the stuff in "Notes 1" by comparing the dTk[1-4] values in Helena to their Peter counterparts. This by itself uncovered an important typo in my dTk4 equation (using M rather then Mtemp). That's fixed, and that certainly helps, but it's not the silver bullet for the atmos profiles problem. It did, however, clue me in to the fact that the dTk[1-4] values (all 4 of them, all the time-- not one individual one of them) hold the key to getting the right profiles out of this RK4 process.

The only quantity unique to the dTk[1-4] calculations is "nab," defined in Eqn (1) as:

$$\nabla \equiv \min\left(\nabla_{\mathrm{rad}}, \nabla_{\mathrm{ad}}\right) \qquad (1)$$

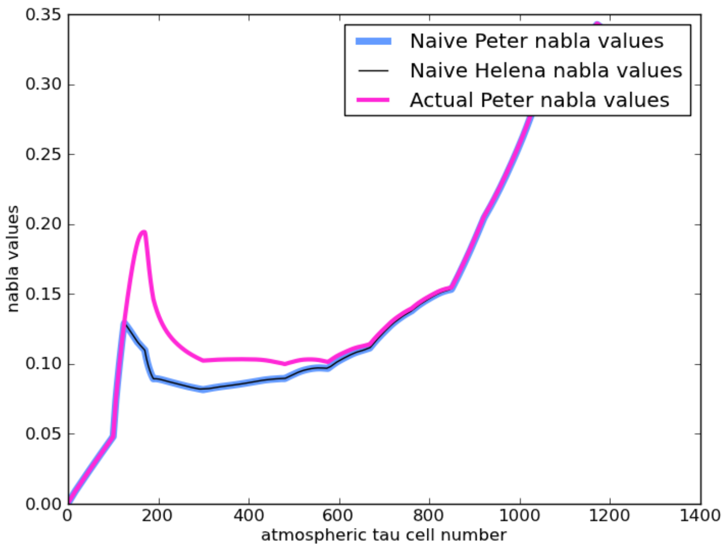


Figure 1:

A comparison of the nabla values used in Helena's and Peter's atmospheric calculations. The pink curve represents the nabla values that Peter's code actually uses in the calculations. The black and blue curves represent the nabla values you'd expect to get from Peter and Helena using Eq. (1).

Interesting result (see Figure 1).

I think the difference b/w the pink and other curves may be due to Peter's code using some sort of convective energy gradient, in addition to the radiative and adiabatic ones. The question is: why isn't that info included in the ABNAB value that his code returns? And, how are the ADNAB and CONVNAB (for lack of a better variable name) values combined? Averaged? Inverse fraction adding? Simply added? Need to find the portion of his code that calculates CONVNAB & combines it w/ ADNAB to answer these questions.

Looking at the NABLA subroutine in Peter's code points to a sort of fiddly, slog-y process of incorporating the CONVNAB calculations into Helena. Before diving into that whole processes, I should check that this nabla-discrepancy-business really *is* the thing that will fix my atmos calculation problems.

Today's Work:

1) Write out the tpnab[1-4] values from Peter's code to a text file (convert them from tpnab to nab before writing them out)

2) Read that table into Helena

3) Use those Peter's nabla values from the file, rather than the ones Helena calculates itself. Have Helena generate all of the other RK4 values itself, though (temperature, pressure, etc)

4) Capture the outputs of the Helena run (in the RK4_debug... file format)

5) Plot the resultant T1 curves from that Helena run, and compare them to the T1 curves from the analogous Peter atmos calculation run.

The results from this test are plotted in Figure 2.

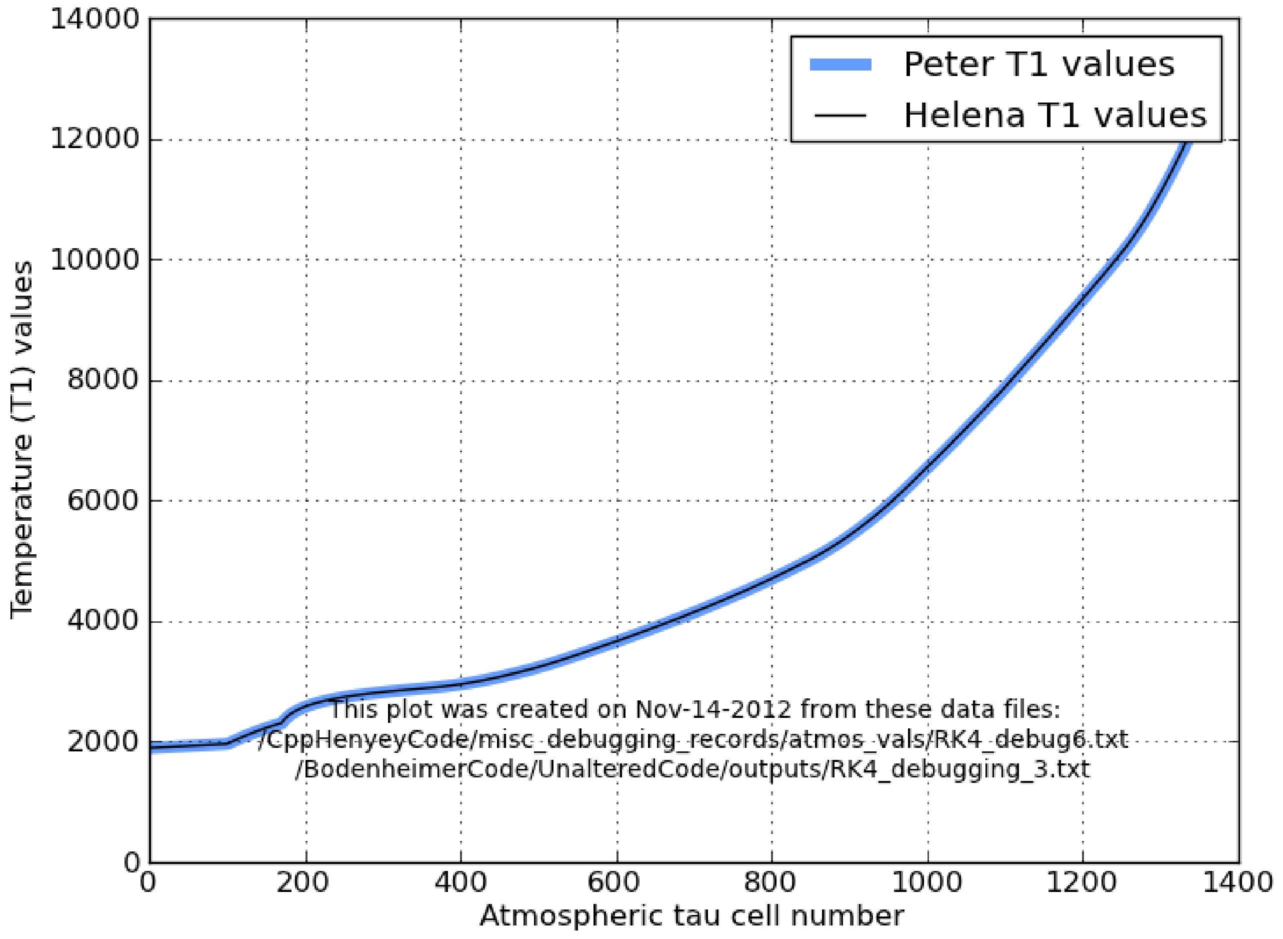


Figure 2

The results for the T2, T3, and T4 values are the same, in that they all now agree very well b/w the two codes. This proves that the thing that was causing the difference in the atmos codes' results was the difference b/w the nabla values they were using. Specifically, Peter's code is using Eqn. (2), while Helena is using Eqn. (1).

$$\nabla \equiv \min\left(\nabla_{\rm rad}, \nabla_{\rm ad} + \nabla_{\rm conv}\right)$$

Eqn. 2

For reference, the difference that this causes in the nabla values used b/w the two codes is shown in Figure 3.

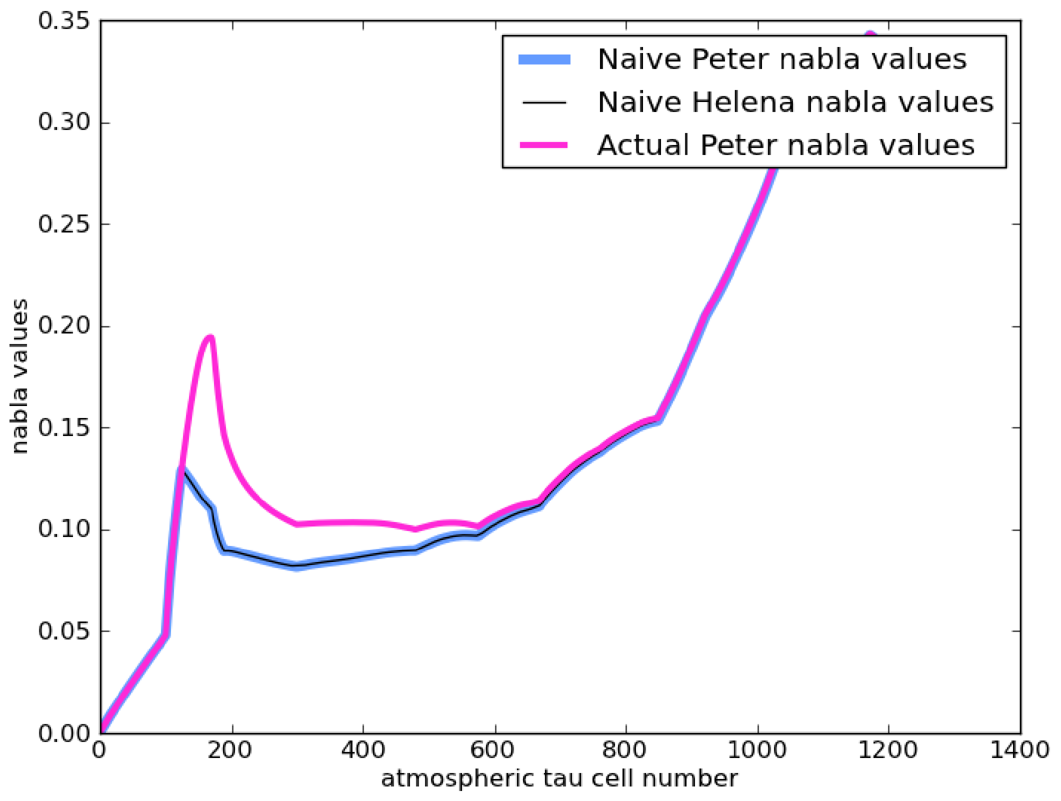


Figure 3

Need to remember to ask Greg why the nabla profile difference shown in Figure 3 would lead to the different atmospheric temperature profiles-- in terms of physical processes/intuition for what's going on in there.

So, now that we know that the difference b/w the atmos codes' results is due to the non-zero values of CONVNAB in Peter's code, the next thing to do is to see if I can figure out a way to (artificially?) suppress that in Peter's code. In his nabla subroutine, it looks like this convective gradient thing depends on the mixing length scale height parameter.

If I set the L/H parameter to zero (in the input conditions file), will that turn off the convective gradient contribution in Peter's code?

Using inputs/10MjNF_no_conv_nab.start as the input. It has L/H set to zero.

Well, it doesn't cause the code doesn't crash!

Outputting the RK4 debugging results from that run to outputs/RK4_debugging_4.txt

Let's plot those results in python, and compare them to the Helena atmos outputs from one of its runs that used Eqn. (1) to generate nabla values.

Results from this test shown in Figure 4.

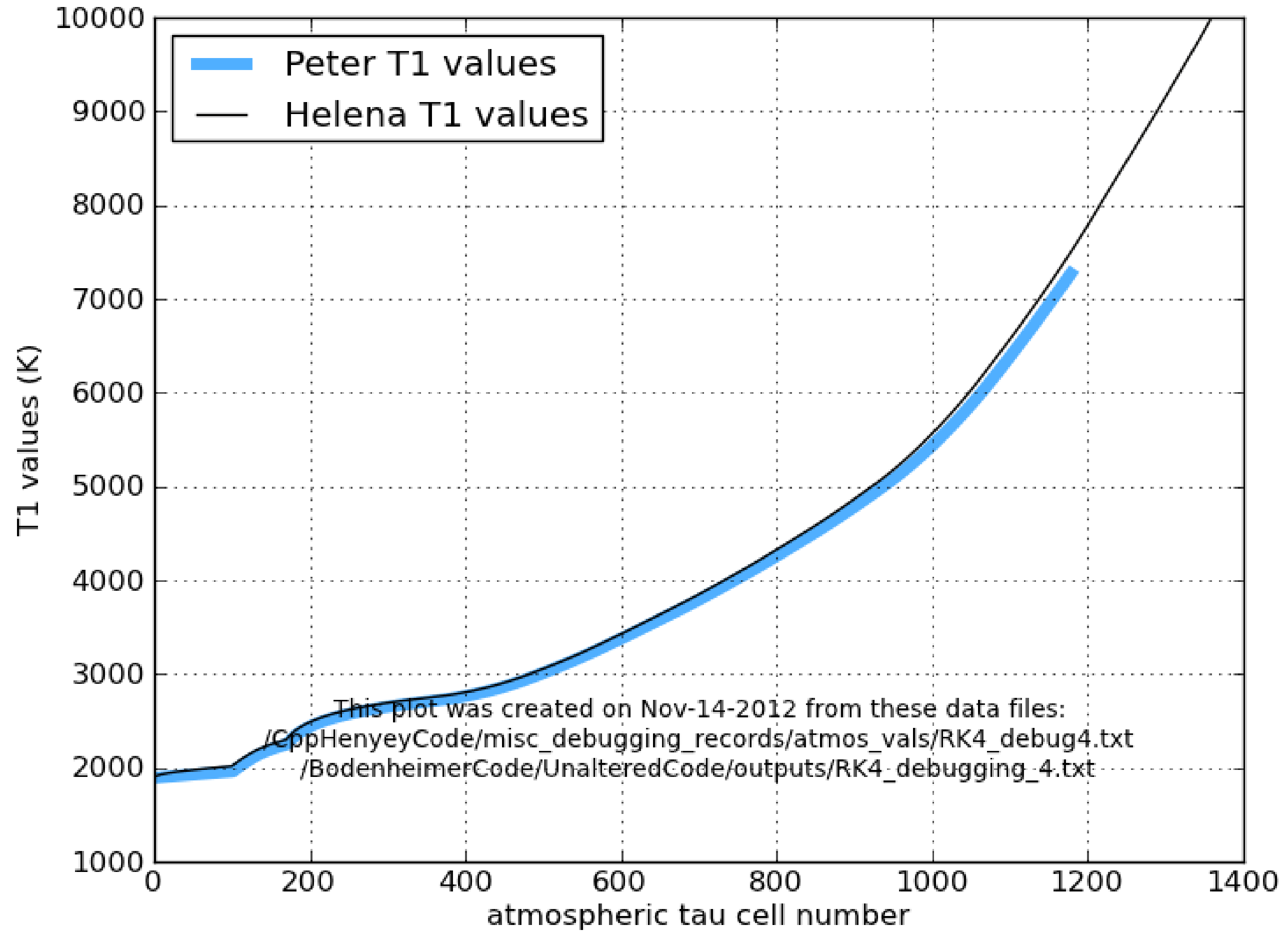
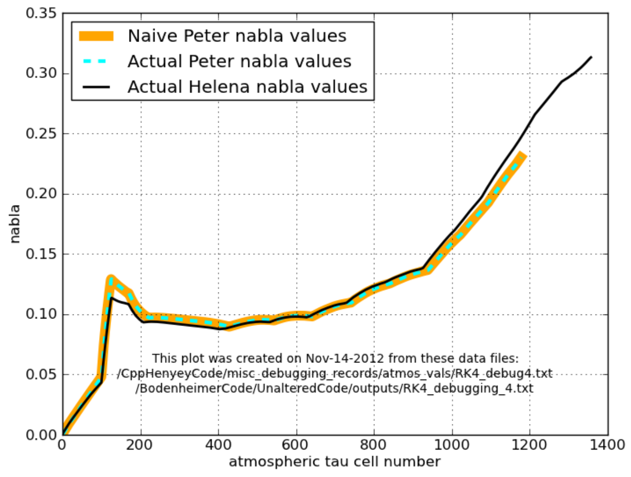


Figure 4:

Comparison of the atmospheric temperature profiles generated by Peter and Helena when the mixing-length parameter in Peter's code is set to zero (i.e. CONVNAB contribution to overall TPNAB value is eliminated...?).

Figure 4 shows that the two codes results agree very well, though not perfectly, when the CONVNAB value is Peter's code is forced to equal zero (at least, I think that's what setting the L/H parameter to zero does. It's possible that in the case L/H=0, Peter's code resets the RHL parameter to some default non-zero value. The RHL parameter shows up in the CONVNAB calculations, and is derived from the L/H parameter specified in the input file.)

Just a note that the difference in the two profiles' "lengths" (number of tau cells) isn't an issue. I've written in 1358 tau cells to the Helena atmos code for this debugging process, since that's the number of tau cells that the L/H > 0 Peter run created. However, when L/H = 0, Peter's code calculates a different atmos profile. It's not surprising that it has a different number of tau cells. The important part is how well the two profiles agree above the floor of the atmosphere (around tau cell # ~1200 in the Peter profile). In particular, the P, T, and R values at the bottom of the atmosphere in both runs agree well, though not 100% perfectly. I may want to go back, later, and actually hard-code CONVNAB = 0 into Peter's code, because it looks like there's still maybe some non-zero contribution from that parameter going on. See Figure 5 for evidence of that.



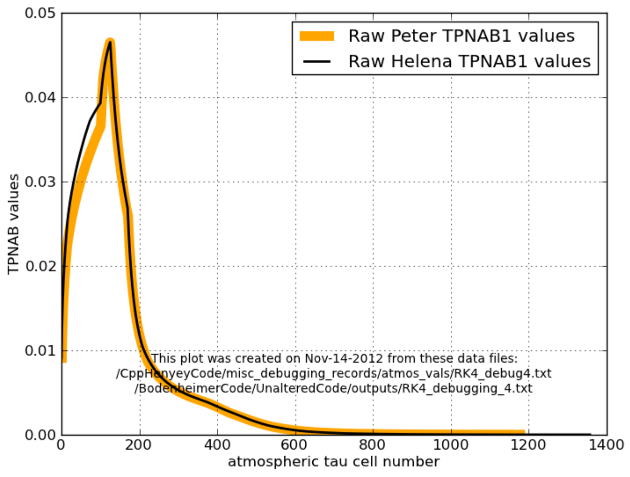


Figure 5:

A comparison of the nabla and TPNABLA values generated by the two codes when Peter's code uses the input L/H = 0.

For reference, Eqn (3) defines TPNAB.

$$\mathrm{TPNAB} = \frac{T}{P} \cdot \nabla$$

Eqn. 3

To sum up:

I think I know why Peter's code and mine weren't agreeing. It's due to the way the nabla values in the atmosphere are calculated.

MAYBE NOW IS A GOOD TIME TO BREAK OUT THE CAUTIOUS OPTIMISM?

From here, the next steps are:

- See if Peter's code will still converge a 10 Mj no fusion model w/ the L/H parameter set to zero.
    - If this step fails:
        - See if Peter's code will converge a 1 Msun with fusion system, starting from the PMSS tar.star-like polytropic initial Conditions.
            - If this fails: you need to bite the bullet and implement those CONVNAB Calculations in Helena.
        - Try Peter's code (w/ L/H = 0) on a 1 Msun polytropic model WITHOUT fusion. I
            - If this fails, get Greg's help, b/c I have no idea why this would happen if it worked on the otherwise identical with-fusion case.
        - Try Peter's code w/ L/H=0 on a somewhat evolved 1Msun no-fusion model.
            - If that fails, compare how the internal structural evolution in both the body of the system and its atmosphere differ b/w the L/H=0 and L/H>0 cases. Maybe that will offer some clues about why it fails here and not before.
        - Try Peter's code w/ L/H=0 on the mass chain-down process that you used

to get the 10Mjup test-case model you've been using for the past ~year. If this succeeds, then you've just found yourself the 10Mjup no-fusion test case that you need for subsequent steps.

- If this doesn't work... ugh. First, find the mass at which the chain-down process stops working. That may give some clues about what's going on.
- COME BACK HERE AND FILL IN MORE DEBUGGING IDEAS ON THIS BRANCH.

- Make sure both codes now produce the same CDEG profiles on the 10Mj no-fusion system, particularly at the outer boundary / bottom of the atmosphere.
    - Reacquaint myself with all the procedures I used to get CDEG profiles out of the main body of the system in each of the two codes
    - Run each code for a single iteration on the 10Mj no-fusion system.
    - Capture the relevant outputs into txt files
    - Compare the results in python. (You already have scripts for comparing the two codes' results for this stuff, remember.)
- Make sure that both codes now produce the same P/R/L/T correction profiles.
    - If not, then you've still got problems with the outermost CDEG values... probably
    - THERE ARE PROBABLY STILL ADDITIONAL DEBUGGING STEPS TO PLOT OUT, HERE, SO COME BACK TO THIS.
- Check that Helena successfully converges the 10Mj no fusion model
- Compare the converged model to Peter's results
- Check that Helena and Peter also produce the same converged model with dTime > 0