Gaia and the HR Diagram

2018 MESA Summer School

August 15, 2018

1 Lab 1: Timescales

In this minilab, we will focus on how the observable properties of stars vary as a function of age, and the scale of those variations in comparison to observational uncertainties. We'll also play around with stopping conditions, and some custom stopping conditions in `run_star_extras`.

1.1 Examine a Solar Model

First, we'll take a look at the behavior of a solar mass, solar metallicity model with `inlist_jvansaders`. Specify some starting parameters:

```
initial_y = 0.26275
initial_z = 0.016498013
minxing_length_alpha = 1.745.
```

1.1.1 (Near-MS) Evolutionary Phases of the Sun

Let's run this solar mass model to several different reasonable stopping points:

1. The beginning of the main sequence (zero-age main sequence, or ZAMS). For our purposes today, we'll define this as the first point on the stellar track where the nuclear energy generation exceeds 99% of the total luminosity. (As opposed to the pre-main sequence, where gravitational collapse contributes significantly.)

2. Somewhere in the middle of the main sequence. Let's use where the core hydrogen mass fraction drops below 0.3.

3. The end of the MS. We'll define this as the point at which the core H mass fraction drops below 1×10^{-4}

4. Bonus: Somewhere on the subgiant branch, let's say roughly midway in temperature between the main sequence turnoff and the base of the red giant branch. You may want to let MESA run from the main sequence turnoff to the giant branch to inform your choice of Teff-based stopping condition.

MESA already has built-in functionality to get you to these points: take a look at the `controls` inlist options for "where to stop". For each phase, make sure you output a model and a photo at the appropriate timestep, because we'll be coming back to these momentarily. To save yourself time, you can restart your run from a saved model or photo after reaching each evolutionary phase. The file `inlist_jvansaders` provided in your work directory has these options for output set, although you should choose appropriate filenames for output. This inlist is also set up to display a `pgstar` plot of the HR diagram as the star evolves, and writes the age, luminosity, and effective temperature at termination to the terminal window. Do models stop *precisely* where you want them to? What control could we add to do better?

1.1.2 Movement on the HRD

When we talk about using HR diagram position to measure ages, what really matters is how rapid that motion is in comparison to our ability to constrain observational quantities. Here, we'll look at contrived scenario: imagine we know the metallicity, helium abundance, and mass of a star perfectly. Given a measurement of the luminosity and effective temperature, how well can we expect to measure the age? Let's assume our temperature errors are $\sigma_{Teff} \sim 100$K and luminosity errors are $\sigma_L \sim 10\%$ (both reasonable, in the Gaia era!).

Let's define a toy custom stopping condition to give us a feeling for where we have good constraints on age. Take a look at `run_star_extras.f`, and the subroutine `extras_check_model`. Let's say we want the code to halt if the star evolves outside of the ellipse defined by $(\sigma_{Teff}, \sigma_L)$, centered on the last (Teff,L) from your model. Let's make the code stop when:

$$\frac{(T_{eff} - T_{eff,0})^2}{\sigma^2_{Teff}} + \frac{(L_\star - L_{\star,0})^2}{\sigma^2_{L_\star}} > 2 \qquad (1)$$

MESA has built-in controls for interfacing with `run_star_extras.f`. There are empty, 100 element arrays for real (`x_ctrl`) and logical (`x_logical_ctrl`) variables that we can use for our own purposes. These variables are passed to `run_star_extras` and available for use there, and can be specified by the user in the `controls` inlist. We'll use these to pass values to our stopping condition. Let's call `x_ctrl(1)` $= T_{eff,0}$, `x_ctrl(2)` $= \sigma_{Teff}$, `x_ctrl(3)` $= L_{\star,0}$, `x_ctrl(4)` $= \sigma_L$, and use `x_ctrl(5)` as a tolerance value. We'll use `x_logical_ctrl(1)` as a toggle to turn the stopping condition on or off. In `run_star_extras.f`, let's add some code. Before you implement the code below, read through it, and write out in "pseudo-code" what the routine is actually doing: make sure you understand what we're about to do.

```
! returns either keep_going, retry, backup, or terminate.
integer function extras_check_model(id, id_extra)
   integer, intent(in) :: id, id_extra
   integer :: ierr
   real(dp) :: gerr, sigL
   type (star_info), pointer :: s
   ierr = 0
   call star_ptr(id, s, ierr)
   if (ierr /= 0) return
   extras_check_model = keep_going
   ! is the new step inside the error ellipse?
   if (s% x_logical_ctrl(1)) then
        sigL = s% x_ctrl(4)* s% x_ctrl(3)
        gerr = (s% Teff- s% x_ctrl(1))**2 / s% x_ctrl(2)**2 + &
               (s% photosphere_L - s% x_ctrl(3))**2 / sigL**2
        if (abs(gerr-2.0d0) < s% x_ctrl(5)) then
               ! we've reached the stopping point
               extras_check_model = terminate
               ! save a profile
               s% need_to_save_profiles_now = .true.
               ! and update the star log,
               s% need_to_update_history_now = .true.
        else if (gerr-2.0d0 < s% x_ctrl(5)) then
               ! model is still inside error ellipse
               extras_check_model = keep_going
        else if (gerr-2.0d0 > s% x_ctrl(5)) then
               ! model has overshot the stopping condition
               extras_check_model = retry
        endif
   endif
end function extras_check_model
```

In the subroutine `extras_startup` in `run_star_extras.f`, let's add some handy things to output when our stopping condition is turned on:

```
   if (s% x_logical_ctrl(1)) then
      s% x_ctrl(1) = s% Teff
      s% x_ctrl(3) = s% photosphere_L
      write(*,*) "Starting Teff is", s% Teff
```

```
    write(*,*) "Starting photosphere_L is", s% photosphere_L
endif
```

Let's also add some output for after the step is taken in the subroutine `extras_after_evolve`:

```
write(*,*) "Starting Teff is", s% x_ctrl(1)
write(*,*) "Stopping Teff is", s% Teff
write(*,*) "Starting photosphere_L is", s% x_ctrl(3)
write(*,*) "Stopping photosphere_L is", s% photosphere_L
write(*,*) "Starting age is", s% x_ctrl(6)/1.0d9
write(*,*) 'Delta t is ', (s% star_age - s% x_ctrl(6))/1.0e9
```

Now, let's run your models with the new conditions. Load a model or a photo from one of our 4 stopping points, specify the starting temperature and luminosity and uncertainties, provide the necessary `x_ctrl` values, toggle the stopping condition on, and set it running. Now consider:

1. At termination, take a look at the luminosity and effective temperature. Did MESA do what you asked it to do, terminating when the condition at Eqn 1 was met? What happens if you relax the tolerance value?

2. Take note of the age at which our custom stopping condition is met for each of the phases from part 1: what was the Δt during the evolution that took you outside of the error ellipse? How does this compare to the MS lifetime of the Sun?

3. How does Δt vary between the ZAMS, middle-MS, and the end of the MS? What evolutionary phases lend themselves to this kind of HR diagram-based age inference?

1.2 Other masses

Now, let's look at stars that are not the Sun. Repeat this exercise, for a 1.5 and 0.5 $M_\odot$ model. Is there anything noteworthy about the model at $X_{H,cen} = 0.3$? How does the movement on the HRD compare to the 1.0 $M_\odot$ case? Should we be using HR digram position to measure the ages of late-type dwarfs?

2 Lab 2: Fundamental Parameters

In this mini-lab, we'll focus on the manner in which changes in helium abundance, bulk metallicity, and mass affect a star's location on the HR-diagram. While these quantities are the fundamental governors of stellar structure, they are rarely directly measured. Masses are generally only directly measured in binary or multiple systems; the surface metallicity is readily measurable, but not necessarily representative of the bulk interior abundances; and although surface or near-surface helium abundances can be inferred in hot stars or through detailed asteroseismology of the helium ionization zones in cool stars, it is generally difficult to constrain. We'll take a look at how these fundamental properties affect those observables you are more likely to have available, with an emphasis on HR-digram position (because Gaia!).

3 Mass

Let's take our solar mass model from the last exercise and change it by some modest amount - a few hundredths of a solar mass, and run it to solar age (4.57 Gyr). Take note of the final luminosity and effective temperature. How dramatic is the change in comparison to our ability to measure these quantities? Play around a bit with different choices for the mass, and different stopping points along the evolutionary track (ZAMS, MSTO, SGB, RGB) to get a feeling for how mass shifts your evolutionary tracks (Those whose machines are slower to run models might want to divide different runs among members of the group and share their results.) What is the sense and magnitude of shifts that changes in mass induce?

4 Metallicity & Helium

Let's alter our starting metal abundance. Keep in mind that this is a mass fraction, and not a logarithmic metallicity, where the relation between the two is

$$[Z/X] = \log\left(\frac{Z_\star/X_\star}{Z_\odot/X_\odot}\right) \tag{2}$$

where Z is the metal mass fraction and X is the hydrogen mass fraction, and here we've adopted $Z_\odot/X_\odot = 0.02289$

Let's look at a star that's a sub-solar in it's metal fraction ($[Z/X] = -0.1$), while keeping the helium fraction the same as the solar metallicity model. How does this model compare to a solar model at the solar age in terms of our observables? How does it compare to changing the mass in terms of the effects of the tracks? Now, instead, let's hold Z/X fixed and change the helium mass fraction by a few percent (recall that the primordial helium abundance is ~ 0.248, while the Sun is at ~ 0.27 today). How does this shift your tracks? If you had specified an `initial_z` but no `initial_y`, what would MESA have done? (Look first, before moving on!)

You've discovered that MESA defaults to using an assumption about chemical evolution to set the helium abundance if you don't tell it otherwise, namely that $Y = Y_p + \frac{dY}{dZ}Z$, where Y_p is the primordial helium, and dY/dZ describes how helium and metals increase in lockstep. Now, let's say we want $dY/dZ = 1.4$ instead, motivated by observations. Let's look at our sub-solar case again, but this time enforce the above chemical evolution law when you set the helium abundance. How is this model shifted with respect to our original solar model?

5 Detailed Abundance Patterns

When we talk about the measured metallicities of stars, we're generally talking about the quantity [Fe/H], not a bulk [Z/X]. For simplicity, we often assume that the overall *chemical pattern* of our models is the same as that of the Sun: that all stars have identical [X$_i$/Fe] ratios, where X_i is any "metal" heavier than helium, even if the overall metal fraction is allowed to change. This is equivalent, then, to assuming that [Fe/H] = [Z/X] = $\log(\mathrm{Z/X/Z_\odot/X_\odot})$. In reality these chemical patterns vary from star to star, and study of these patterns is a field unto itself. It's worth noting that standard isochrones and stellar models often include only examples of solar abundance patterns and "α-enhanced" variants. It's also worth noting that there is real controversy over the solar abundance pattern itself, particularly the amount of oxygen in the Sun ("the Solar oxygen problem")- with disagreements at the 40% level (Asplund vs. older oxygen determinations)!

Look at `initial_zfracs`, and see that we have made a specific choice here. If we decide we want to use the Asplund 2009 solar values instead, we need to

1. Change `initial_zfracs` to reflect our choice of Asplund et al. 2009
2. Change `kappa_file_prefix` and `kappa_lowT_prefix` so that the new abundance pattern is reflected in the opacities.
3. Adopt $Z_\odot/X_\odot = 0.0181$ to reflect the lower total solar metal content inferred by Asplund et al. 2009. This becomes your new definition of [Fe/H] = 0.
4. Be aware that in many cases your atmosphere boundary condition also depends on composition. Tabulated boundary conditions from atmosphere models generally assume an abundance pattern. Do the interior and atmosphere abundance patterns match here?
5. The MESA equation of state takes Z as an input, and allows for Y enrichment. What does this mean for switching abundance patterns?

Run a model using the Asplund abundances. When comparing this to GS98 abundance models, we'd want to do a proper solar calibration atop of these changes (too lengthy for today) so that we reproduce the Sun at the solar age with both sets of abundance patterns. If you've got extra time, you can try to match the Sun, varying α and Y, and see what a touchy process this can be. When a calibration is actually done, distinct differences show up in the sound speed profile, particularly near the base of the convection zone - much of which is due to oxygen opacity. So: this choice of abundance pattern can matter in detail!

6 Lab 3: A Physics Case Study: Diffusion

The meat of a stellar model is in the input physics. We're treating a rotating, magnetic, convecting star as a 1D object, and in doing so need to make choices about how parameterize the impact on those physical processes on the stellar structure. We also need to include microphysics: nuclear reaction rates, equations of state, opacities, etc. Some of these data come from measurements, but many come from detailed theoretical calculations that are then packaged in tables that stellar evolution codes ingest and utilize. In this lab, we're going to do a case study of one physical model ingredient that is important for measuring ages in MS/turnoff stars: atomic diffusion, or the gradual settling of elements in a star. You should take a look at Dotter et al. (2007, ApJ, 840, 99) for a full discussion of the topic.

6.1 A more careful inlist

For this section, let's look more deeply at our inlist. Notice we have:

```
extras_lcpar = 1
extras_cpar(1) = 'photosphere_tables'
```

and take a brief look at `run_star_extras` to see what we're doing. This helps with convergence in our early models, and is motivated by the MIST isochrone set (Choi et al. 2016). Bonus points if you can connect this choice to something you see in our model tracks. MESA automatically interpolates these tables to your specified metallicity.

Let's also move (closer) to a solar calibrated model for the diffusive case. Adopt Z = 0.01835, Y = 0.27240 and $\alpha = 1.865$

6.2 Turn diffusion on

Finally, let's enable diffusion. In your inlist, set `do_element_diffusion = .true.`... Woah, that was easy! Run a 1.0 $M_\odot$ model to solar age. Make sure you set up `pgstar` to show an HR diagram.

In reality, you've just initiated a whole suite of default settings that dictate the efficiency of the diffusion, when diffusion occurs, and how different elements are treated. While your models are running, take a look at the diffusion section of `controls.defaults`. You've just inherited this set of choices (and one custom condition, search "diffusion" in`run_star_extras`).

... And now that your model has finished, you can see it's actually *not* so easy. If you look at the HR diagram, you can clearly see that we have some rapid changes in L and Teff on the MS that aren't terribly physical. Let's diagnose.

6.2.1 Abundance window in `pgstar`

Let's set up pgstar to show you some details about the metals. You can use the abundance window to look at H and He - both should show evidence of diffusion.

6.2.2 Examine the surface metallicity

Let's look specifically at the surface metallicity, since this one of our observables, and a likely culprit. Let's add a custom history column in `run_star_extras`. You'll need to specify the number of extra columns in `how_many_extra_history_columns`, while you specify what you actually want output in `data_for-_extra_history_columns`. Let's output the logarithmic surface metallicity with respect to your initial model metallicity. Plot this using the history plots in `pgstar`. (hints [1]).

1. Rerun your model, and take a look at the plots. What happens if you just tighten up your timestep and mesh controls a bit?

2. Use plots of the surface metals to locate models that are behaving badly on the MS. Use your profile output options to output files for these models (making use of ./re to restart your models.)

3. One of the *best* things about MESA is the community. Go to the archives of the mesa mailing list and do some reading. I'd suggest searching for "diffusion instability". This is a known issue! Implement the fix in your inlist, and look to see if your surface abundance is still very sensitive to your timesteps and mesh choices.).

6.2.3 Explore

Now, let's run a few variations. Choose a model from the spreadsheet. Run this model to $X_c = 0.3$. Record the Teff, L, [Z/X], and age, and record it in the spreadsheet. Then restart the model, and let it run to the giant branch. What happens?

Now, turn diffusion off, adopt the surface [Z/X] at $X_c = 0.3$ as the *bulk* metallicity. Rerun models, varying the mass, and try to match logT and logL within 0.001. Revert to the *alpha* from our earlier labs: the value you would have used if you didn't have diffusion on. Record the age of this model on the spreadsheet. (Hint: the stopping condition in Lab 1 can be easily adapted for this, as can your knowledge of mass and metallicity effects from Lab 2.) How important is diffusion in our age inference for stars of different ages and masses?

6.2.4 What we're doing wrong

In reality, stars are not as idyllic as we've assumed here. Stars more massive than ~ 1.3 $M_\odot$ tend to be rather rapid rotators, and it's easy to imagine that there are extra mixing processes in play [2]. This goes for lower mass stars too: any processes that aids in mixing can reduce or erase the signatures of diffusion. We're being optimistic here about the magnitude of the settling.

---

[1] Hints: add `use crlibm_lib` to the beginning of `run_star_extras`. Remember that $Y + X + Z = 1$, and that you can look for named quantities in `public/star_data.inc`

[2] We've actually ignored a *lot* of important (and uncertain) physics here. We've left out overshooting, semiconvection, rotation, etc... Take a look at the test suite 1.5 $M_\odot$model with diffusion to get a sense of what we've been leaving out.