Physics of Stars
ASTR 220A Winter 2025
Homework 1
Due January 14

1 (50 points). Install MESA (`http://mesa.sourceforge.net`). Run the `1M_pre_ms_to_wd` model (`https://docs.mesastar.org/en/release-r24.08.1/test_suite/1M_pre_ms_to_wd.html#m-pre-ms-to-wd`) to make sure it works. You should get a bunch of data in the `LOGS` subdirectory. If it doesn't crash and you have files there, it is probably working. You do not need to run the simulation to its end.

Copy that model to another directory (maybe something like `astr2201/homework1/`) Adjust the model to examine a 6 $M_\odot$ star through the end of the main sequence (terminal-age main sequence or TAMS, which is equivalent to central hydrogen depletion). Make the simulation save profile files for the star near the start of the main sequence (zero-age main sequence or ZAMS) and at the TAMS. This will require editing a few things in the `inlist` files.

(a) Once the simulation has completed successfully, use the `history.data` file to plot the star's evolution on the H-R diagram. If you ran Mesa with their plotting enabled, that will produce a plot you can compare to, but you should make your own plot using the output file. Make the plot the traditional way (logarithmic and with temperature increasing to the left). Mark the start and finish of the simulations as well as the ZAMS and TAMS. You might need to ignore the first few data points from the start/end of the file to make the plot is readable.

(b) Using the TAMS `profile.data` file generated, plot the "ρ–T" diagram. That is, plot temperature as a function of density with logarithmic scales. Label where the center of the star is and where the surface of the star is.

(c) Using the output `profile.data` files for ZAMS and TAMS, plot the H, ^{4}He, ^{12}C, and ^{14}N abundance profiles for the star. That is, plot the abundance of each element (in log space) as a function of radius (choose between log and linear to make the data easy to see). Be careful about the extreme radius points, which may need to be ignored for the plot.

(d) Briefly explain why the abundances of each element changes.

Depending on your settings, you may generate these plots while running the simulation. But we want to make sure that you can analyze the output files, so please make your own plots separately. There are some nice python functions to help you analyze the outputs (check out `https://docs.mesastar.org/en/release-r24.08.1/using_mesa/output.html`).

When you make your plots, make sure the ranges, symbols, etc are chosen to make analyzing the data easy. Someone should be able to look at your plot and pull out the trends.

2 (50 points). M67 is a nearby open cluster. That means that all of the stars were formed from the same molecular cloud at roughly the same time. Because all of the stars in an open cluster are coeval (they have the same age), open clusters are an extremely powerful tool for understanding stellar evolution.

Step-by-step, you will separate cluster stars from other stars in the same direction as M67 that are not in the cluster. Be conservative at first, trying not to lose any stars. Each progressive cut will help you remove interlopers, so it doesn't need to be perfect at the beginning. In fact, it is often best to set up your code with each cut as a parameter and then adjust/iterate to get good values as you go.

You will use the python notebook I provided on Canvas. This will guide you through much of the assignment.

Remember when making plots to make them clear. Make sure to label all axes and points (or describe them in a caption). Some plots might have a lot of points where the individual points overlap, making it one big blob. When this happens, we can't see the information, and you should use an alternative plotting method like contours.

We will examine open clusters using *Gaia* data.

(a) Read and run the provided python notebook. Make sure to read the code so you understand each step. This notebook will query *Gaia* for stars near the position of the open cluster M67. It will make plots for the positions and parallax for the stars. When you get the same output as provided in the notebook, move to the next step.

(b) When making the parallax histogram, there is a peak near zero, and the code specifically ignores all negative values. Explain what the peak represents physically and why we can ignore negative parallax.

(c) Now rewrite all of the code to be functions. We want to be able to run a short set of commands like:

```
ra_hex = "08h51m23.0s"
dec_hex = "+11d48m50s"
radius = 25*u.arcmin
cluster_data = gaia_query(ra_hex, dec_hex, radius)
plot_cluster_position(cluster_data[''ra''], cluster_data[''dec''])

p_min, p_max = 0.96, 1.31
par_ind = plot_cluster_parallax(cluster_data[''parallax''], p_min=p_min, p_max=p_max)
plot_cluster_position(cluster_data[''ra''], cluster_data[''dec''], good=par_ind)
```

You might want/need to add additional parameters like the symbol size, bin size, etc.

(d) Using the peak and width of the parallax overdensity, determine the "best" **distance** and the distance *range* for that peak. This is roughly the distance and distance uncertainty to M67. Report those values.

(e) To further select M67 stars, plot the proper motions ($\mu_{\rm RA}$ vs. $\mu_{\rm Dec}$) of the selected stars. Cluster stars should move together. Mark a region that selects cluster stars with a polygon.

(f) Using both the parallax and proper motion cuts, plot the positions of cluster stars. You will notice that our conservative radius was probably *too* conservative. Write a new function or edit the previous one to plot a dashed circle at a center an radius that better selects cluster stars. At this point, you should still be conservative!

(e) Now that you have functions to easily select stars based on position, parallax, and proper motion, iterate through the functions, slowly changing your constraints. The output from the previous iteration can be the input for the new one. This means that after you exclude stars by proper motion, there will be fewer stars in the parallax histogram, which should increase the cluster signal-to-noise ratio, making it easier to choose better bounds. Continue until you converge. Still try to be pretty conservative — it is better to have some interlopers than to exclude a bunch of good stars. Just try to balance things.

(f) Using the stars that pass all of the above cuts (position, distance, proper motion), make a color–magnitude diagram (CMD) for M67. Plot M_G as a function of $(BP - RP)$, where M_G is the absolute magnitude in the Gaia G band and $(BP - RP)$ is the Gaia blue minus red color.

When making the CMD, you have to be careful to only use stars that have a color in the catalog (avoid stars where that is missing). Plot the stars in a way where you can see the individual points by having reasonable symbol sizes and opacities (no big blob where everything smears together).

If you want to be especially impressive, you can do several things to make even cleaner CMDs. For instance, you can use the uncertainties for each measurement rather than just making hard cuts. Or you could generate a metric in a multi-dimensional space (RA, Dec, p, μ_{RA}, μ_{Dec}) to avoid iterative cuts. Or you could add your knowledge of stellar evolution to remove outliers in color/magnitude space. Or you could combine everything and place it into a Bayesian framework to probabilistically produce a catalog! No need to gild the lily, but I also don't want to hold anyone back from trying more complex analyses.

Please submit all python notebooks/code along with a PDF that has plots and answers to questions. Your version of python might be different from mine and so making the plots and the PDF is necessary to make sure that I see what you want me to see.