Limitations and Stuff

There is one main limit to our code, and that is the kind of data and amount of data we get from EZ-Web. While EZ-Web is an excellent stellar evolution code, it is limited in that it cannot handle elemental degeneracy or major hydro-dynamical effects, and when the star its evolving runs into either of these things, it stops and only gives data up to that point in time. What does this mean for us? It means that for stars below 2.1 times the mass of the sun, our timeline stops right before the star becomes a white dwarf. For stars above 2.1 times the sun’s mass, our timeline stops before the star starts fusing carbon. While this is somewhat annoying, it’s not failure on us or our project. EZ-Web stops because the conditions and state of the star no longer comply with the already extremely advanced mathematical model it is using. This suggests that if we had tried to take a mathematical approach and not use EZ-Web, we would have run into the same problem as well.

It’s also prudent to note that since we’re using a database, the list of masses available for testing is limited. We solve this by having a selector palate in our UI, but if you just run stellarEvolutionSimulator with some weird value like -2000.57 or something like that, the program will throw an error and continue with an assumed value of one solar mass. The same is true with our timeline bounds.

As for challenges we encountered, we had a lot. The aforementioned mathematical challenge was a doozy, but we also had a bunch of smaller challenges, especially in designing our output.

* It turns out that Mathematica doesn’t like when you try to invert the x-axis and log scale the y‑axis of a graphic at the same time. We had to do a lot of research, and eventually had to use a directive for commands it wasn’t intended for. It works, but it appears in red in our code.
* Getting the program to run even close to the speed it does now was also a challenge.