Computational Problem Solving Final Project Proposal

Karthik Boyareddygari and Reese Danzer

**STAR LIFE EMULATOR**

**Summary**

This project would simulate the lifetime of a star after formation but before any type of death sequence showing both external and internal functions and layers, and using realistic data to model the star as a system. The user can choose the starting conditions of the star, the portion of the star’s life being viewed (via a time slider), and start points and endpoints of the timeline (if the user wants to focus on a certain area of the timeline, they can change the start and endpoints and the program will only iterate within that range on the timeline).

**Final Product (diagrams below)**

Our program will be run the same way standard mathematica commands do: the user enters the name of the program and chooses from the interface the initial conditions. If the user wishes, they can also enter start and endpoint values for the timeline in an input field (as fractions or decimals), meaning that the bounds on the viewing timeline are not necessary for the program to run. Once the user executes the program, it will output a dialog similar to that of a Manipulate command. A slider at the top will allow the user to manipulate the timeline; the display will show a picture of the star at the current point on the timeline; an input field will take lower and upper bounds for the timeline; the animation will be composed of concentric regions in the apparent form of circles that will fluctuate in response to the evolution of the star and have different colorations (this will allow for viewing inside the star and may also allow for indication of fusion processes etc); various data types will be displayed to the right of and below the diagram in the form of readouts. There will be a plot with the star as a point on it as it progresses throughout its lifetime. There would also be a sphere representing the exterior of the star. The user can either scroll through the diagram at will, or use the play function to watch the star change at a scaled equivalent of how it would change in real time. Users would be able to use the start/endpoint box to control how much and what parts of the timeline are generated.

**Increments**

We’re going to take a top-down approach to this task, and thus our first task will be to design the primary features of the program, such as the interface and the timeline setup. The interface would be prepared so as to make it able to take all the information that we intend to pass to it that would be calculated in the next increment.

The second increment is where we must take in the data and adapt our code to handle it and interpret it for the interface. The data will be in an inconvenient form, and making the graphics continuous will introduce some difficulty, as the data will be in discrete data points. The code also must be generalized to take data for any set of starting conditions.

The third increment is the major integration phase. We would need to mesh the data of the star’s dynamics with the graphics, which should not be very difficult if our first increment was done correctly. As long as our foresight into what values will be passed to the graphic is good in the first increment, it should be a straightforward insertion of functions (for the regions and their attributes) into the interface and printed output.