Simulation of the evolutionary process of the Sun into a Red Giant

Steven Li
Phys 140, Fall 2017, San Jose State University

The sun and all stars undergoes an evolutionary track similar to the lifecycle of a human being. I have created a python program that attempts to stimulate the process that the sun undergoes into becoming a red giant through utilizing a polytrope to model the sun. Modeling the sun after a polytrope allows me to find a portion of density and pressure from a given n value which behaves like specific stars. From that I will able to find physical properties of the sun and utilizing a luminosity function over time, I can find physical properties of the sun throughout it's lifetime on the main-sequence before becoming a red giant.

INTRODUCTION

All stars including the sun undergo an evolutionary track called stellar evolution where stars undergo a sequence of dramatic changes that warp the nature of the star resulting in changes of size, temperature, luminosity and behaviour throughout it's lifetime ending with a bang or a quiet whimper. Stellar evolution is a process which we can't directly observe because the amount of time for a star to perform one of these phase shifts is way beyond our time on earth. So based on the observation of the types of stars in the universe, we have a general idea of how the evolution causes a star change over the course of it's lifetime. Most stars in the universe are main-sequence stars which are in a state of equilibrium and it will be in this state for the majority of it's life. Any star that diverge from the main-sequence will rapidly change signifying the beginning of the end of it's life. In this project, I will be stimulating the evolutionary track of the Sun up to the Red Giant phase using a polytrope model and other assumptions.

THEORY

In order to model the sun, I will make the assumption the star is a non-rotating spherically symmetric, an ideal gas, black-body and is isolated from other stars that might affect it's evolutionary track.
The stellar structure of a star following these assumption is dictated by 5 stellar equations.[1]

Hydrostatic equilibrium:

$$\frac{dP}{dr} = -G\frac{M_r\rho}{r^2} \tag{1}$$

where G is the gravitational constant, M_r is the mass of the star, r is the radius and ρ is the density. This equation describes the pressure gradient on the star in a static distribution of pressure and density to keep the star in equilibrium. Otherwise, the gravity towards the star created by the star will be greater than the pressure exerted by the star, causing the star to collapse on itself and that would move it away from the main-sequence signifying the beginning of it's end.

Mass Conservation:

$$\frac{dM_r}{dr} = 4\pi r^2\rho \tag{2}$$

Energy Conservation:

$$\frac{dL_r}{dr} = 4\pi r^2\rho\epsilon \tag{3}$$

where L_r is the enclosed luminosity of the star and ϵ is the local energy flux. These two conservation stellar equations are important to keep the star isolated because the mass conservation relates the relationship between mass and density and how it has to be equivalent to each other. The same goes for energy conservation which describes the relationship between the energy generation of the star and the energy distributed in the form of light measured in luminosity.

Radiative Transport:

$$\frac{dT}{dr} = -\frac{3}{16\pi a_r c r^2}\frac{\kappa\rho}{T^3}L \tag{4}$$

Convective Transport.

$$\frac{dT}{dr} = \frac{\gamma-1}{\gamma}\frac{T}{P}\frac{dP}{dr} \tag{5}$$

The remaining two equations describes the transport of energy either in the form of radiative or convective. Where κ is the opacity, a_R is the radiation constant and γ is the ratio of the specific heat based on the star's pressure and volume.

To simply the simulation, I'm making the assumption that energy generation and transport will be ignored resulting in only a relationship between pressure and density to remain:

$$P(r) = K\rho^{1+1/n}(r) \tag{6}$$

Where K is the constant of proportionality of a star between it's total mass and radius and it's different and specific for each star. This results in a polytrope with a polytropic index n that describes the behavior of the star such that:

TABLE I. Polytropic Index Behavior

n	Behavior
1.5	Non-ionized gas in Convective equilibrium
3	Non-ionized gas in Radiative equilibrium

Combining the remaining stellar equations with the polytrope results in the Lane-Emden 2nd Order Differential Equation.

$$\frac{1}{\xi^2}\frac{d}{d\xi}(\xi^2\frac{d\theta}{d\xi}) = -\theta^n \tag{7}$$

Where the variables ξ is the dimensionless variable to indicate radius and θ is a dimensionless variable of density to scale the star with. The solution to the Lane-Emden equation is different for polytropic index n. The Lane-Emden equation requires either a value for K or the central density of a planet ρ_c. In this case, I will be giving the value for central density ρ_c which would give me a value for K through:

$$K(M,\rho) = \frac{2(2\pi)^{\frac{1}{3}} G M^{\frac{2}{3}} \rho^{\frac{1}{3}(1-\frac{3}{n})}}{\frac{d\theta}{d\xi}^{\frac{2}{3}}(n^3+3n^2+3n+1)^{\frac{1}{3}}\xi_1^{\frac{4}{3}}} \tag{8}$$

Getting the value for K allows me to find the radius of the star only giving a parameter of the density of the star through the manipulation of the Lane-Emden equation:

$$R(\rho) = (\frac{n+1}{4\pi G})^{\frac{1}{2}} K^{\frac{1}{2}} \rho^{\frac{1-n}{2n}} \xi_1 \tag{9}$$

Now that I have the radius R_* and K, I can determine the pressure of the star which will be useful when the time-dependent equation changes the density of the star since mass is loss as a function of time.

$$M_{lost}(L,t) = \frac{L}{\epsilon_{react}} M_{react} * t \tag{10}$$

Where ϵ_{react} is the energy in J of each reaction and M_{react} is the mass lost of each reaction. The luminosity is the amount of energy being emitted and I can find the number of reactions that occur during a time-frame utilizing ϵ_{react} and M_{react} to find the total mass during a certain time-frame.
At this point, I can determine what the temperature of the star at a certain point by providing a parameter of pressure P, composition of the star denoted by the mean molecular weight of the star μ and K since I'm assuming the star is an ideal gas:

$$T(\rho) = \frac{P\mu m K^n (\frac{(\frac{k}{\mu m})^{-n}}{k})^{\frac{1}{n+1}}}{\omega} \tag{11}$$

Where k is the Boltzmann constant, m is the mass of half a proton which is how much mass is being utilized and ω is the ratio between the central temperature and effective temperature of the sun.

Assuming that the sun is a black-body, I can determine the Luminosity since I know the values of radius R and temperature T

$$L(T,R) = 4\pi R^2 \sigma T^4 \tag{12}$$

Where σ is the Stefan-Boltzmann constant. From this, I have obtained all the physical proprieties initial values of my star by only giving the values of n to stimulate a certain star behavior, mass of the star M_* and the central density of the star ρ_c.
To stimulate the evolution of the sun, I will be using a time-dependence Luminosity function for the star that gives me the value of the Luminosity of the sun based on it's age.

$$L(t, M_*) = T_0(1 - \frac{5}{4}(\psi+1)(\frac{\mu T_0}{M_* Q})t)^{\frac{\psi}{\psi+1}} \tag{13}$$

Where ψ is a value of the wave function ,T_0 is the initial temperature of the star and Q is the energy per reaction of the star. The luminosity of the star at certain time-step allows me to calculate how many chemical reactions are happening per second within the star and tie that amount with a certain energy output that will tell me how much mass is loss within a certain time-frame.

The mass loss of the star will result in a different central density of the star which can be determined by manipulating the Lane-Emden equation with relation to K and the current mass M of the star due to the K value of the star being mainly be constant because it is within the main-sequence portion of it's life meaning it's the most stable and the change in K would be negligible:

$$\rho(M) = -(\frac{\frac{\frac{d\theta}{d\xi}K^{1.5}\xi_1^2\sqrt{\frac{n}{G}+\frac{1}{G}}}{2\sqrt{\pi G}} + \frac{\frac{d\theta}{d\xi}K^{1.5}n\xi_1^2\sqrt{\frac{n}{G}+\frac{1}{G}}}{2\sqrt{\pi G}}}{M})^{\frac{2n}{n-3}} \tag{14}$$

METHOD

Before delving into Methods, here are all the values for the constant variables in my programming portions.

```
#Constants
G = 6.67408e-11 #Gravitational Constant
Msolar = 1.99e30 #Solar Mass in kg
Rsolar = 6.957e8 #Solar Radius, in m
Lsolar = 3.842e26 #Solar Luminosity in Watts
tSun = 4.6e9 #Time of sun
denSun = 1.622e5 #Central Density of Sun
m = 0.84e-27 #Half a proton mass
#Mean Molecular Weight
(70% Hydrogen, 28% Helium, 2% Other) Ionized
mu = 0.62
k = 1.38064852e-23 #Boltzman Constant
sigma = 5.697367e-8 #Stefan-Boltzman Constant
#Rate Energy that is released in J/kg
Q = 6.0e14
psi = 15/2
energyFromReaction = 4.32e-12
secondInYear = 3.154e7
massPerReaction = 6.692e-27
```

Solving the Lane-Emden 2nd Differential Equation required a while-loop because I didn't know the number of time steps required to reach the maximum radius of the star. The time-step varied from different values of n. The code ran until right before the dimensionless radius became zero signifying the radius of the star, giving the constant ξ_1 which is the dimensionless length right before the radius was reached and $\frac{d\theta}{d\xi}_{\xi_1}$ which is the rate of radius change at the maximum point. I solve the Lane-Emden equation by giving it a value for central density ρ_c.

```
for n in nArray:

    #Boundary Conditions/Initial Values
    theta = 1.
    #Dimensionless Scaling Length
    dthetadxi = 0.

    #Setup
    xi = 0.0 #Dimensionless Radius
    dxi = 0.000001 #Time Step,
    #lower the more accurate it is

    #Arrays
    thetaArray = []
    thetaArray.append(theta)
    dThetaArray = []
    dThetaArray.append(dthetadxi)
    xiArray = []
    xiArray.append(xi)

    #Loop until Scaling Length
    #goes below time-step.
    #(Roughly right before it
    # becomes negative)
    while theta > dxi:
        if(xi == 0):
            dthetadxi -= (theta**n * dxi)
        else:
            dthetadxi -= ( 2 *
            dthetadxi/xi
            + theta**n) * dxi
        theta += dthetadxi * dxi
        xi += dxi

        thetaArray.append(theta)
        dThetaArray.append(dthetadxi)
        xiArray.append(xi)


    #Results
    print('\n--------------------','\
\nxi_1:',xi)
    print('dTheta/dXi_1:',dthetadxi)

    xi_1 = xi
    dthetadxi_1 = dthetadxi
```

The remaining physical properties can be achieved after achieving the constants for the polytrope above. The K will remain constant for the main-sequence star, so I can achieve density from the changing mass while keeping K the same value.

```
def kConstant(mass,density):
    k = (((2) * (2*np.pi)**(1/3) * G
    * mass**(2/3)
    * density**((1/3)*(1-(3/n))))\
    /   (dthetadxi_1**(2/3) *
    (n**3 + 3*n**2 + 3*n + 1)**(1/3)
    * xi_1**(4/3)))
    return k

def densityFromK(mass):
    return ((-((((dthetadxi_1)
    * K**(3/2) * xi_1**2
    * np.sqrt((n/G) + (1/G)))/(4 \
    * np.sqrt(np.pi) * G))+(((dthetadxi_1)
    * K**(3/2)
    * n * xi_1**2 * np.sqrt((n/G)
    + (1/G)))/(4 \
    * np.sqrt(np.pi) * G)))
    /mass)**((2*n)/(n-3))).real
```

Radius and Pressure can be achieved with the changing central density of the star through the manipulated equations:

```
def radius(density_c):
    return ((n+1)/(4*np.pi*G))**0.5 * K**(0.5)
    * density_c**((1-n)/(2*n)) * xi_1

def pressure(density):
    return K * density**(1+(1/n))
```

The remaining physical properties to achieve for the first star creation object so I can iterate it over a luminosity time-dependence function would be the initial temperature and luminosity of the star. These functions can be reused during the iteration to achieve the new values from the evolving star. $effRatio$ is currently a placeholder way of obtaining the Effective temperature of the star through the ratio between the central temperature and the effect temperature of the sun.

```
effRatio = 2722 #Ratio between
#Central Temperature and Effective Temp

def temp(density):
    return (((pressure(density)
    * mu*m * (K**n)
    * (k / (mu*m))**(-n))/k)**(1/(n+1))
    /effRatio).real

def Lum(temp,radius):
    return abs((4 * np.pi * radius**2
    * sigma * temp**4).real)
```

With Luminosity, I can derive a function that returns the mass from the new luminosity from the time-dependence function.

```
def L(t,mass):
    l = (initialLum) * (1 - (5/4)*(psi + 1)*
    ((mu * initialLum)/(mass * Q))*t)\
    **((-psi)/(psi+1))
    return abs(l)

def massLoss(lum,time):
    reactPerSec = lum/energyFromReaction
    #Time convert from seconds to years
    return reactPerSec * massPerReaction
    * (time * secondInYear)
```

Utilizing the derived equations above, I obtained a set of equations to describe and find the physical property of the sun at every time interval for an N number of steps up to any amount of years. The higher the number of time slices, the more accurate the approximation and stimulation will yield.

```
N = 1000 #Number of Time Slices
time_space = np.linspace(1,1e9,N)
#Time from 1 Year to 10 Billion Years

for t in time_space:

    #Mass Change from temperatrure difference
    lum = L(t,randMass)
    mass =
    starArray[len(starArray)-1].getMass()
    - massLoss(lum,t)
    centralDensity = densityFromK(mass)
    radiusStar = radius(centralDensity)
    centralPressure = pressure(centralDensity)
    temperature = temp(centralDensity)

    newStar
    = Star(mass,lum,radiusStar,temperature)
    starArray.append(newStar)
```

I then take the data of each star and plot it against the time-slices and I get a graph of the sun's physical properties changing in relation to time.
Simplifying my procedure of obtained the physical properties of the sun in relation to time:

- Solve the Lane-Emden equation for a given polytrophic index n that describes the star's behavior and get the constants ξ_1 and $\frac{d\theta}{d\xi}$.
- Give the star's mass and another parameter K which is a constant of proportionality between total mass and radius of the star or the star's central density ρ_c. Mass and central density will yield a value for K or Mass and K will yield a value for ρ_c with ξ_1 and $\frac{d\theta}{d\xi}$.
- With the values of M_*,K,ρ_c, solve for the R_* then pressure P_c.
- Assuming ideal gas, determine initial temperature of the star.
- Assuming star is a black-body, determine initial luminosity of star.
- Start looping through N time-slices and each time, ρ_c will change due to mass loss and the remaining physical properties can be determined again.
- Plot the results and see the trend the physical properties of the stars taken in relation to time.

RESULTS

Figure 1. shows the solution for the Lane-Emden ODE up to $n = 5$. The axis of dimensionless in the sense that will be scaled up depending on what your star is and what value of K and M_* given.

My Solutions		
n	ξ_1	$\frac{d\theta}{d\xi}$
0	2.4494800000075796	-0.8164933333361557
1	3.1415600000121136	-0.3183160507592273
2	4.35281000000438	-0.12725209559906353
3	6.896649999908076	-0.04243205765318847
3.25	8.0186699998656	-0.030323878794759708
4	14.970529999602418	-0.008019140282462962

Table 2-5 Constants of the Lane-Emden functions†

n	ξ_1	$-\xi_1^2\left(\frac{d\phi}{d\xi}\right)_{\xi=\xi_1}$	$\frac{\rho_c}{\bar{\rho}}$
0	2.4494	4.8988	1.0000
0.5	2.7528	3.7871	1.8361
1.0	3.14159	3.14159	3.28987
1.5	3.65375	2.71406	5.99071
2.0	4.35287	2.41105	11.40254
2.5	5.35528	2.18720	23.40646
3.0	6.89685	2.01824	54.1825
3.25	8.01894	1.94980	88.153
3.5	9.53581	1.89056	152.884
4.0	14.97155	1.79723	622.408
4.5	31.83646	1.73780	6,189.47
4.9	169.47	1.7355	934,800
5.0	∞	1.73205	∞

† S. Chandrasekhar, "An Introduction to the Study of Stellar Structure," p. 96; reprinted from the Dover Publications edition, Copyright 1939 by The University of Chicago, as reprinted by permission of The University of Chicago.

FIG. 1. Solutions to Lane-Emden

Figure 2. shows my results for N = 1.5 Lane-Emden ODE which is a star that behaves like a non-ionized gas in convective equilibrium achieved from the ODE method used which is very similar to the table of constants that I am suppose to get.

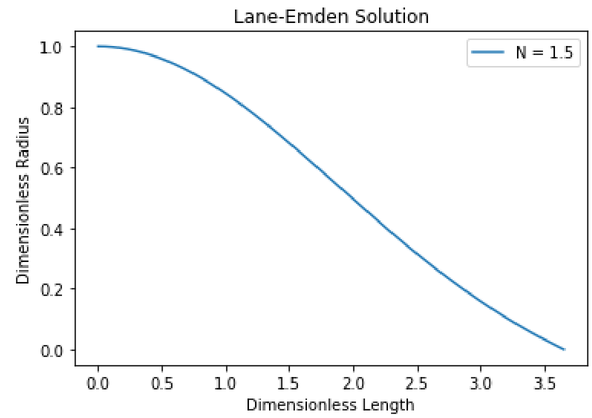

FIG. 2. Solution to Lane-Emden for N = 1.5

Figure 3.. is the solutions that I got for the Lane-Emden Equation with a polytropic index of 3.25 which was the value that I got the best approximation to the true values of the sun regarding to the sun's mean density. While **Figure 4.** is the remaining Lane-Emden equations that my program solved up to $N = 4$ with intervals of 1.

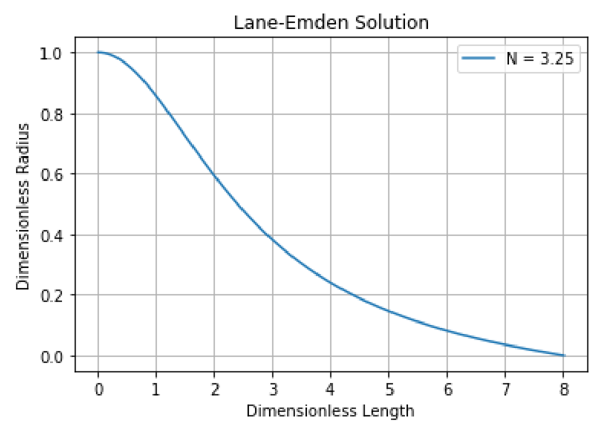

FIG. 3. Solution to Lane-Emden for N = 3.25

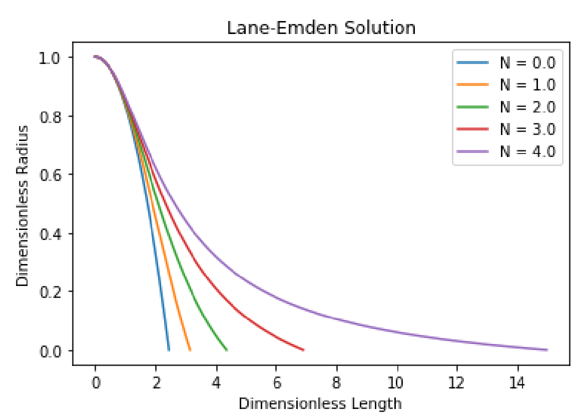

FIG. 4. Solution to Lane-Emden for N = 0 to 4

Once the constants for the Lane-Emden is solved for each N, it doesn't have to be re-calculated because it will remain the same alongside K. Only the central density changes that results in the time-dependence variable to change the other physical properties are different time-steps.

Figure 5. is the plot of the finished product of my program showing the physical properties of the sun throughout time. The effective temperature results are consistent with my predictions and how the sun's effective temperature will change with relation to time. As the sun evolves, it will need to use more fuel thus decreasing it's mass in order to counteract it's gravitational force with pressure from nuclear fusion. This explains it's logarithmic growth since the mass of the sun will decrease at a logarithmic growth due to have to use more and more mass to create more energy which would raise the effective temperature of the sun.

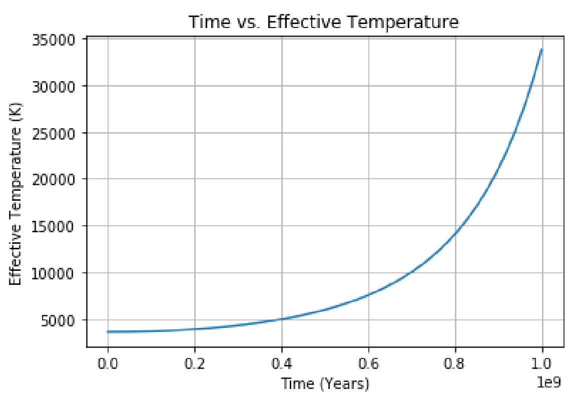

FIG. 5. Graph for the value of Effective Temperature in K as the Sun ages

Figure 6. is the plot of the Sun's luminosity in relation to time. This result is consistent with my prediction alongside the mass-luminosity relationship where it's linear just like the growth of luminosity in relation to mass loss.

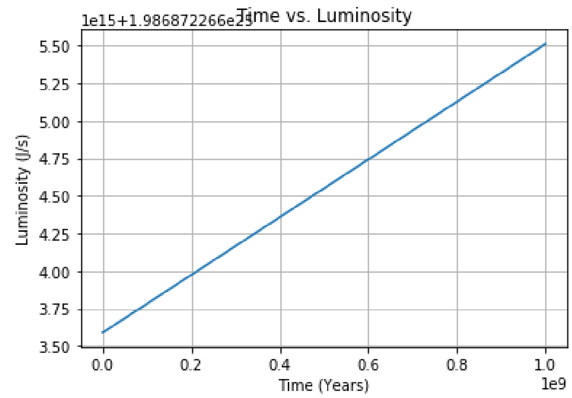

FIG. 6. Graph for the value of Luminosity in J/s as the Sun ages

Figure 7. is my result from how the mass changes with relation to time changes and it's just as I expected. The sun will lose mass at a logarithmic rate because it needs to generate enough energy to create a pressure outwards that stabilize with it's gravitational force. But the sun only has a limit supply of energy which is roughly 10% of it's mass and the sun has to create energy at faster rates as it ages because it's mass loss causing the radius of the star to increase making the sun less denser to lessen it's gravitational force to stop the itself from collapsing on it's own mass.

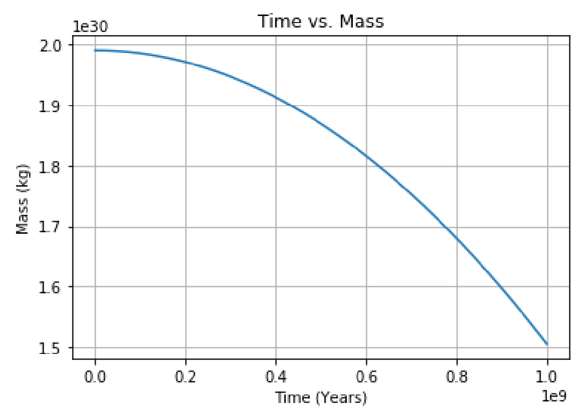

FIG. 7. Graph for the value of Sun's Mass in kilograms as the Sun ages

Figure 8. shows the final physical property of the sun that I intended to plot which is the radius of the sun in relation to time. As you can see, the radius seems to be constant but it's increasing at a very small rate but as time approaches 10 billion years, it grows exponentially. This happens because the sun's fuel has ran out and the sun starts swelling because it is now igniting hydrogen fusion outside of the core as an attempt to balance to it's gravitational force with immense pressure outwards. The sun will swell up in size up to 200 times it's current size which is about $0.01AU$ up to $2AU$.

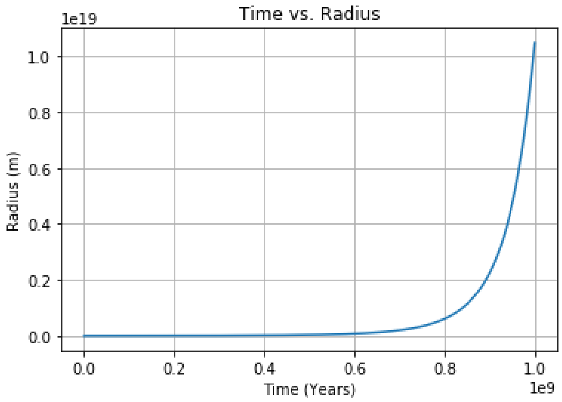

FIG. 8. Graph for the value of the Sun's Radius in meters as the Sun ages

DIFFICULTIES

The main difficulty of this project for me was the lack of generalized models and models for star types that was beyond main-sequence because initially, I wanted to simulate randomized stars with a random mass and temperature which would categorize into a specific type of star but there lacked known model for the behavior and evolutionary process for these stars. This lead to using the polytrope models which only work for main-sequence stars and I attempted randomization of stars with this model again but came to a halt as it turns out polytrope require a specific K or central density ρ_c and that is specific to every star. This result in the inability to stimulate randomized star's stellar evolutionary. This resulted me to settle for stimulating the evolutionary process of our sun because it's a star that we know some of the physical values of. I wanted to utilized a K or constant of proportionality of the sun but couldn't find one and ended up settling for the central density of the sun which was found on NASA's fact sheet for the sun.

Another difficulty with this project was that my calculation used very big numbers and resulted in a lot of complex numbers and Python would occasionally create hiccups and errors that result in unexpected or bad approximate results.

POSSIBLE IMPROVEMENTS

It is possible to stimulate the sun's evolutionary process beyond just it's beginning into becoming a Red Giant like following it until it truly becomes a dead star but there's no model that I found for anything beyond main-sequence. It can be done by calculating the energy production utilized by the star when it enter into being a Red Giant where it no longer utilizes hydrogen fusion as a energy source in the core. It's possible to calculate the hydrogen fusion that happens outside of it's core at certain layers within the Sun but the assumption of the sun as a polytrope will no longer be valid since it's layers are asymmetric.

It's also possible to model the sun with more complicated model such as the asymptotic giant branch model which takes metallicity of the star into account or the Eddington solar model that assumes that star is an symmetrical spherical incandescent ionized gas.

SUMMARY

Although I was unable to achieve the correct values for the physical property of the sun since I utilized assumptions that made my model and calculation much simpler; it still resulted in the trend that I predicted and wanted to see my calculation converge on. This error can be seen because the accepted radius of the sun compared to the radius I calculated with the assumptions made results in about a 40% error since I most accurate I wanted one parameter of my results to relate to the accepted value of the sun, the more inaccurate the other parameters become. Based on this problem, I went to optimize the parameter of mean density of the star closest to the accepted value of the sun, since that was the defining parameter that result changes in the other parameters. My goal of simulating the evolutionary track of the sun was successful because the trend of my results were consistent to how our current models predict the evolutionary process of the sun will undergo.

[1] W. Carroll, Bradley & A. Ostlie, Dale (2006). An Introduction to Modern Astrophysics (2nd Edition)

[2] W. Collins, George (2003). The Fundamentals of Stellar Astrophysics.
`http://ads.harvard.edu/books/1989fsa..book/`

[3] Michael Zingale (2003). PHY 521: Stars, Stellar Models.
`http://bender.astro.sunysb.edu/classes/`
`s̃tars/notes/models.pdf`

[4] O.R. Pols (2011). Stellar Structure and Evolution.

[5] N. Langer, Pennsylvania State University. Main Sequence Stars.

[6] Wang, Daniel (2010). Stellar Modeling.
`http://www.astro.umass.edu/ wqd/astro640/model.pdf`

[7] Rappaport, Saul (2006). Introduction to Astronomy - Polytropic Models for Stars.
`https://ocw.mit.edu/courses/physics/8-282j-introduction-`
`t̃o-astronomy-spring-2006/assignments/compset2`$_{s}oln.pdf$