Modelling a star

Term project 2

The second term project involves modelling the central parts of a Sun-like star, including both radiative and convective energy transport. The code developed in project 1 will be used for the energy production in your star[1].

The governing equations for solving the internal structure of the radiative zone of the Sun are as follows:

$$\frac{\partial r}{\partial m} = \frac{1}{4\pi r^2 \rho} \tag{1}$$

$$\frac{\partial P}{\partial m} = -\frac{Gm}{4\pi r^4} \tag{2}$$

$$\frac{\partial L}{\partial m} = \varepsilon \tag{3}$$

$$P = P_G + P_{\text{rad}} \tag{4}$$

$$\frac{\partial T}{\partial m} = \nabla \frac{T}{P} \frac{\partial P}{\partial m} \tag{5}$$

The reason m is the independent variable is that the set of equations is more stable than it would be if r was used. Note that Eq. (5) is for both radiative transport and convective transport. When dealing with radiative transport only, this equation can be reduced to

$$\frac{\partial T}{\partial m} = -\frac{3\kappa L}{256\pi^2 \sigma r^4 T^3} \tag{6}$$

You will notice that there are 7 unknown variables $(r, \rho, P, L, P_G, P_{\text{rad}}, T)$ but only 5 equations (assume that m, κ and ε are known since m is chosen,

[1] If you had problems with your code from project 1, reach out to the teaching assistant.

κ is picked from `opacity.txt`[2] and ε is calculated by using your code from Project 1). Thus two more equations are needed; an equation of state and an equation for the radiative pressure.

Below are the assumptions, goals, initial parameters, tasks and sanity checks, as well as instructions on the code and report. Note that there are exercises given in the Report section that are handy to do before starting on the tasks in the Tasks section. Read the description thoroughly before you start coding.

Assumptions

- All assumptions from Project 1.
- Ideal gas.
- No heat conduction in the star, only radiation and convection.
- The α_{l_m} in Eq. (5.84) can be assumed to be 1.

Goals

Your code should produce a star that:

- Has L, m and r all going to 0 or at least within 5% of L_0, M_0, and R_0, respectively.
- Has a core ($L < 0.995$) reaching out to at least 10% of R_0.
- Has a continuous convection zone near the surface of the star. The width of this convection zone should be at least 15% of R_0. A small radiation zone at the edge and/or a second convection zone closer to the centre is acceptable, but the convective flux should be small compared to the "main" convection zone near the surface.

[2]Taken from Asplund, M.;, Grevesse, N. and Sauval, A. J.; Astronomical Society of the Pacific, 2005., p.25

Initial parameters

Begin with the actual values of the solar surface (except P, which will differ because we are assuming an ideal gas). $\bar{\rho}_\odot$ is the average density of the Sun given by $\bar{\rho}_\odot = 1.408 \cdot 10^3$ kg m^{-3}. The initial parameters are the following:

$$
\begin{aligned}
L_0 &= 1.0 \cdot L_\odot \\
R_0 &= 1.0 \cdot R_\odot \\
M_0 &= 1.0 \cdot M_\odot \\
\rho_0 &= 1.42 \cdot 10^{-7} \cdot \bar{\rho}_\odot \\
T_0 &= 5770 \text{ K} \\
X &= 0.7 \\
Y_{^3_2\mathrm{He}} &= 10^{-10} \\
Y &= 0.29 \\
Z_{^7_3\mathrm{Li}} &= 10^{-7} \\
Z_{^7_4\mathrm{Be}} &= 10^{-7} \\
Z_{^{14}_7\mathrm{N}} &= 10^{-11}
\end{aligned}
$$

Tasks

- Create a method that reads the file `opacity.txt` and takes T and ρ as input and returns κ. The input and output parameters must be given in SI units. You will need to use linear 2D interpolation[3] for the common case where the input value is not exactly found in the opacity table. Your code should also be able to extrapolate if you are outside the bounds of the table (have your program output a warning when you do so). Test your method against the first sanity check below. The structure of the file is as follows:
  - The top row is $\log_{10}(R)$, where $R \equiv \frac{\rho}{(T/10^6)^3}$ and ρ is given in cgs units $\left(\frac{\text{g}}{\text{cm}^3}\right)$.
  - The first column is $\log_{10}(T)$, with T given in K.

[3] scipy.interpolate.RectBivariateSpline is useful for this

  - The rest of the table is $\log_{10}(\kappa)$ given in cgs units $\left(\frac{\text{cm}^2}{\text{g}} \right)$.

- IF you have failed at solving project 1, you can identically create a method that reads the file `epsilon.txt` and takes T and ρ as input and returns the ε in Eq. (3), the total energy produced by the star per unit mass per unit time[4] (not including the energy lost to neutrinos). The structure of the file is as follows:

  - The top row is $\log_{10}(R)$, where $R \equiv \frac{\rho}{(T/10^6)^3}$ and ρ is given in cgs units $\left(\frac{\text{g}}{\text{cm}^3}\right)$[5].
  - The first column is $\log_{10}(T)$, with T given in K.
  - The rest of the table is $\log_{10}(\varepsilon)$ given in cgs units $\left(\frac{\text{erg}}{\text{g s}} \right)$.

- Implement methods to calculate $\rho(P, T)$ and $P(\rho, T)$.

- Solve exercises 5.11-5.13 to obtain an expression for the convective flux, and through that get the temperature gradient.

- You are now ready to begin on the main body of the program. Solve the four partial differential equations numerically. You are free to decide on which numerical solver you want to use (Euler, Runge-Kutta, Simpson's method etc.).

- Include a check for convective stability at each mass shell. If the shell is convectively stable, you need to use Eq. 6 (radiative transport) for the temperature. Otherwise, you need to use the expression from exercise 5.13 (convective transport).

- Implement dynamic step size (see `variablesteplength.pdf`).

- See "Sanity checks" for instructions on how to make sure your program is working properly and "Report" for details about the report.

[4]Note that it does not give you ϵ_{CNO}, etc. so you will not be able to solve all the tasks in this project without first solving project 1.

[5]Note that this table includes more numbers for R than the table for κ does.

Sanity checks

The following sanity checks need to be implemented in your code, and it must be possible to turn them on and off so that the instructors can verify your code. You do not need to include values and figures from the sanity checks in your report. 7.5 points

Opacity table

Check that the interpolation of opacity values give results close to the following values:

$\log_{10} T$	$\log_{10} R$ (cgs)	$\log_{10} \kappa$ (cgs)	κ (SI)
3.750	-6.00	-1.55	$2.84 \cdot 10^{-3}$
3.755	-5.95	-1.51	$3.11 \cdot 10^{-3}$
3.755	-5.80	-1.57	$2.68 \cdot 10^{-3}$
3.755	-5.70	-1.61	$2.46 \cdot 10^{-3}$
3.755	-5.55	-1.67	$2.12 \cdot 10^{-3}$
3.770	-5.95	-1.33	$4.70 \cdot 10^{-3}$
3.780	-5.95	-1.20	$6.25 \cdot 10^{-3}$
3.795	-5.95	-1.02	$9.45 \cdot 10^{-3}$
3.770	-5.80	-1.39	$4.05 \cdot 10^{-3}$
3.775	-5.75	-1.35	$4.43 \cdot 10^{-3}$
3.780	-5.70	-1.31	$4.94 \cdot 10^{-3}$
3.795	-5.55	-1.16	$6.89 \cdot 10^{-3}$
3.800	-5.50	-1.11	$7.69 \cdot 10^{-3}$

Epsilon table

IF you use the epsilon table instead of using your own code from project 1, check that the interpolation of epsilon values give results close to the following values:

$\log_{10} T$	$\log_{10} R$ (cgs)	$\log_{10} \varepsilon$ (cgs)	ε (SI)
3.750	-6.00	-87.995	$1.012 \cdot 10^{-92}$
3.755	-5.95	-87.267	$5.401 \cdot 10^{-92}$

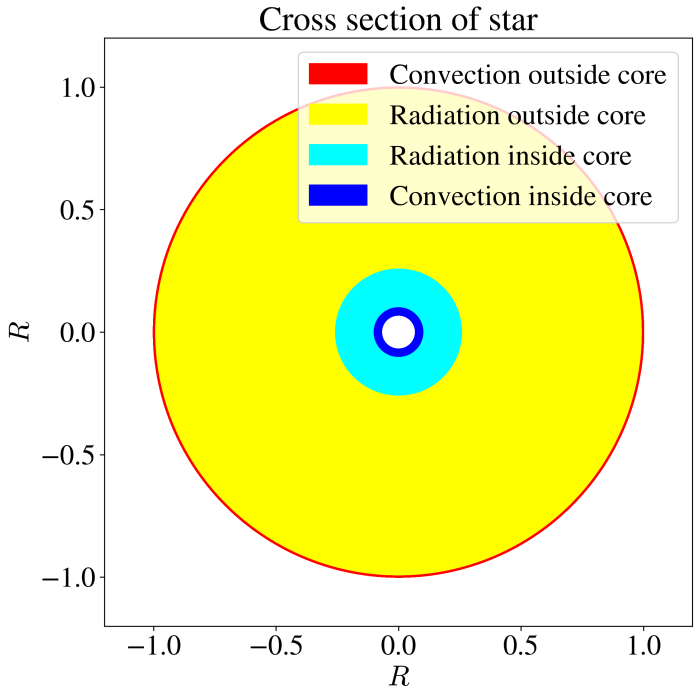
Gradients

Implement the sanity check in example 5.1 in the notes and see that you get values close to the ones presented there. Use the actual μ calculated in your

code - it should be close to 0.6.

Model verification

The cross-section of your star (see `cross_section.py`) should look like this (note the empty space in the middle since r does not reach 0 in this case):

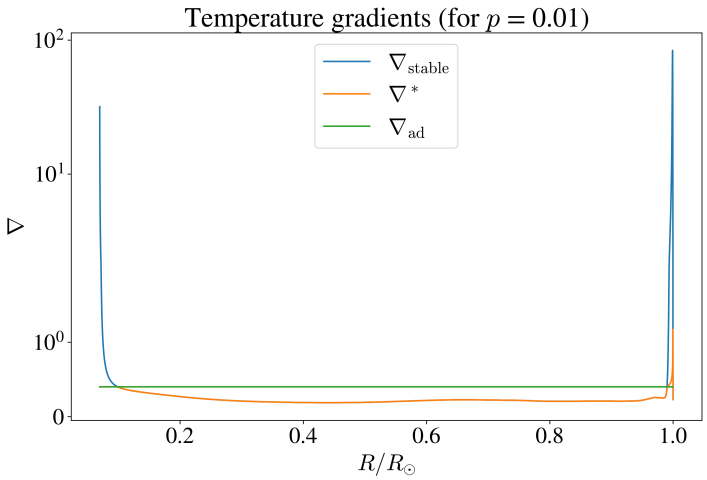


Use coloured circles with no fill and the following colour codes:

- Red: Outside core ($L \geq 0.995 L_\odot$) and convection ($F_C > 0$).
- Yellow: Outside core ($L \geq 0.995 L_\odot$) and radiation ($F_C = 0$).
- Blue: Inside core ($L < 0.995 L_\odot$) and radiation ($F_C = 0$).
- Cyan: Inside core ($L < 0.995 L_\odot$) and convection ($F_C > 0$).

Draw a circle for every 50 steps or so, otherwise your program will slow down significantly when plotting.

Using the given initial conditions, you should get a temperature gradient plot looking like this:



Code

The code has to be written using the `python3` programming language. The way your code is written impacts the number of points you get for this project. It should be easy to read, well commented and logically structured. The instructors should be able to run your code. 7.5 points

Report

You are required to write a report in this project. The report should include an introduction, method, results, discussion and conclusion. The maximum number of pages is 10 (excluding front page and references). How your report is written impacts the amount of points you get on this project. It is recommended that you write the report using LaTeX. Use a spellchecker to avoid spelling mistakes and typos. Your figures should have a clear layout with proper axis labels and units, with a caption explaining what it shows. All figures should have a reference in the main text. Make sure that you incorporate the feedback you received from your Project 1 report.

You should include answers and explanations to as many as possible of the following bullet points. Start from the top and work your way down.

1. Explain the governing equations. 5 points
2. Describe how you calculate the average molecular weight μ. 5 points

3. Solutions to exercises 5.11-5.13 (you do not need to refer to them by exercise number). You should do (almost) all exercises in Chapter 5 before solving these. 10 points

4. Before you start changing parameters to find your best model, you should investigate what happens to your star when you do the following:

   - Change the value of R_0 (keeping all other initial parameters unchanged).
   - Change the value of T_0 (keeping all other initial parameters unchanged).
   - Change the value of ρ_0 (keeping all other initial parameters unchanged).
   - Change the value of P_0 (keeping all other initial parameters unchanged).

   Hint: Try changing ρ_0 and P_0 up by several orders of magnitude.

   Run sufficient experiments for each of these, and plot the results. Explain why the changes happen. Does changing some of the parameters produce very similar effects? Can the effects be explained with the given equations? Comment on which of the initial parameters you need to change to create a wider convection zone in your star. 10 points

5. Create a model where the goals presented in the beginning of the project description are reached. You are free to change m_0, r_0, L_0 and T_0, but no parameter should change more than a factor 5. Additionally, you can change ρ_0 as much as want, but within reason. From your best model, include the following:

   a) What parameters did you end up changing? Present the values of the initial parameters of your best-fit model for m_0, r_0, L_0 and ρ_0 in units of $M_\odot$, $R_\odot$, $L_\odot$ and $\bar{\rho}_\odot$, respectively. The final temperature T_0 can be given in K. 2.5 points

   b) The main parameters ($m/M_\odot$, T, $L/L_\odot$, $\rho/\rho_\odot$ and P) plotted as functions of radius $r/R_\odot$ (use logarithmic axes for ρ and P). Consider sub-plots in order to save space. 7.5 points

c) Plot of the fractions of energy being transported by convection and radiation (F_{con} and F_{rad}), both as functions of radius. 7.5 points

d) Plot of the relative energy production from PPI, PPII, PPIII and CNO as functions of radius. Include $\varepsilon(r)/\varepsilon_{\max}$ in order to estimate how much of each branch contributes to the overall luminosity, where $\varepsilon_{\max}$ is the sum of all energy produced from PPI, PPII, PPIII and CNO at a given radius r. How does this compare to the temperature plot in Project 1? 10 points

e) Plot of ∇^*, ∇_{stable} and ∇_{ad} as functions of radius. Use a logarithmic axis for the ∇'s. 7.5 points

f) A cross-section of your star (using the same code as in the sanity check). Compare to a cross-section of the actual Sun. What is different, and what is similar? 5 points

6. Your report should be well structured and easy to read. Include a summary in your conclusion where you explain what you have learned from this exercise and if you had any trouble on the way. 15 points

The maximum score of this project is 100 points. Deliver your report as a pdf together with the `python` files needed to run your code at `https://devilry.ifi.uio.no/`. Submit separate files, do not combine them as a single .zip, .tar or .rar file. There will be no extension to the deadline, except in the case of documented medical circumstances. If you cannot solve the project, write the report anyway, explaining your problems and what you have tried in order to solve them.