Draft version December 16, 2023
Typeset using LaTeX **twocolumn** style in AASTeX631

Stellar Interiors Numerical Assignment

Ted Johnson

1. INTRODUCTION

We will construct a polytropic model of a stellar interior using polytropic index $0 \leq n \leq 4$. Following Section 7.2.2 of Hansen et al. (2004), we cast the Lane-Emden equation into two first-order differential equations before numerically solving with a fourth-order Runge-Kutta integrator.

The integrator is implemented in native Python in the repository linked at the top of the page (see `polysolver/`). It is also re-implemented in Rust (`rust/`). In addition to learning about numerical solvers and polytropes, this project allowed me to experiment with Rust-Python bindings and with show your work! (Luger et al. 2021), which makes it possible for this document (and all the source code associated with it) to be open-source and completely reproducible.

Following Hansen et al. (2004) we will focus on the profile $\theta_n(\xi)$ as well as the location of the surface ξ_1, the derivative of θ_n at the surface $\theta'_n(\xi_1)$, and the central density in units of the bulk density $\rho_c / < \rho >$.

Section 2 describes the code and it's outputs, Section 3 examines the effects of numerical resolution on the results, and Section 4 describes the characteristics of polytropic solutions as a function of polytropic index.

2. METHODS

Most of the work is done by the Python function

```
polysolver.solve(
    x_int,n,h,
    max_iter=1000,impl='rust'
)->x,y,z
```

The details of the code's backend can be easily read from the source code itself, but it essentially starts with some values x, y, and z and then computes the next value for each using a 4th-order Runge-Kutta integrator. It stops when the condition $y \leq 0$ is met or after `max_iter` iterations.

These three variables are defined by Hansen et al. (2004) as

$$x = \xi$$

$$y = \theta_n$$

$$z = \frac{\mathrm{d}\theta_n}{\mathrm{d}\xi}$$

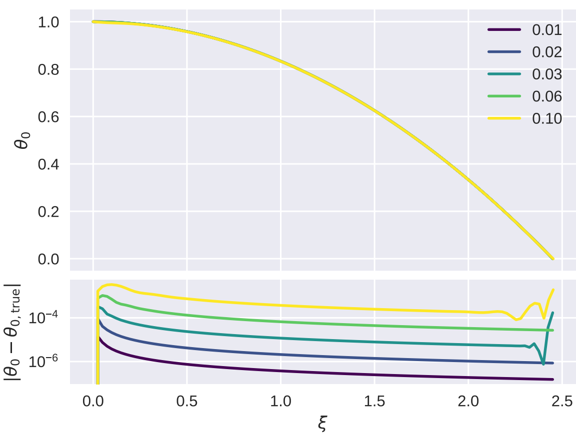

Figure 1. Polytropic curve for $n = 0$ for various values of h. An analytic solution is used as the ground truth. Note that the residuals are highest near the boundaries.

We also define ξ_1 as the value of ξ when θ_n is zero (i.e. the surface).

`x_init` is the initial value of x and must be set close to zero. However, it cannot be zero because at zero the derivative of z goes to infinity. Unless stated otherwise, we set `x_init` to 10^{-20}.

`n` is the polytropic index of the model. We explore $0 \leq n \leq 4$.

`h` is the step size. This is an important parameter that we will explore below. Ideally this parameter should be less than the pressure scale height at all points.

`max_iter` and `impl` are optional parameters that do not effect the mathematics of the model. `max_iter` prevents an infinite loop, and should be set high enough that the condition $y \leq 0$ is met. `impl` is the solver implementation that will be used. The code has been implemented in both Python and Rust.

3. RESOLUTION STUDY

The step size parameter h determines the resolution of our integration in units of ξ. A more sophisticated algorithm might choose a variable step size to improve accuracy, but we hold this parameter constant for each integration.

A major flaw in our choice of constant h is that the likelyhood that the program terminates when $\theta_n = 0$ is very small. Except when h and `x_init` are chosen very

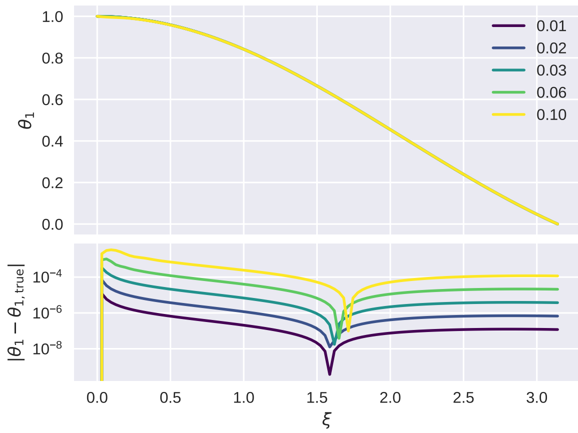

Figure 2. Polytropic curve for $n = 1$ for various values of h. An analytic solution is used as the ground truth. Note the local minimum at $\xi \sim \pi/2$. There is an inflection point here, and our numerical solution crosses the true value. The location of this minimum approaches the inflection point as h approaches zero.

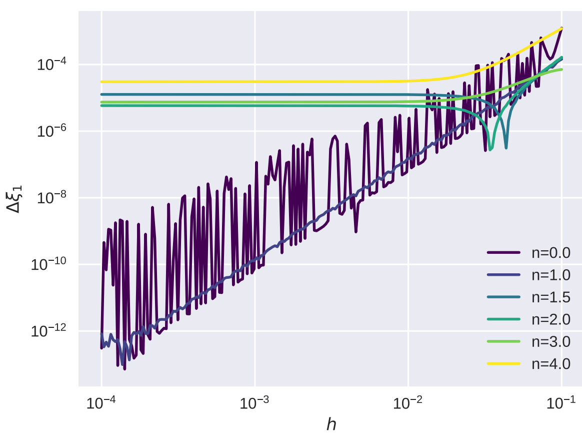

Figure 3. Deviations of ξ_1 from the true value as a function of h. Polytropes without analytic solutions quickly converge as h approaches zero, while the $n = 0$ and $n = 1$ solutions continue to imporove indefinitely. It is expected that these values would eventually be limited by machine precision or `x_init`. They y-axis is defined as $\Delta q = \frac{|q - q_{\mathrm{true}}|}{q_{\mathrm{true}}}$.

carefully, ξ_1 will end up between grid points – producing additional error for the value of our solution at the surface.

To mitigate this effect we can modify our equation for z'

$$z' = \begin{cases} -y^n - \frac{2}{x} z & \text{if } y \geq 0 \\ (-y)^n - \frac{2}{x} z & \text{if } y < 0 \end{cases}$$

This way z' is continuous and real as y crosses zero. We then record one additional point $\xi > \xi_1$ and use interpolation to determine the surface value. We choose

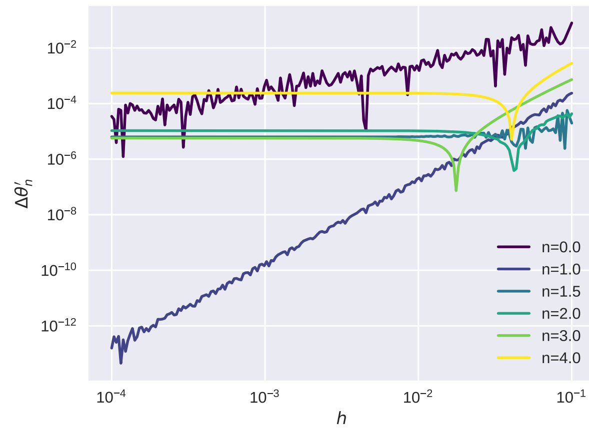

Figure 4. Same as Figure 3, but for $\theta_n'(\xi_1)$.

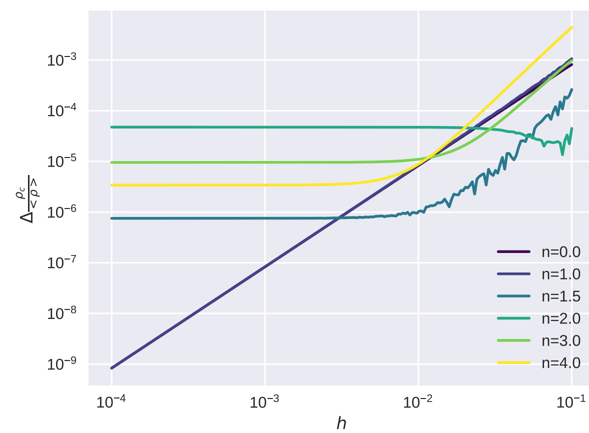

Figure 5. Same as Figure 3, but for $\rho_c/ < \rho >$. The curve for $n = 1$ is mostly covered by the $n = 0$ curve.

a cubic spline interpolation of the final three points for a balance between accuracy and speed.

Figures 1 and 2 show the polytropic curve for $n = 0$ and $n = 1$ for various values of h. The curve shown is a resampled solution from `x_init` to $\xi_{1,\mathrm{measured}}$ using a cubic spline. The resampling is done so that the value of θ_n at the surface can be shown most accuratly.

We can also discuss the effects of h on our measured quantities ξ_1, $\theta_n'(\xi_1)$, and $\rho_c/ < \rho >$. Hansen et al. (2004) gives values for these quantities for $n = 0, 1, 1.5, 2, 3, 4$ in Table 7.1. Deviations of our numerical results from these given quantities fall into a few regimes:

1. Dominated by interpolation errors. This is our uncertainty due to h.

2. Dominated by the finite precision of the true value. Once within the precision of the value given by

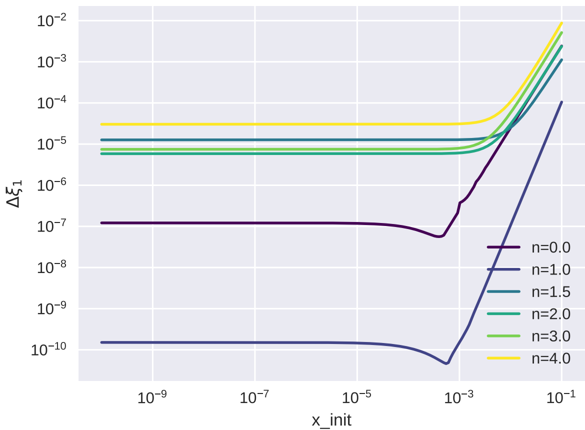

Figure 6. Deviations of ξ_1 from the true value as a function of `x_init`. As expected, when `x_init` is large $\Delta\xi_1$ is a function of `x_init`, and the error is independent of `x_init` when it is small. This relationship holds for arbitrary n, confirming that for small h and small `x_init` the numerical error we measure is dominated by the true value's uncertainty.

Hansen et al. (2004), improving the resolution no longer yields a better agreement.

3. Dominated by the value of `x_init`. Since we cannot start our integration at $\xi = 0$, our numerical solution will converge differently depending on the location of this boundary.

We expect Regime 1 to be the main source of error for large h. Figures 3, 4, and 5 show that at large h the residuals are indeed a function of h, but for the case of general n our model converges when h is sufficiently small. The fact that we do not see convergence for $n = 0$ and $n = 1$ suggests that the convergence of residuals in the general case is due completely to finite knowledge of the true value (i.e. Regime 2). If it were the case of Regime 3 then the solutions that are known analytically would also converge to some near-true value and we would see the same behavior for all n.

Herein we choose a value of $h = 10^{-3}$ based on Figures 3, 4, and 5.

3.1. *The effects of* `x_init`

Let us quickly examine the effects of our lower boundary on our numerical results. Figure 6 shows the deviations of ξ_1 similar to Figure 3, but in this case our independent variable is `x_init`. For every solution there is a value of `x_init` after which there is no improved accuracy. This value appears to be 10^{-3}, but that is also the value we set for h.

Figure 9 examines the joint effects of `x_init` and h. We do find that for $n = 0$ and $n = 1$ `x_init` should be at a maximum h, but for general n we become dominated by Regime 2 when `x_init` is less than $\sim 10^{-2}$. However, unlike h, we can choose arbitrarily small `x_init` without sacrificing computational resources, so we will continue with the value 10^{-20}.

n	ξ_1	exp.	$-\theta_n(\xi_1)$	exp.	$\rho_c/<\rho>$	exp.
0	2.4495	2.4495	0.81589	0.8165	1	1
0.25	2.5921	-	0.63371	-	1.3635	-
0.5	2.7527	-	0.5	-	1.8352	-
0.75	2.9345	-	0.39809	-	2.4571	-
1	3.1416	3.1416	0.31831	0.31831	3.2899	3.2899
1.25	3.3791	-	0.25468	-	4.4226	-
1.5	3.6538	3.6538	0.2033	0.2033	5.9907	5.9907
1.75	3.9744	-	0.16147	-	8.2047	-
2	4.3529	4.3529	0.12725	0.12725	11.403	11.402
2.25	4.8055	-	0.099211	-	16.146	-
2.5	5.3553	-	0.076265	-	23.406	-
2.75	6.0356	-	0.057562	-	34.951	-
3	6.8968	6.8969	0.04243	0.04243	54.182	54.183
3.25	8.0189	-	0.030322	-	88.153	-
3.5	9.5358	-	0.020791	-	152.88	-
3.75	11.69	-	0.013462	-	289.46	-
4	14.972	14.972	0.0080181	0.00802	622.41	662.41

Table 1. Relevent quantities for various polytropic solutions to the Lane-Emden equation. The column exp. lists the value for the previous column given by Hansen et al. (2004).

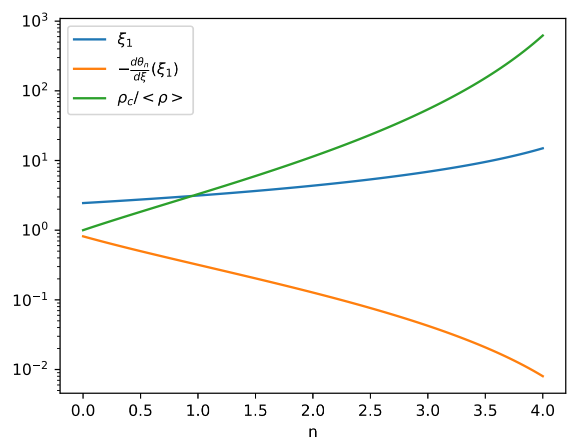

Figure 7. Quantities ξ_1, $-\theta_n'(\xi_1)$, and $\rho_c/<\rho$ as a function of polytropic index n. As n increases our solution flattens out near the surface. The radius, however, does not increase by very much, so we see a concentration of material neart the core.

4. RESULTS

Now that we have an established understanding of the errors in our study we will look at our numerical results for arbitrary n. As stated in Section 3, we choose a

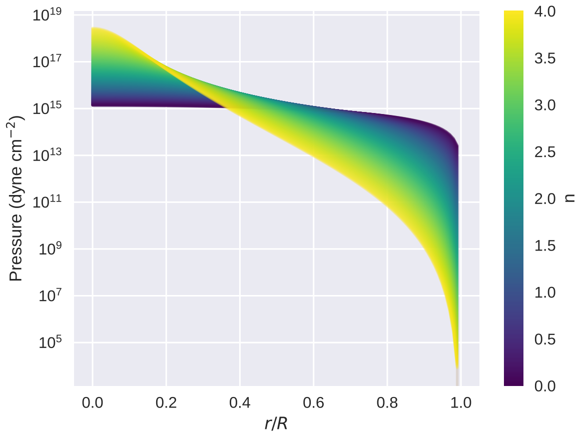

Figure 8. Pressure profile as a function of polytropic index n. The x axis is normalized to the star's radius while the central pressure has been calcualted assuming the star has the mass and radius of the sun.

value of $h = 10^{-3}$ and set `x_init` to 10^{-20}. Table 1 gives these results in the style of Table 7.1 in Hansen et al. (2004).

Figure 7 shows the quantities ξ_1, $-\theta_n'(\xi_1)$, and $\rho_c/<\rho>$ as continuous functions of polytropic index n. Figure 8 shows the pressure profile as a function of polytropic index n. These two figures clearly demonstrate that polytropic index is analogous to the degree to which a star is centrally condensed.

Figure 8 shows that for models with near-constant density (i.e. low n) that pressure changes very little until very near to the surface. The centrally condensed models, however, have a very steep pressure profile because less pressure is needed to support the diffuse outer layers.

APPENDIX

A. ADDITIONAL FIGURES

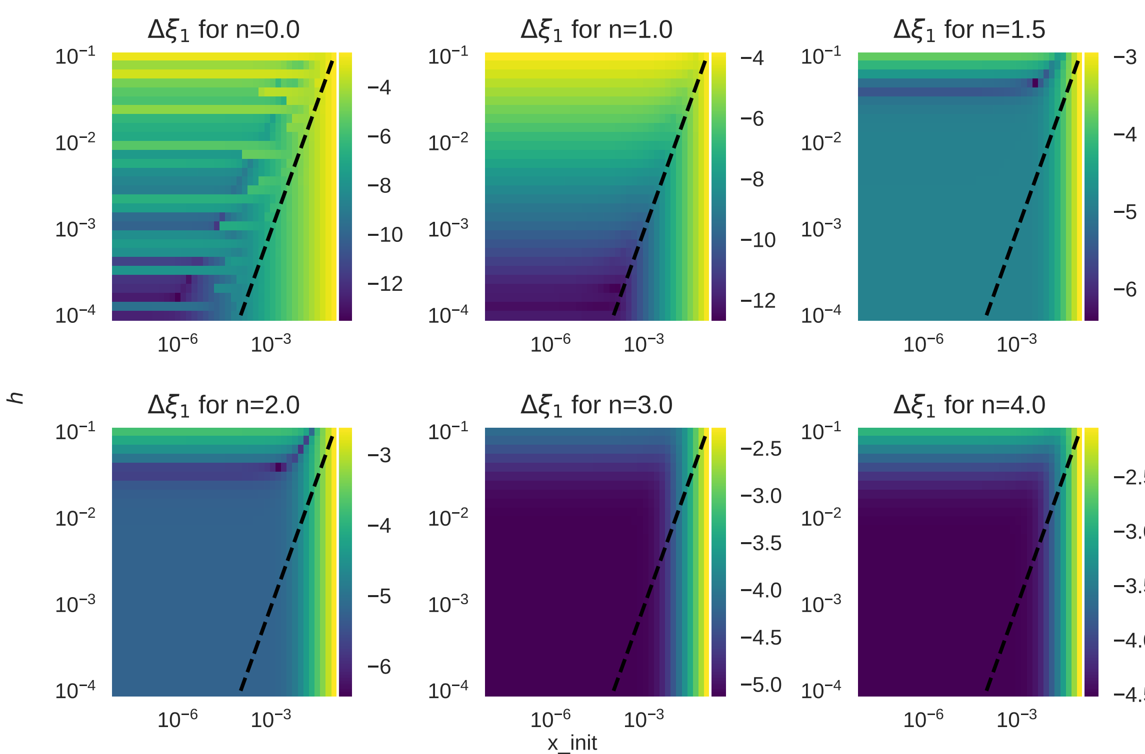

Figure 9. Deviations of ξ_1 from the true value as a function of both `x_init` and h. The dashed line marks the curve `x_init`$= h$.

REFERENCES

Hansen, C. J., Kawaler, S. D., & Trimble, V. 2004, Stellar interiors : physical principles, structure, and evolution

Luger, R., Bedell, M., Foreman-Mackey, D., et al. 2021, arXiv e-prints, arXiv:2110.06271. https://arxiv.org/abs/2110.06271