Creation of Multiple Random Equation of States based on the Speed of sound model

Andreas Konstantinou

June 2021

1 Introduction

Here you can find a way to create a set of random Equations of State (EOS). The purpose of this document is not to go over the theory and the methods that have been used by the authors, but it has as a goal to give you an idea for these methods and show you how you can use our codes. If you want to learn more about the EOS models you can take a look at the references that we provide. In these notes we will be using the notation as listed below:

Symbol	Quantity	Units
P	Pressure	dyne cm^{-2}
ϵ	Energy density/c^2	g cm^{-3}
ρ	Mass density	g cm^{-3}
h	Specific enthalpy	$(cm/s)^2$
N	Relativistic enthalpy	$(cm/s)^2$

2 Speed of Sound method ($SoSm$)

This method is based on the speed of sound method (The detailed process can be found in [1]). The main idea for this method is pretty simple, once I know an equation that relates the square of the speed of sound with the energy density I can integrate with respect to the energy density and find the Pressure (and therefore the EOS).

$$c_s = \sqrt{\frac{dP}{d\epsilon}} \tag{2.1}$$

$$P(\epsilon) = \int_0^{\epsilon} (c_s(\epsilon')/c)^2 d\epsilon' \tag{2.2}$$

For the crust we can use one of the pre-existed EOSs ([2] (eosNV), the [3](eosFPS), or the [4] (eosBPS)). We use the crust up to 0.5 n_0 (where $n_0 = 0.16$ fm^{-3}). After this point, we add a chiral Effective Field Theory (cEFT) band that is provided by [5]. The cEFT band takes place from 0.5 n_0 up to 1.1 n_0. Beyond that point the authors by using: a) the fact that the speed of sound has to be less than the speed of the light, b) the fact that the EOS has to support the heaviest observed neutron star and c) the fact that perturbative quantum chromodynamics (**pQCD**) predict that $\frac{c_s}{c}$ approaches $\frac{1}{\sqrt{3}}$ from below at very high densities they define the $c_s^2(\epsilon)$ equation as:

$$(\frac{c_s}{c})^2 = \alpha_1 \exp\{-0.5(x-\alpha_2)^2/\alpha_3^2\} + \alpha_6 + \frac{1/3 - \alpha_6}{1+\exp(-\alpha_5(x-\alpha_4))} \tag{2.3}$$

Here x $\equiv \epsilon/(m_N n_0)$ and m_N is the nucleon mass and is equal to 939.565 MeV. The range of possible values for the five coefficients is given in [1] paper. α_6 can be found by matching the c_s at low densities (1.5 n_0) to the c_s of the upper or the lower point of the cEFT band (for the 50% of the cases we use the upper point and for the rests the lower one). Therefore, the EOS profile of the SoSm is going to look as follows:

$$P(\epsilon) = \begin{cases} P_{crust} & n \leq 0.5n_0 \\ P_{cEFT} & 0.5n_0 \leq n \leq 1.1n_0 \\ \int_0^{\epsilon}(c_s(\epsilon')/c)^2 d\epsilon' & 1.1n_0 \leq n \end{cases} \tag{2.4}$$

Based on that, we take random values within this given range and we create a set of random EOSs.
Keep in mind that for the region where baryon density takes values from 1.1 n_0 up to 1.5 n_0 we also take into account the fact that the fluid is going to behave as a Fermi liquid. For this reason the authors made the conservative assumption that within this region the speed of sound has to be less than $\sqrt{0.163}$ c. If this is not the case we discrad the produced EOS.

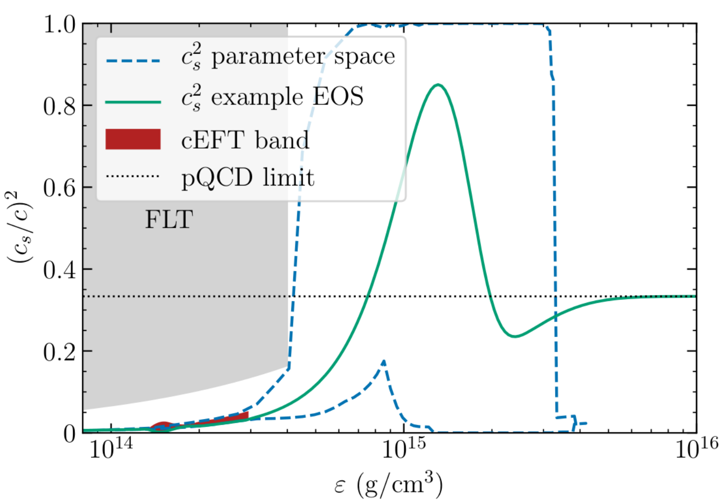

Figure 2.1: Speed of sound profile from [1]

3 The code

3.1 Compiling the code

The code is written in C++, but it uses some tools that come from ROOT which is a Data Analysis Framework that has been developed by CERN [6]. To run the code, you have to download Root and then you have to type the word "root" followed by the name of the file in the terminal window (i.e "root greif.cpp"). Once it runs, the code asks from the user the number of the random EOSs that he or she wants to create.
"Give the number of EOS"
If you give the number and press "Enter" the code will run by using the predefined parameters.

3.2 The outputs

Once the code finishes with the compiling process, it is going to create a number of EOS files (which depends on user's choice).

The name of the file is "eosGreif+number". In the first line of the table we give the number of inputs. Below that, you are going to find 4 columns. The first column is the energy density in g/cm^3, the second is the Pressure in $dyne/cm^-2$, the third is the Relativistic enthalpy in cm^2/s^2 and the last one is the baryon density in $1/cm^3$ (is the mass density divided by 1.66 10^{-24} g). The file is going to look as follows:

80			
7.8	1.01e+08	1	4.6988e+24
7.86	1.01e+09	1.15795e+08	4.73494e+24
7.9	1.01e+10	1.26945e+09	4.75904e+24
7.9	1.01e+10	1.26945e+09	4.75904e+24
.	.	.	.
.	.	.	.
.	.	.	.

Also, a file with the name "EosTable" is created, and includes 9 columns. In each column we have the name of the EOS, the 6 coefficients α_1 - α_6, the central energy density (g/cm^3), and the side of the cEFT band that is been used.

Furthermore, the code solves the TOV equations by using the Runge-Kutta4 method. In file "eosGreif+number+table" we provide the Mass and Radius of a neutron star for different values of the central energy density. This file is composed by 3 columns. The first one is the mass in $M_\odot$, the second one is the Radius in km and the last one is the central energy at the center in g/cm^3.

Finally, the code creates a plot for the Mass-Radius curves with the name "Mass_Radius.png". You can plot the EOSs by using the "plotGr.cpp" code. Once you have the EOS files you type "root plotGr.cpp" in the terminal and the code is going to ask you for the number of the EOS files. The code will give you back a plot with the name "plotEOS.png".

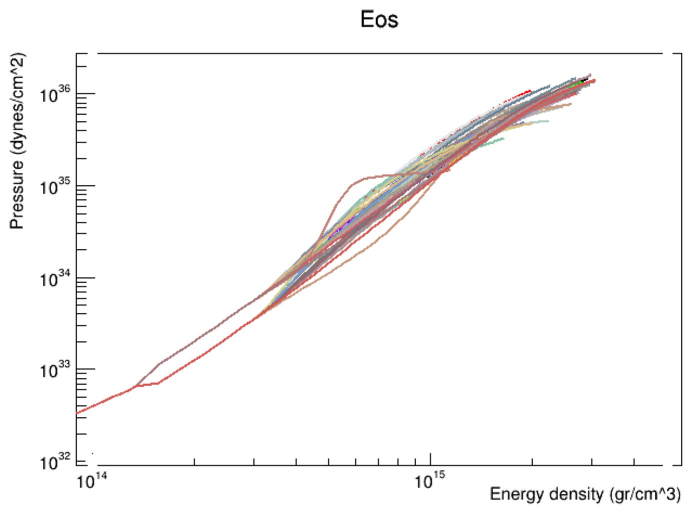

Figure 3.1: plotEOS.png

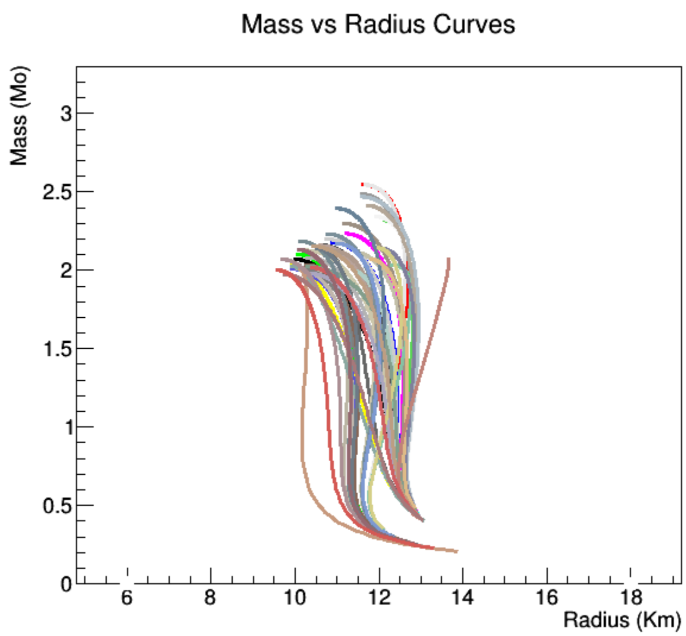

Figure 3.2: "Mass_Radius.png"

3.3 How to change the pre-defined parameters

The main program starts at line 216 "void greif()". Before that point someone can find the functions that we used in our code. At line 289 you can choose the crust EOS by changing the value of the "sEOSnum" parameter. 0 is the NV EOS, 1 is the FPS EOS and 2 is the BPS EOS. You can change the mass of the heaviest observed neutron star at "Max_mass_limit". All the EOSs can support a star with such mass. Next, you can choose the value of the crust density border by modifying the if statement at line 309.
Finally, at line 362 you can change the values of the range of possible values of the 6 coefficients.

4 Acknowledgements

I would like to thank Sharon Morsink for suggesting this project and supervise my work.

References

[1] S K Greif et al. "Equation of state sensitivities when inferring neutron star and dense matter properties". In: *Monthly Notices of the Royal Astronomical Society* 485.4 (Mar. 2019), pp. 5363–5376. ISSN: 1365-2966. DOI: 10.1093/mnras/stz654. URL: http://dx.doi.org/10.1093/mnras/stz654.

[2] J. W. Negele and D. Vautherin. "Neutron star matter at sub-nuclear densities". In: 207.2 (June 1973), pp. 298–320. DOI: 10.1016/0375-9474(73)90349-7.

[3] C. P. Lorenz, D. G. Ravenhall, and C. J. Pethick. "Neutron star crusts". In: *Phys. Rev. Lett.* 70 (4 Jan. 1993), pp. 379–382. DOI: 10.1103/PhysRevLett.70.379. URL: https://link.aps.org/doi/10.1103/PhysRevLett.70.379.

[4] M.Baldo, I.Bombaci, and G.F.Burgio. "Microscopic nuclear equation of state with three-body forces and neutron star structure". In: 1997.

[5] K. Hebeler et al. "EQUATION OF STATE AND NEUTRON STAR PROPERTIES CONSTRAINED BY NUCLEAR PHYSICS AND OBSERVATION". In: *The Astrophysical Journal* 773.1 (July 2013), p. 11. ISSN: 1538-4357. DOI: 10.1088/0004-637x/773/1/11. URL: http://dx.doi.org/10.1088/0004-637X/773/1/11.

[6] ROOT. *About ROOT*. URL: https://root.cern/about.