Interim Report 2: Testing Universal Relations Under Non-Parametric Equation of State Models

Bubakar O. Sy-Garcia
(Dated: July 28, 2022)

Neutron stars have long been a point of interest in astronomy due their extreme qualities. Of these, the core of these super-dense star remains of especially large interest. Due to the extremely high densities that are present inside of these objects, the way matter moves and acts is unknown. At the Laser Interferometer Gravitational-Wave Observatory (LIGO), this knowledge is very important as it would give data we collect on gravitational waves much more power in terms of deducing properties of the neutron stars that cause them. We hope to be able to, through the course of this project, see if possible relations between gravitational wave data we collect and neutron star properties can be found without necessarily knowing how matter acts in these super-dense states with what we call "Universal Relationships". We aim to test these relationships and see if they truly hold regardless of the Equation of State (EoS) that controls the relation between pressure and density in neutron stars rigorously so as to ascertain their validity and reliability. With this, we will be better able to collect and restrict neutron star measurements if these relations prove to be fruitful in their utility, and gain insight into possibly incorrect assumptions about how matter behaves in neutron stars if these relationships prove to be less coherent than previously thought.

I. PROGRESS MADE

A. Analysis

After many of the efforts detailed in my previous report, we are now in a great place to start testing the validity of different universal relations. Since this project is largely a continuation and improvement on previous research done with parametric EoS, our analysis will largely mirror that of Ref. [1].

The first thing I needed to get working was a fit that could be applied on upwards of one-thousand different EoS models at once. This was done using a linear least-squares operation within the `scipy` package as detailed in Ref. [2] for a majority of our relations. However, some of thee relationships use nonlinear fitting, and as such are much more a source of trouble. This operation was also able to provide us with the much needed residuals of the individual EoS from the fit. Fig. 1 shows some example plots of what we've been able to make so far.

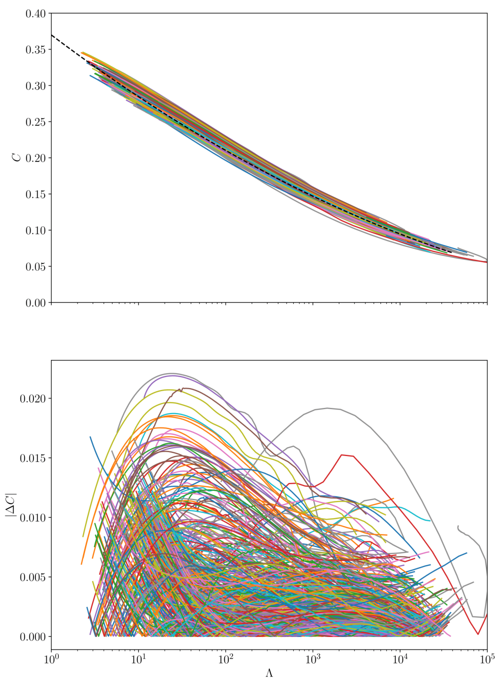
Of course, these residuals mean nothing unless we come up with some sort of metric to measure them to, so in this we have decided to compare them to the uncertainty in measuring parameters from LIGO data. This way we can accurately measure if the uncertainty of our universal relations are adequately smaller than than the uncertainty in our experimental measurements. As such, we hope to gain insight into whether or not these relations provide any additional constraints to our collected data.

In order to find this info, I have used the integrated LSC package `Bilby` in order to infer our uncertainty in parameter estimation as detailed in Ref. [3]. This involves injecting a mock signal into simulated LIGO/Virgo/KAGRA data to replicate the process of determining parameters from a physical gravitational-wave signal. The process uses the LIGO cluster and takes many days to run, and so at the time of writing this I have many runs currently ongoing to try and constrain our measurement uncertainties at many different points in parameter space.

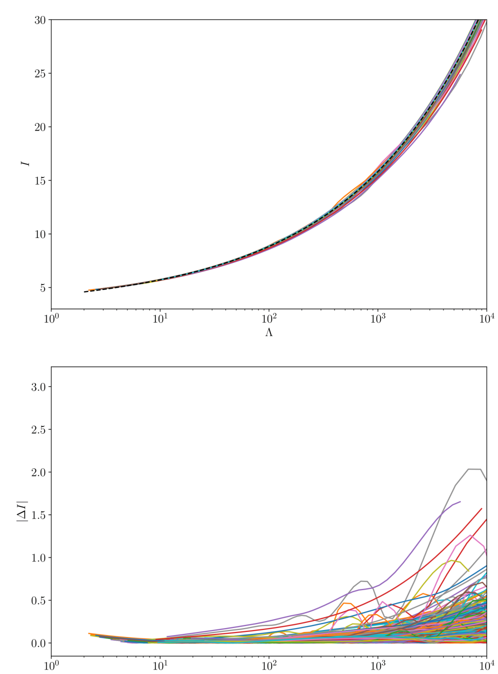
B. Optimizations

While analysis is our first and foremost goal, I have also made progress in the general functionality of my code. The first thing I would like to discuss is the overall generalization and expansion of my code for greater modularity and future use. After ensuring the basic functions of my code, such as data retrieval and cleaning, I moved on to allowing results to be repeatable by seed implementation. While the random sampling of models allowed the models we analyzed to be a very good generalization of the population of models as whole, to ensure that we both could conduct much more in-depth analysis of certain sample groups and allow future readers to verify our data, seed implementation and continuous documentation of these seeds and their results became paramount. Along with these changes, I also added functionality to skip random sampling all together if the user provides a list of EoS they would like to use for data gathering.

After this, my research mentor had noticed that three-dimensional array of data returned by our code can be very hard to share and discuss with someone less familiar with how the code is set up. Due to this, I also added implementation to allow users to return all the data as a dictionary rather than an array with



(a) Compactness (C) vs Deformability (Λ)



(b) Moment Of inertia (I) vs Deformability (Λ)

Figure 1: 2 universal relation candidates plotted with their respective fits and residuals

added labels for the data. Unfortunately, this makes the data much more difficult to handle when it comes to analysis, so the functionality of the arrays are still invaluable, but this extra customization allows users to get more out of our tools.

However, the single biggest change I've made to generalize my code is allowing the user to provide what criteria they would like all sampled EoS to meet. While my code beforehand had the ability to provide a number for the minimum weight they would like their models to be, this has been condensed into the user providing a list of different functions detailing any and all statements they would like to be met by any model they analyze. This is a huge improvement as it future-proofs any possible decisions we may want to make about what specific EoS we want data from with essentially any possible calculation we can do with the data provided.

II. PROBLEMS ENCOUNTERED

Overall I think the biggest problem I have encountered and overcame was making all the improvements to my code for data retrieval and cleaning. With more and more new features, it became increasingly difficult to stay on top of both all the things I wanted my code to do, but also how they work together.

I also encountered many problems with getting my least-squares fits to work. Due to the fact that I was fitting multiple plots, I had to construct a way to be able to concatenate all data points together and properly make a matrix out of them in order to solve the linear system. This took some trial and error, but luckily the linear nature of the program made it much easier to learn.

This was soon overshadowed by the nonlinear fits needed for testing a universal relation of $\Lambda_s = (\Lambda_1 + \Lambda_2)/2$ and $\Lambda_a = (\Lambda_1 - \Lambda_2)/2$ in which each Λ is decided from a chosen mass ratio q. The main problem encountered with this specific plot was the high variability in the amount of data points that could be gathered for each EoS. Depending on the mass ratio and volatility of the EoS used, it was possible for models to have anywhere from 100 to 3 data points. While this was overcame with careful interpolation, there was then the problem of learning to use a nonlinear least-squares function for fitting and analysis. The fit for this relation followed 1, a function with many different parameters that was tough to get a handle on, but this as well was something I got more used to handling.

$$\Lambda_a = \frac{1-q^{10/(3-n)}}{1+q^{10/(3-n)}} \frac{1+\sum_{i=1}^{3}\sum_{j=1}^{2} b_{ij} q^j \Lambda_s^{-i/5}}{1+\sum_{i=1}^{3}\sum_{j=1}^{2} c_{ij} q^j \Lambda_s^{-i/5}} \Lambda_s^{\alpha} \qquad (1)$$

As alluded to in the previous section, properly analyzing the data and coming up with the proper metrics to carry out said analysis has proved tough. The σ values needed to properly contextualize our data take weeks to create through Bilby runs, and this is only for one relation. Getting proper estimates for our measurement variance eats up a lot of time and might be a limiting factor into the future as well.

III. REMAINING GOALS

While we still have a ways in terms of getting all of our data in forms we'd like for analysis, we still have to be able to provide appropriate insight on the goodness of our universal relations. This is where the bulk of our remaining goals lie.

The first goal I'd like to get working before getting into the main portion of the analysis is making sure I can implement nonlinear fits into my code. This both allows me to create fits for plots that may not be linear and improve the fitting on plots that may have worked with linear fitting. of course, this is only a precursor to the main goal we still are yet to achieve.

After that, we still have much to do in ensuring the analysis of our data is a holistic view of the validity of our universal relations. Our current formulation follows the equations 2 and 3, where we hope to come up with an overall χ^2 quantifying the goodness of our relations that depends on our parameters α_i at each data point x_j.

$$\chi^2_{EoS}(\alpha_i) = \sum_{j=1}^{n} \frac{(Fit(\alpha_i, x_j) - Model(x_j))^2}{\sigma_j^2} \tag{2}$$

$$\chi^2(\alpha_i) = \sum_{k=1}^{N} \chi^2_{EoS,k}(\alpha_i) \tag{3}$$

Currently, the only missing part of this formulation is σ. We would like this value to not only be correlated to how much variation there are in measurements of some values, but we'd also like sigma to encode the physical likelihood of any model. Ideally, the less physically likely a model is, the less it would contribute to our overall χ^2 value. Both of these aspects are an important part in properly interpreting our data and are the last large parts of our analysis.

Outside of this, due to the sheer amount of different models being used, I'll also want to clean up the readability of many of my graphs for publication, as well as organize my functions and graphs into an organized and easy to retrieve repository.

[1] C.-J. H. K. Y. Zack Carson, Keterina Chatziioannou and N. Yunes, Equation-of-state insensitive relations after gw170817 10.1103/PhysRevD.99.083016 (2022).

[2] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods **17**, 261 (2020).

[3] G. Ashton, M. Hübner, P. D. Lasky, C. Talbot, K. Ackley, S. Biscoveanu, Q. Chu, A. Divakarla, P. J. Easter, B. Goncharov, F. Hernandez Vivanco, J. Harms, M. E. Lower, G. D. Meadors, D. Melchor, E. Payne, M. D. Pitkin, J. Powell, N. Sarin, R. J. E. Smith, and E. Thrane, BILBY: A User-friendly Bayesian Inference Library for Gravitational-wave Astronomy, **241**, 27 (2019), arXiv:1811.02042 [astro-ph.IM].