Angular Momentum of Rapidly Rotating Neutron Stars

Authored by Luke Skon

Advised by Professor Demian Cho, St Mary's University of Minnesota and Dr. James Skon of Mount Vernon Nazarene University

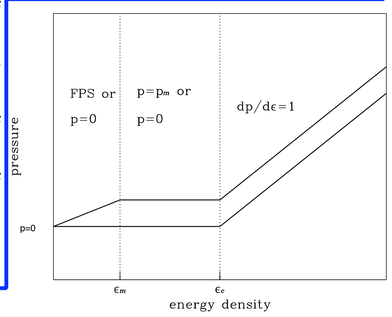
Abstract:

It is difficult to observe the behavior of nuclear material in extreme circumstances. The constraints of neutron stars is a good example. Neutron stars are gravitationally bounded, extremely dense and relatively cold. However, the density is too high and temperature too low to create in laboratories, so we must model them computationally to the best of our ability. Physicists can use the Einstein equations to model the behavior of a rotating neutron star. Given a specific equation of state, the Einstein equations reveal a maximum angular momentum. However, determining the maximum angular momentum is difficult. Because of the high degree of recursive computation involved, this study requires a computer program. The program was originally programmed in C in 1995. Advances in computer technology will allow the computation of one configuration to be done in two seconds where it would have taken two minutes. The program was updated to C++ and an SQL database was constructed to store all of the data aggregated by the program. These updates improve the accuracy and increase the efficiency of parameter space searching. Slight modifications could allow the program to be used in future projects involving rapidly rotating neutron stars.

Introduction:

A star is a fluid ball in hydrostatic equilibrium, a balanced state between its own gravity and pressure. Pressure of fluid is a function of various physical quantities, called an equation of state (commonly abbreviated as EoS). The fluid of neutron stars, which is a mixture of neutrons, protons, electrons and small amount of exotic hyperons, is relatively cold. Thus, the pressure is approximately just a function of energy density, ρ. The project is to find a maximum angular momentum that neutron star can possibly have. Naively, angular momentum depends on matter distribution and spin of neutron star. We need to consider each EoS that satisfies causality, and ensure the speed of sound doesn't exceed speed of light. There must also not be too great of a mass at the core, or it will collapse into a black hole.

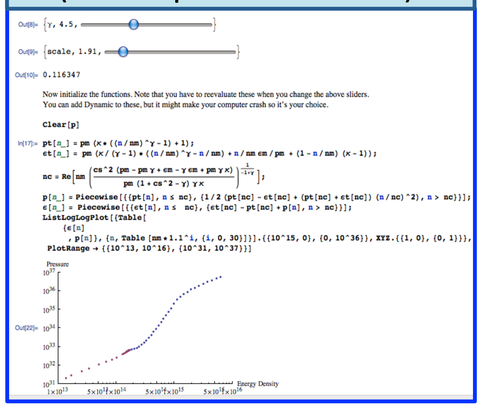
As an example, EoS below is the EoS used to find a neutron star with maximum mass. EoS at core is the stiffest possible allowed by causality to prevent star from collapsing to black hole, while at outer part is the softest possible matter to allow it to pack up as much matter as possible.



An approximate similarity:

(3) Pictured above is a very fast rotating star known as VFTS 102, and while this is not a neutron star, it nicely demonstrates the oblong compression expected in a rapidly rotating neutron star. We can see that the star matter acts like a fluid at this speed, held together by gravity. (Neutron stars that spin very fast have also been discovered, such as XTE J1739-285 (4), the fastest rotating neutron star known as of this writing)

Mathematica code (Of the parametric EoS):



Switching to C++ and Program Discrepancy:

A major goal of this research was to improve the efficiency and usability of the existing C code base use to compute the models of rapidly rotating neutron starts. Both C++ and Java were evaluated, and C++ proved to be significantly more efficient. The original code base used complex and error prone pointer structures for representing multi-dimensional models. These structures were converted to C++ multidimensional arrays. To our surprise, Both the C++ and Java implementations, though identical in function, produced slightly different, though identical, results from the C version. Significant investigation and experimentation did not lead us to an explanation for the differences, nor a determination as to which solutions were best. Below are some comparisons of the values of Java and C. Please note that the Java values are way more precise than presented:

	Ratio	Energy density	Mass	Baryonic Mass	Radius	Spin	Omega_K	Moment of Inertia	Angular Momentum
C first row	1.000	1.0	0.810	0.863	9.945	0.0	0451.0	0.000	0.000
C++ first	1	1	0.810071	0.862595	9.94533	0	10451	0	0
Java first	1.000000	1.00000	0.8100714	0.8625951	9.945328	0.0	10451.04	0.0	0.0
C last	0.575	1.0	1.007	1.077	14.091	6980.8	6980.8	0.86	0.672
C++ last	0.616685	1	0.923248	0.987357	14.0637	6612.22	6612.72	0.572157	0.505025
Java last	0.616685	1.0000	0.9232477	0.98736	14.06374	6612.222	6612.724	0.572157	0.505025

More research into this topic is needed, and is another interesting topic for further study.

Parametric Automation:

We changed all equation of state passing and retrieval to arrays instead of file reading and writing. Turning the data-gathering from file storage to arrays allowed me to create an entirely new program that generates parametric equations of state, but after lots of experimentation of various equations of state and modifications of equations of state, we settled on states that were based on Koranda, Freedman and Stergioulas's (5) parametric equation of state. This equation of state is specifically crafted to minimize the early equation to the lowest known state and to maximize to the speed of light constraint. However it is discontinuous and there is a unitary error, so it is not entirely valid for my purposes. This equation is the lowest EoS, the fps until a specific number density. That number density is labeled nm, the pressure at that number density is pm, the energy density is εm, the parameters are gamma and kappa. There is a critical n where the speed of light will be surpassed if it goes any higher, so we accounted for that in my calculations:

$$\kappa > 0; \gamma > 1; \gamma\kappa < 1 + \frac{\varepsilon m}{pm}; nc = \text{Real}[nm\ (\frac{c^2(pm - pm\gamma + \kappa m - \gamma\varepsilon m + pm\ \gamma\ \kappa))}{(pm\ (1 + c\text{^}2 - \gamma)\ \gamma\ \kappa}\ )^{1/(-1+\gamma)}];$$

$$p(n) = pm\ (\kappa((\frac{n}{nm})^{\gamma} - 1) + 1);\ \{nFPS<n \le nc\}; p(n)=\frac{1}{2}(p(nc) - \varepsilon(nc) + (p(nc) + \varepsilon(nc))\ (\frac{n}{nc})^2);\ \{n> nc\}$$

$$\varepsilon(n)=pm\ (\kappa/(\gamma - 1)*((n/nm)^{\gamma} - n/nm) + 10^{-21}\ \frac{n}{nm}\ \frac{\varepsilon m}{pm} + (1 - n/nm)\ (\kappa - 1));\ \{nFPS<n \le nc\}$$

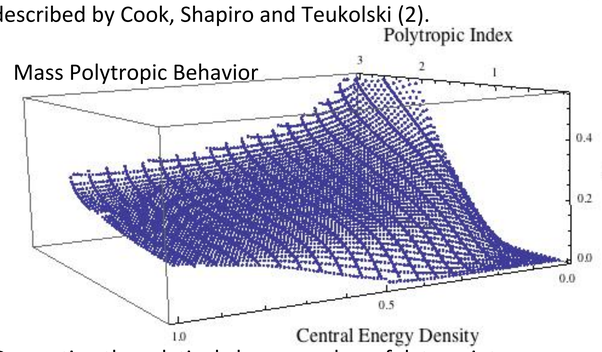
$\varepsilon(n)=\varepsilon(nc) - p(nc) + p(n); \{n > nc\}$. : These equations were algorithmically computed in C++. An equation of state has to be complete before it is put into RNS, so a number of spaced apart points are calculated before the program is run. In order to satisfy the constraints that λ>1, κ>0 and λκ<1+εm/pm, we had the loop calculate a series of gammas, and calculated kappa as κ=(1+εm/pm)/(λ y) where y was an arbitrary scale factor of at least 1. This arrangement results in a fairly good search throughout the parameter space. The convenient part is, it takes one line of code to change where it is searching, where before the equation of state had to be written.

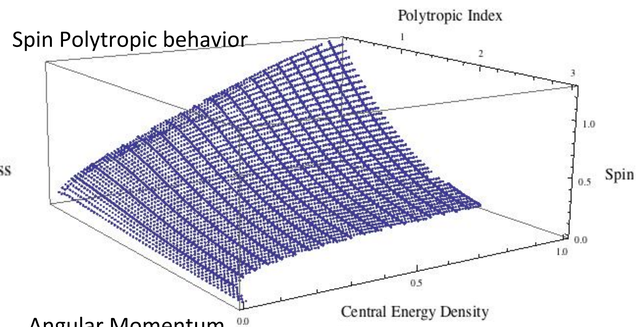
Original RNS.c code:

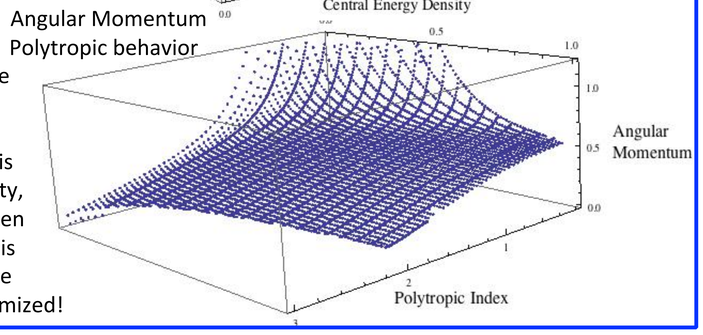
The code used to model the rotating neutron star was originally made by Nikolas Stergioulas as a part of his PhD thesis. The program has not been modified since 1997. Many programming techniques used by Stergioulas at the time are either obsolete due to the current speed of computation, use less than optimal programming habits or outright confusing to an outside editor. However, the code was clear enough and comprehensive enough to modify it without a complete rewrite. The first programming modification we made was to change the output method from a print to an array, and to output it to another source rather than as text output. From here we modified the code to accept the input parameters as a range of values, and to output the critical values to a database. The inputs were looped parameter searches of the polytropic index and the central energy density. From here, we began my first analysis as a polytropic analysis.

Polytropic Analysis:

We used a polytropic equation of state as our first measure of gauging the behavior of our input. The equation of these polytropic equations of state are: (1) $P = K\rho^{1+1/N}$ where P is the pressure, K is a constant ρ is the density and N is a parameter called the polytropic index. The units of this polytropic equation of state has been programmed in such a way so as to create a dimensionless set of quantities as described by Cook, Shapiro and Teukolski (2).







Computing the relatively large number of data points necessary to generate these graphs was prohibitively time consuming prior to the creation of a fast automated version for this project. We wish to maximize mass, spin and angular momentum. We can see that, at a higher central energy density and higher polytropic index, the mass is high. At a smaller polytropic index and a higher central energy density, we get much higher spin, but angular momentum only goes high when the central energy density and polytropic index is low. What does this mean? It confirms that if the inside of a neutron star has a non-dense core and a high density outside, then its angular momentum is maximized!

Results and future exploration:

As a result of this project, the code has been rewritten to C++ and Java in multiple formats. The SQL Database is used to store all results and eliminate re-computation for identical parameters. SQL allows later convinient offline exploration of the results. An SQL search makes searching for maximums and minimums or specific ranges of values vastly simpler. Our Mathematica program makes a great visualization tool for further study. Finally, we have discovered a discrepancy between the C code and our new C++ code, which means that old values that may have been incorrect can now be corrected.

References:

(1) Sharon Morsink (2012*) User Manual for RNS: version 2.*

(2) Cook, Shapiro and Teukolski, APJ 422, 227 (1994) *Rapidly Rotating Neturon Stars In General Relativity: Realistic Equations of state*

(3) *NASA, ESA, and G. Bacon (STScI) (2012), Image of the day: VFTS*
http://www.nasa.gov/multimedia/imagegallery/image_feature_2141.html

(4) P. Kaaret, Z. Prieskorn, et al. (2006) *Evidence for 1122 Hz X-Ray Burst Oscillations from the Neutron-Star X-Ray Transient XTE J1739-285*, http://arxiv.org/abs/astro-ph/0611716

(5) Stergioulas, Koranda and Freedman, APJ 488, 799, (1997) *Upper Limits Set By Causaltiy on Rotation and Mass of Uniformly Rotating relativistic stars.*

Acknowledgements:

I would like to thank a great deal of people for their assistance on this project. First of all, I would like to thank Professor Demian Cho of St. Mary's University of Minnesota, who informed me about and advised me on this project, even though he had a very busy moving schedule this summer. Secondly, I would like to thank Dr. James Skon of Mount Vernon Nazarene University for advising and assisting me on the computational upgrade and expansion on this project. Thirdly, I would like to thank Nikolas Stergioulas, Scott Koranda and John Freedman for providing the project and the program that I have modified. I would like to thank Kenyon College Summer Science Program for the funding. Finally, I would like to thank all of the faculty of the Kenyon physics department, especially Dr. Timothy Sullivan, who assisted me on this project during the middle of the summer when Professor Cho was unavailable.