NSSS
(Neutron Star Spin Sequences)

September 26, 2019

1 What can it do?

The NSSS is based on the RNS code by Stergioulas, & Friedman [2]. For an introduction to the parameters and terminology see Cook, Shapiro & Teukolsky [1]. This code is able to compute

- One neutron star (NS) with a certain value of oblateness;
- Sequences of neutron stars (NSs) with constant rest mass (mass of the particles that make up the star, denoted by M_0) by increasing the rotational frequency from zero to the limiting spin frequency, ν_K (Kepler limit); and
- Sequences of neutron stars, each with constant M_0, by increasing the rotational frequency from zero to a maximum frequency, $\nu_{\max}$, given by the user.

2 Compilation

Before compiling the code, check that the definition of the compiler in the "Makefile" is correct for your computer. In this same file, we can modify the size of the grid, which is, by default, MDIV$\times$SDIV=151$\times$301.

To compile, at the command line, type:

```
>make
```

Note that after modifying the grid size in the "makefile", you need to delete the object files (with '.o' extension).

Another important thing to have in mind before running NSSS, is that the EOS files have to be in the **same directory** as the rest of the files for NSSS. Otherwise there will be a "Segmentation fault" error message.

3 Command Line Flags

The parameters that the code accepts to compute either of the previous tasks are:
-f name of the file with the tabulated equation of state (EOS)
-e minimum central energy density (in g/cm^3)
-n number of sequences produced
-m central energy density for the maximum mass neutron star for a given equation of state
-s spin frequency (in Hz)
-r ratio of r_p/r_e (between 0 and 1)
-t maximum spin frequency

4 Equations of State

Equations of state[1] used in NSSS are in tabular format. The example EOS are shown in Table 1, where we can see, for ten EOS, the range in the central energy density to compute the different sequences of NSs.

[1]They can be obtained from: `https://github.com/rns-alberta/eos`

Table 1: Data corresponding to the value of R, r_p/r_e, and ε_c when $M \sim 1\,\mathrm{M}_\odot$ for each EOS considered here (this is one combination). It also shows the value that ε_c has to have when a nonrotating NS with the maximum mass, M_M, is computed; the corresponding values of radius and rest mass are R_M, and $M_{0,M}$, respectively.

	R and ε_c when $M \sim 1\,\mathrm{M}_\odot$				ε_c when M_M is obtained			
EOS	M ($\mathrm{M}_\odot$)	R (km)	r_p/r_e	ε_c (g/cm^3)$\times10^{15}$	M_M ($\mathrm{M}_\odot$)	$M_{0,M}$ ($\mathrm{M}_\odot$)	R_M (km)	$\varepsilon_{c,M}$ (g/cm^3)$\times10^{15}$
ABPR1	1.009	11.48288	0.996	0.739545	1.93468	2.26467	10.8059	2.3317
ABPR2	1.004	10.49412	0.994	1.09338	1.49662	1.69859	9.28537	3.3454
ABPR3	1.004	11.57512	0.984	0.983153	1.47598	1.65273	9.67557	3.1508
APR	1.000	11.38782	0.998	0.775	2.23685	2.71921	9.89745	2.8437
BBB1	1.000	11.21119	0.988	0.836243	1.78951	2.08233	9.6527	3.07827
BBB2	1.000	12.09300	0.872	0.777099	1.91853	2.26798	9.48855	3.7733
HLPS1	1.003	9.68367	0.996	1.04467	2.0423	2.51353	9.23051	3.1781
HLPS2	1.000	14.48549	0.706	0.5598	2.49576	3.05768	11.5378	1.9929
HLPS3	1.005	13.21363	0.998	0.449864	2.98219	3.7009	13.3676	1.455
L	1.000	15.18746	0.914	0.352224	2.71085	3.22955	13.7477	1.44216

5 Computing one neutron star

To compute a single neutron star we need a tabulated equation of state (EOS), a value of the central energy density (ε_c), and the value of the quotient that tells us how oblate the neutron star is (Figure 1). This quotient is indicative of how fast this object is spinning. The faster it rotates, the more oblate it becomes. When we have a ratio of 1 the neutron star is spherical. The EOS and the central energy density will be used to create a zero-spin star with a certain mass and radius. Then, the quotient r_p/r_e, will be used to make the star oblate, and therefore, spinning. So, an example of the command line would be

```
./nsss -f eos-master/eosL -e 0.36e15 -r 0.9
```

This command line uses the tabulated equation of state L located in the folder "eos-master", the NS will have a central energy density of $\varepsilon_c = 0.36 \times 10^{15}$ g/cm^3 and it will have a ratio of the polar and the equatorial radii of $r_p/r_e = 0.9$, which tells us that the neutron star is rotating. The output is the following set of data that belongs to this one star, which like the previous case, it is denoted by a sentence before the headings:

```
Computing one neutron star
e_c    Mass      Mass_0    Radius     R-ratio   Spin      K freq
e15    Msun      Msun      km         –         Hz        Hz
0.36   1.05561   1.10903   15.34503   0.900     384.982   995.61897
```

In the headings, "e_c" is ε_c, "Mass_0" is M_0, "R-ratio" is r_p/r_e, "Spin" is the spin frequency, and "K freq" is the Kepler frequency. This output only appears in the terminal, it is not saved to a file.

6 Computing a sequence from zero spin to the Kepler limit

The sequences that are computed in this part are made of neutron stars with a constant value of rest mass (also called baryonic mass), in other words, no matter is being added to it.

To compute a sequence of neutron stars we need to have a value of central energy density, ε_c, a tabulated equation of state, and the value of ε_c at which we get the maximum-mass neutron star.

First, to create a sequence of neutron stars a non-rotating star is created using the previous description and this value of M_0 is stored. Now, the star need to rotate gradually faster until it reaches the break

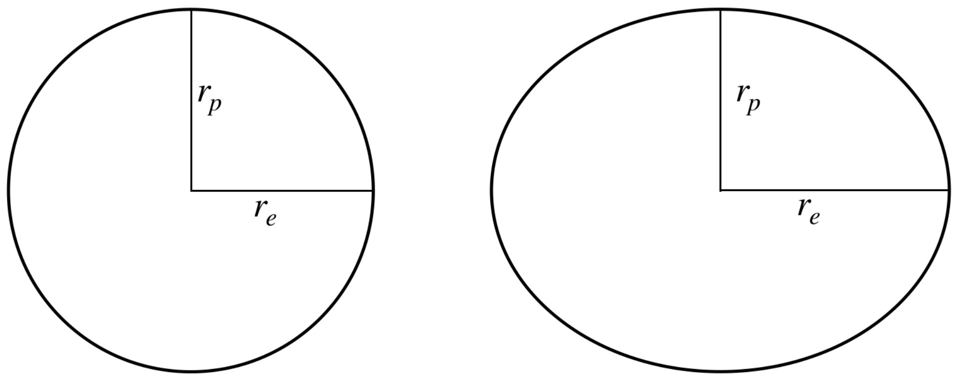

Figure 1: Left: Model of a non rotating neutron star with the two radii considered in NSSS, the equatorial radius, r_e, and the polar radius, r_p. In this case, they are equal to each other. Right: Rapidly rotating neutron star, where now $r_e > r_p$, because of the rotation.

up limit (Kepler limit). To do this, the value of r_p/r_e is decreased and to find a neutron star with the same rest mass as the one stored the code makes use of the 3-point interpolation, and the following star takes the value of ε_c to continue and create another star This process of computing one neutron star continues as r_p/r_e is decreased (making the neutron star rotate faster) until the star gets to the Kepler limit, which is the maximum rotational frequency that the star can rotate at. Now the code has produce one sequence of neutron stars with the same value of M_0.

To compute another sequence the value of ε_c is increased and the code computes another sequence of stars in the same way.

This process can stop after a given number of sequences that we need (by input) or if it reaches the maximum mass for the equation of state being used.

An example of an input line to compute 5 sequences would be

./nsss -f eos-master/eosL -e 0.36e15 -n 5 -m 1.44216

This command will take the tabulated L equation of state, that is located in the folder "eos-master", it will also take a starting value of the central energy density, $\varepsilon_c = 0.36 \times 10^{15}$ g/cm^3; the "-n" parameter tells NSSS to compute 5 sequences of NSs, each with a constant M_0. The next command, "-m 1.44216" tells NSSS to compute the nonrotating neutron star with the highest mass for the L equation, which happens at $\varepsilon_{c,M} = 1.44216 \times 10^{15}$ g/cm^3 (There is no need to type "e15" in the command line for this parameter, just "1.44216"). Therefore, this line will output the following

Computing sequences with spin frequency from 0 Hz to the Kepler limit

e_c	Mass	Mass_0	StatM	Radius	R-ratio	StatR	Spin	K freq
e15	Msun	Msun	Msun	km	–	km	Hz	Hz
Energy center = 0.36								
0.36	0.99737	1.04661	0.99737	14.51368	1.000	14.51368	0.000	1047.09802
0.35978	0.99738	1.04661	0.99737	14.52606	0.998	14.51368	52.552	1044.55302
0.359603	0.99743	1.04662	0.99737	14.54023	0.996	14.51368	76.061	1042.68951
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

The headings are the same as the previous case for one NS, the additional ones are just "StatM", which is the mass of the first nonrotating NS in each sequence, and it has a radius given by "StatR". Even if not all of the physical properties of the neutron star are displayed (see Table 2), all of them will be saved to an output file.

7 Computing sequences up to a given value of spin frequency

This task is very similar to the previous one, to compute a series of sequences it takes a tabulated EOS, a value of ε_c and the value of ε_c at which the maximum-mass neutron star is obtained. The difference is that, in this case, the maximum rotational frequency is not the Kepler limit, it can be set by the user, no matter which EOS is being used. An example of the line of initialization for this task is

./nsss -f eos-master/eosL -e 0.36e15 -n 5 -m 1.44216 -t 800

The last parameter, "-t 800", tells the code that the sequences will finish when the NSs reach a spin frequency of $\nu_{\max} = 800$ Hz. The output data will be the same as for the case reaching the Kepler limit, but now the statement above the headings will be "Computing star with spin frequency from 0 to 800 Hz". The output data for this case will also be saved to a file.

8 Properties obtained from the code

The physical properties of the neutron star that NSSS outputs in columns are seen, in order, in Table 2. Note that these are some of the parameters used in [1]

Table 2: Physical properties obtained from NSSS for every single NS computed.

Parameter	Description
ε_c	Central energy density
M	Total mass (in $M_\odot$)
M_0	Rest mass, also known as baryonic mass (in $M_\odot$)
M_*	Mass of the first nonrotating NS in a sequence (in $M_\odot$)
M_M	Maximum mass of a nonrotating NS for a given EOS (in $M_\odot$)
R	Equatorial radius of the NS (in km)
r_p/r_e	Ratio of the polar radius and the equatorial radius
R_*	Equatorial radius of the first nonrotating NS in a sequence (in km)
ν	Rotational frequency (in Hz)
ν_K	Kepler limit for rotation (in Hz)
J	Angular momentum (in $\mathrm{cm^2 g/s}$)
T	Rotational kinetic energy (in g)
U	Gravitational binding energy (in g)
R_M	Radius of the maximum-mass NS for a given EOS (in km)
M_M/R_M	Compactness of the nonrotating maximum-mass NS for a given EOS

References

[1] Cook, G. B., Shapiro, S. L., & Teukolsky, S. A. 1994, ApJ, 424, 823

[2] Stergioulas, N., & Friedman, J. L. 1995, ApJ, 444, 306