Users Manual for RNS

Nikolaos Stergioulas

June 15, 2022

1 Introduction

RNS is a code that constructs models of *rapidly rotating, relativistic, compact stars* using tabulated equations of state which are supplied by the user. The code is written after the KEH method (Komatsu, Eriguchi & Hachisu, 1989) and modifications introduced by Cook, Shapiro & Teukolsky (1994). It can compute individual models as well as sequences of fixed mass, rest mass, angular velocity or angular momentum models. All models assume uniform rotation. You can read more about this code in Stergioulas & Friedman (1994) and references therein.

2 Compiling the code

RNS is written in the ANSI C programming language. Machine specific commands were avoided, so that the C program should compile on any platform using an ANSI C compiler. The source code comes with a Unix `Makefile`. Before running the `Makefile` for the first time, one should check to see if it uses the correct compiler (the default is `cc`) and the correct flags for choosing the ANSI C option (the default is `-std1`), and optimization (default is `-O`) - if not, the `Makefile` should be edited. Once the `Makefile` is correct, run `make` to obtain the executable `rns`. If significant changes to the `Makefile` are needed in order to compile it on your platform, please send the changes to the author, so that they can be included in future versions.

3 Supplying the equation of state

RNS can compute stars with either a polytropic or tabulated equation of state (EOS). The format and units for each type of EOS is described in this section.

3.1 Tabulated Equation of State

RNS needs a tabulated, zero-temperature equation of state (EOS), as an input, in order to run. You will have to format the EOS file in a specific way, as described here: The first line in the EOS file should contain the number of tabulated points. The remaining lines should consist of four columns - energy density (in gr/cm^3), pressure (in dynes/cm^2), enthalpy (in cm^2/s^2), and baryon number density (in cm^{-3}). The *enthalpy* is defined as

$$H(P) = \int_0^P \frac{c^2 dP}{(\epsilon + P)}, \tag{1}$$

where ϵ is the energy density, P is the pressure and c is the speed of light.

The number of points should be limited to 200. Example files (e.g. `eosC`) are supplied with the source code. The EOS file needs to be specified using the following flag:

`rns -f` *filename*

A program "HnG.c" is included which will convert an EOS file tabulated in the form pressure v.s. energy density to the form required by RNS. See the program listing for more details.

3.2 Polytropic Equations of State

The default type of equation of state for RNS is tabulated, but RNS will also compute polytropic stars. The polytropic equation of state is

$$p = K\rho_0^{1+1/n} \tag{2}$$

where p is the pressure, ρ_0 is the baryon mass density, K is the polytropic constant, and n is the polytropic index. To specify a polytrope with index n, the following command line options need to be specified:

`rns -q poly -N` *index*

The program uses the dimensionless units described by Cook, Shapiro & Teukolsky 1994. All quantities are reported in dimensionless units if a polytropic star is indicated.

4 Specifying model parameters and other options

RNS has several options which allow you to specify model parameters and choose from different output formats. A model is defined uniquely by specifying two parameters - one will always be the central energy density and the other can be one of the following: mass, rest mass, angular velocity, angular momentum, or the ratio of the polar coordinate radius to the coordinate equatorial radius (*axes ratio*, see definition of coordinates later in the text) .

The parameters are specified using the following flags:

`-e` *central energy density* in $\mathrm{gr/cm^3}$

`-r` *axes ratio*

`-m` *mass* in $\mathrm{M_\odot}$

`-z` *rest mass* in $\mathrm{M_\odot}$

`-o` *angular velocity* in $10^4\mathrm{s}^{-1}$

`-j` *angular momentum* in $\mathrm{GM_\odot^2/c}$

Note that if a polytropic star is requested, dimensionless units should be used. Consult the tables given in Cook et. al 1994 to find suitable values for these parameters.

The code is written so as to directly construct a model when the axes ratio is specified (which is therefore the fastest option). If one specifies a different parameter e.g. mass, the code constructs several (usually more than ten) models by varying the axes ratio, until it finds a model, for which the chosen parameter is within some allowed tolerance of the specified value. The default tolerance is 10^{-4} (relative error) and can be changed by using the flag

`-b` *tolerance*

A smaller tolerance means more models will be constructed.

A model is constructed by iteratively solving the field equations and the hydrostatic equilibrium equation, until the coordinate equatorial radius changes by less than a specified relative accuracy. The default accuracy is 10^{-5} and this can be changed by using the flag

`-a` *accuracy*

Convergence is monitored by printing out the relative difference in the coordinate equatorial radius from one iteration to the next. This print-out can be suppressed by using the flag

`-d` 0

In rare cases, such as for unstable models of very stiff equations of state, the iteration may not, at first, converge. Such cases can easily be fixed by using a relaxation factor in the iteration. A factor of 0.8 usually makes the iteration convergent (the default is 1.0, which amounts to no relaxation). It can be specified using

`-c` *relaxation factor*

Specifying the above parameters is not sufficient to get the program started. One also needs to select the task that is to be performed.

5 Tasks

RNS can perform eight different tasks, which we list below, along with the required options for each task:

`-t model` Computes a model with fixed central energy density ϵ_c and *axes ratio*. Requires `-e` and `-r`

`-t gmass` Computes a model with fixed ϵ_c and *gravitational mass*. Requires `-e` and `-m`

`-t rmass` Computes a model with fixed ϵ_c and *rest mass*. Requires `-e` and `-z`

`-t omega` Computes a model with fixed ϵ_c and *angular velocity*. Requires `-e` and `-o`

`-t jmoment` Computes a model with fixed ϵ_c and *angular momentum*. Requires `-e` and `-j`

`-t static` For a given ϵ_c computes the nonrotating model. Requires `-e`.

`-t kepler` For a given ϵ_c computes the model with Keplerian angular velocity. The default relative accuracy is 10^{-4} and can be changed with `-b` *tolerance*. Requires `-e`.

`-t test` Computes the test model.

To find out if the source code compiled correctly, you can run the test model

```
rns -f eosC -t test
```

and compare the output to the file `test.out`.

For specific examples of the above tasks and their output, see the file `examples.test`.

6 Sequences

You can obtain a sequence of models with the same two parameters fixed (instead of just one model) by specifying a range of central energy densities and the number of models desired. For example

```
-e 1e15 -l 3e15 -n 10
```

will give a sequence of ten models between the central energy densities of 10^{15} and 3×10^{15} gr/cm^3. The models will be equally spaced in $\log \epsilon_c$. You can do this with any of the tasks described in the previous section. However, you have to make sure that the models you requested actually exist. For example, if you want to compute a sequence of constant rest mass, you should first compute and examine the nonrotating and Keplerian sequences for a wide range of energy densities and then select a sub-range of ϵ_c for which stars of the given rest mass exist - otherwise, the code will fail to converge.

7 Output

RNS prints out 17 physical quantities upon succesfull computation of a model. These are:

ϵ_c *central energy density*

M *gravitational mass*

M_0 *rest mass*

R_e *radius at the equator (circumferencial, i.e.* $2\pi R_e$ *is the proper circumference)*

Ω *angular velocity*

Ω_p *angular velocity of a particle in circular orbit at the equator*

T/W *rotational/gravitational energy*

$cJ/GM_\odot^2$ *angular momentum*

I *moment of inertia (except for nonrotating model)*

Φ_2 *quadrupole moment (program needs to be compiled on HIGH resolution for this to be accurate)*

h_+ *height from surface of last stable co-rotating circular orbit in equatorial plane (circumferencial) - if none, then all such orbits are stable*

h_- *height from surface of last stable counter-rotating circular orbit in equatorial plane (circumferential) - if none, then all such orbits are stable*

Z_p *polar redshift*

Z_b *backward equatorial redshift*

Z_f *forward equatorial redshift*

ω_c/Ω *ratio of central value of potential* ω *to* Ω

r_e *coordinate equatorial radius*

r_p/r_e *axes ratio (polar to equatorial)*

The following values for the physical constants are used: $c = 2.9979 \times 10^{10}\mathrm{cm/s^{-1}}$, $G = 6.6732 \times 10^{-8}\mathrm{g^{-1}cm^3s^2}$, $m_B = 1.66 \times 10^{-24}\mathrm{gr}$, and $M_\odot = 1.987 \times 10^{33}\mathrm{gr}$. The coordinates of the stationary, axisymetric spacetime used to model the compact star are defined through the metric

$$ds^2 = -e^{\gamma+\rho}\, dt^2 + e^{2\alpha}(dr^2 + r^2 d\theta^2) + e^{\gamma-\rho} r^2 \sin^2\theta (d\phi - \omega dt)^2. \tag{3}$$

where the potentials γ, ρ, α and ω are functions of r and θ only. The matter inside the neutron star is approximated by a perfect fluid.

7.1 Printing Formats

There are two different printing formats:

`-p 1` is a detailed vertical list of the above quantities (default)

`-p 2` is a compact horizontal print-out of the same quantities, which is useful when computing sequences.

In addition to the above quantities, one can print out the values of the four metric potentials and the pressure at every grid point with

```
-p 3
```

This is a very long print out and should be directed to a file rather than to the screen. The format is:

$s = r/(r + r_e)$ $\quad$ $\cos\theta$ $\quad$ ρ $\quad$ γ $\quad$ α $\quad$ $\omega(10^4\mathrm{s^{-1}})$ $\quad$ P $(dynes/cm^2)$

8 Numerical Grid

RNS uses a uniform 2-D grid in the variables $s = r/(r + r_e)$ and $\mu = \cos\theta$. The center is at $s = 0$, the equator at $s = 0.5$ and infinity at $s = 1$. The equatorial plane is at $\theta = \pi/2$ while the pole is at $\theta = 0$. If one uses $SDIV$ points in the s-direction and $MDIV$ points in the μ-direction, then the grid points are obtained by the formulas

$$s[i] = 0.9999\Big(\frac{1-i}{SDIV-1}\Big), \tag{4}$$

$$\mu[j] = \Big(\frac{1-j}{MDIV-1}\Big). \tag{5}$$

The default grid size is $MDIV \times SDIV = 65 \times 129$. You can change this by editing the `Makefile`. If you specify too large a grid size and get a memory segmentation fault, you can fix this by issuing the command

```
limit stacksize unlimited
```

in a UNIX environment.

9 Availability

RNS is available as a public domain program and can be downloaded by anonymous ftp from `pauli.phys.uwm.edu` in the directory `/pub/rns`. You can download either `rns.`*version*`.tar.Z` or `rns.`*version*`.tar.gz`. The code and documentation are also available at the website:

```
www.gravity.phys.uwm.edu/Code/rns
```

Please send any suggestions, comments or questions to the author at the following address:

Nikolaos Stergioulas
E-mail: niksterg@pauli.phys.uwm.edu

10 Acknowledgments

I am gratefull to John L. Friedman for suggesting this project and for all his advice and support. Thanks also go to Scott Koranda for making the source code compile correctly on a SUN workstation. This project was supported by NSF grant PHY-9105935.

11 References

Komatsu H., Eriguchi Y. & Hachisu I., 1989, *MNRAS*, **237**, 355
Cook G.B., Shapiro S.L. & Teukolsky S.A., 1994, *APJ*, **422**, 227
Stergioulas N. & Friedman J.L., 1994, in press (*ApJ*, 1995, May 1)