PHYS3071/7073 - Due Friday, 11th April 2014 by 11am

Assignment 5 — ODEs, Part 2 – Neutron Stars

In this assignment you will explore the structure of relativistic neutron stars. Following worksheet 9/10 you will again solve an equation of hydrostatic equilibrium for a spherically symmetric mass distribution. This time, however, we will consider equations of state for quarks with relativistic Fermi pressures. This equation is known as the Tolman-Oppenheimer-Volkoff (TOV) equation and is given in geometrical units $G = c = 1$ by:

$$\frac{d\hat{P}(\hat{r})}{d\hat{r}} = -K \frac{\left(\hat{\epsilon}(\hat{r}) + \hat{P}(\hat{r})\right)\left(\hat{m}(\hat{r}) + 4\pi\hat{P}(\hat{r})\hat{r}^3\right)}{\hat{r}^2\left(1 - 2K\frac{\hat{m}(\hat{r})}{\hat{r}}\right)} \tag{1}$$

with $K = 1.322 \cdot 10^{-42}$. As before, the mass $\hat{m}$, contained in a sphere of radius $\hat{r}$ is connected to the energy density $\hat{\epsilon}$ through

$$\frac{d\hat{m}}{d\hat{r}} = 4\pi\hat{r}^2\hat{\epsilon}(\hat{r}) \ . \tag{2}$$

Here $\hat{m}$, $\hat{P}$ and $\hat{\epsilon}$ are in units of Mev and MeV/fm^3. We also assume that for a relativistic 'gas' of quarks $\hat{P}(\hat{r}) = \frac{1}{3}\left(\hat{\epsilon}(\hat{r}) - 4B\right)$, with the 'bag' constant $B = 57$ MeV/fm^3.

Program Design

Take your C program from worksheet 9/10 that solves the non-relativistic version. We want to extend the code such that after it has solved the non-relativistic case it proceeds to solve the relativistic version, e.g. eqs. 1 and 2 together with the EOS. Again start at $r = 10^{16}$ fm (10 m), and integrate outwards until $P(r) < 0$. Again, the total mass, M, of the mass distribution (stellar mass) is given by $M = m(R)$ where R denotes the radius of the star defined by $P(r = R) = 0$.

Program structure

- We want to set up a `for loop` over 80 different central densities ranging from $\hat{\epsilon}_0 = 4.2B$ to $20B$.
- Inside this loop the program should firstly integrate outwards and solve Newton's version. Secondly the program should solve the TOV version. The integrations should in both cases be done with the modified midpoint method.
- The output should be put into a file called `'massvsradius.dat'`. The file should contain 5 columns: $\hat{\epsilon}_0$, R (via Newton in km), and M (via Newton in solar masses), R (via TOV in km), and M (via TOV in solar masses).

Submission

The full set of files you will have when you are done should be:

- Your C program, which should be named `as05-studentnumber-surname.c`.
- Your `massvsradius.dat` file renamed as `as05-studentnumber-surname.dat`

- A plot of your two M vs. R curves in the same graph and in a file `as05-studentnumber-surname.ps`
- A textfile `as05answers.txt` discussing the following questions:
  Q1: Is there a maximum mass in each sequence? Speculate as to what you think might be causing this.
  Q2: What does the relation between M and R appear to be for small $\hat{\epsilon}_0$? Discuss.

To submit your assignment, create a new directory `as05` and upload the above files to your directory, eg. s1234567/as05/. Your C program, the massvsradius.dat file, your graph and the text file should have COURSENUMBER AS05 SURNAME STUDENTID (case doesn't matter) at the top. Your postscript figure should have the same information in its title, i.e. use `set title <COURSENUMBER> AS05 <SURNAME> <STUDENTID>`. Make sure all files are named correctly. **This information is important for us when marking the assignments and we will have to deduct marks if it is missing or a file has the wrong name.** Also please do not forget to add a sufficient number of comments to your C code so that we understand how the program is working.

PHYS3071/7073 - Due Friday, 11th April 2014 by 11am

Grading Sheet – Assignment 5: Neutron Stars

A: /40% **Function**: Does the program compile, run and produce the correct output?

B: /15% **Readability**: Is the program easy to read and comprehend? Is it well-commented? If the code is sufficiently complex, has it been broken up into manageable subroutines, each of which is well-documented?

C: /10% **Usability**: Is the output formatting easy to understand?

D: /15% **Efficiency**: Does the program run efficiently? Is the coding clunky or unnecessarily complicated?

E: /10% **Presentation (Plots)**: Do the plots clearly convey the results? Does each plot have an appropriate title? Are the axes and the plot items clearly labeled? Was the correct style (points or lines) used for each item?

F: /10% **Analysis**: Were correct answers given to the questions asked in the assignment, and was the process used to obtain them reasonable and clearly explained?

Total Points: $\boxed{\quad /100}$