Solutions Ph 236b – Week 2

Kevin Barkett and Mark Scheel

California Institute of Technology
Course: Ph 236

January 20, 2016

Contents

Problem 1

Part (a)

Since ρ_0 is the rest mass density, and the mass is completely dominated by neutrons, $\rho_0 = m_n n$ where m_n is the rest mass of a single neutron and n is the number density. Since the neutron gas is non-relativistic, each neutron's energy is dominated by its rest mass so then $\rho = \rho_0 \Rightarrow \rho = m_n n$. The problem gives that $P = K\rho_0^\Gamma$. Therefore, $P(n) = K(m_n n)^\Gamma$.

Part (b)

The equation of state and the Newtonian equations of stellar structure and are

$$\begin{aligned}\frac{dm}{dr} =& 4\pi r^2 \rho_0 \\ \frac{dP}{dr} =& -\frac{Gm}{r^2}\rho_0 \\ P =& K\rho_0^\Gamma \end{aligned} \tag{1.1}$$

where dimensionful constants have been included, $\Gamma = 5/3$ and $K = 5.3802 \times 10^9$cgs units. Define the following quantities for ease of computation:

$$\begin{aligned}\tilde{\rho_0} =& \frac{\rho_0}{10^{15}\mathrm{g/cm}^3} \\ \tilde{P} =& \frac{P}{5.3802 \times 10^{34}\mathrm{dyn/cm}^2} \\ \tilde{m} =& \frac{m}{M_\odot} \\ \tilde{r} =& \frac{r}{10\mathrm{km}}\end{aligned} \tag{1.2}$$

Using these definitions, the equations above become

$$\frac{d\tilde{m}}{d\tilde{r}} = 6.318\tilde{r}^2\tilde{\rho}_0 \tag{1.3}$$

$$\frac{d\tilde{P}}{d\tilde{r}} = -2.4666\frac{\tilde{m}}{\tilde{r}^2}\tilde{\rho}_0 \tag{1.4}$$

$$\tilde{\rho}_0 = \tilde{P}^{1/\Gamma} \tag{1.5}$$

In order to perform the integration, initial conditions must be determined. Choose an initial central density ρ_c. To avoid the issue of the $1/\tilde{r}^2$ divergence at $\tilde{r} = 0$, choose the conditions at $\tilde{r} = \Delta\tilde{r}$ to be

$$\begin{aligned}\tilde{m}(\Delta\tilde{r}) =& 2.106\Delta\tilde{r}^3\tilde{\rho}_c \\ \tilde{P}(\Delta\tilde{r}) =& \tilde{P}_c = (\tilde{\rho}_c)^{5/3}\end{aligned} \tag{1.6}$$

The last equation comes from the fact that $\tilde{m} \to \frac{4\pi}{3}\tilde{\rho}_c\tilde{r}^3$ as the radius approaches the origin and therefore $\frac{d\tilde{P}}{d\tilde{r}} \to 0$; thus the difference between $\tilde{P}(\Delta\tilde{r})$ and $\tilde{P}_c$ is second order in $\Delta\tilde{r}$ and can be neglected for the initial conditions. The integration is continued until $\tilde{P} = 0$.

See the example script at the end of this solution for an example of the integration. Note that the exact values will depend on the scheme used but all results should approximate the ones output by the attached script.

(i)

See Figs. 1–3 for the mass and radius profiles.

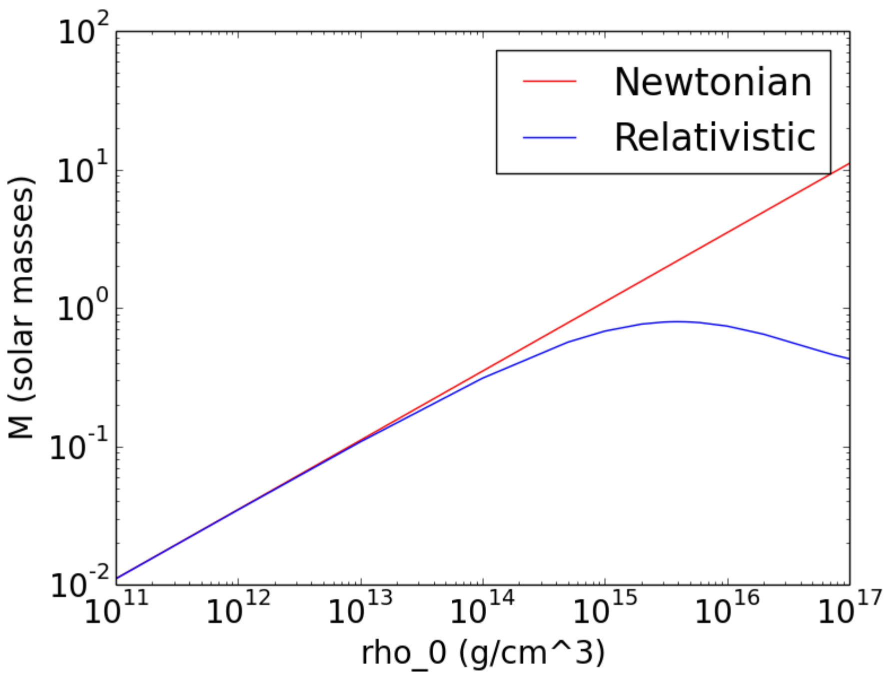

Figure 1: Mass vs central density.

(ii)

Looking at Figs. 1 and 2, both the Newtonian graphs straight lines are on log scale implying that there is a power law relationship between the mass and radius. This can be seen from Fig. 3. One can estimate the exponent by two points on the curves, or by doing a fit. The result is

$$\Rightarrow M \propto R^{-3} \tag{1.7}$$

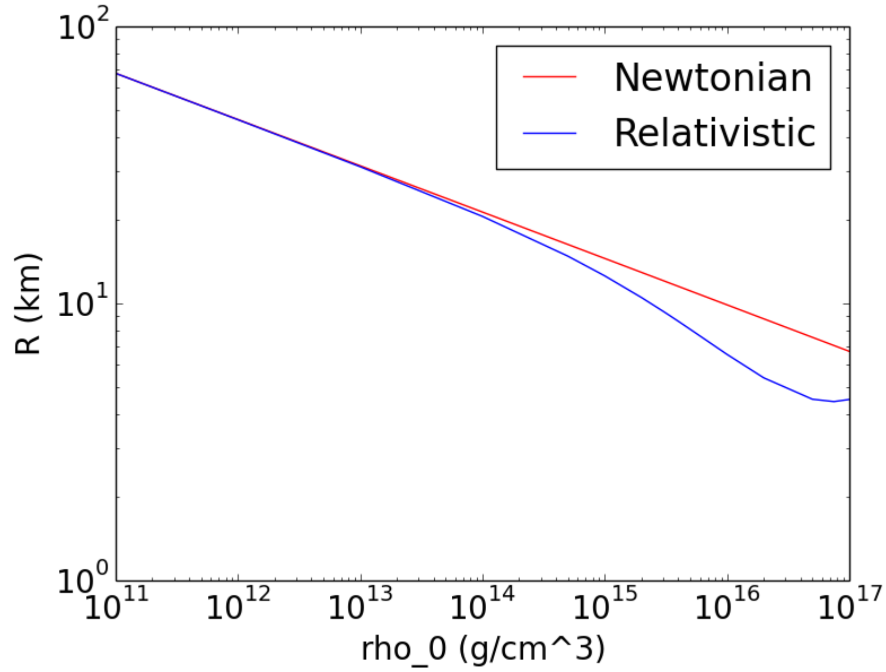

Figure 2: Radius vs central density.

(iii)

For all values of ρ_c, the Newtonian equations give that $\frac{dM}{d\rho_c} > 0$. This implies that all configurations are radially stable.

(iv)

Because the relation between mass, radius, and central density are all power laws, there is no maximum to the possible mass of these Newtonian neutron stars and no maximum to the central density.

Part (c)

The first law of thermodynamics, (assuming no entropy terms because of adiabaticity), from the class notes is

$$d\left(\frac{\rho}{n}\right) = -Pd\left(\frac{1}{n}\right) \tag{1.8}$$

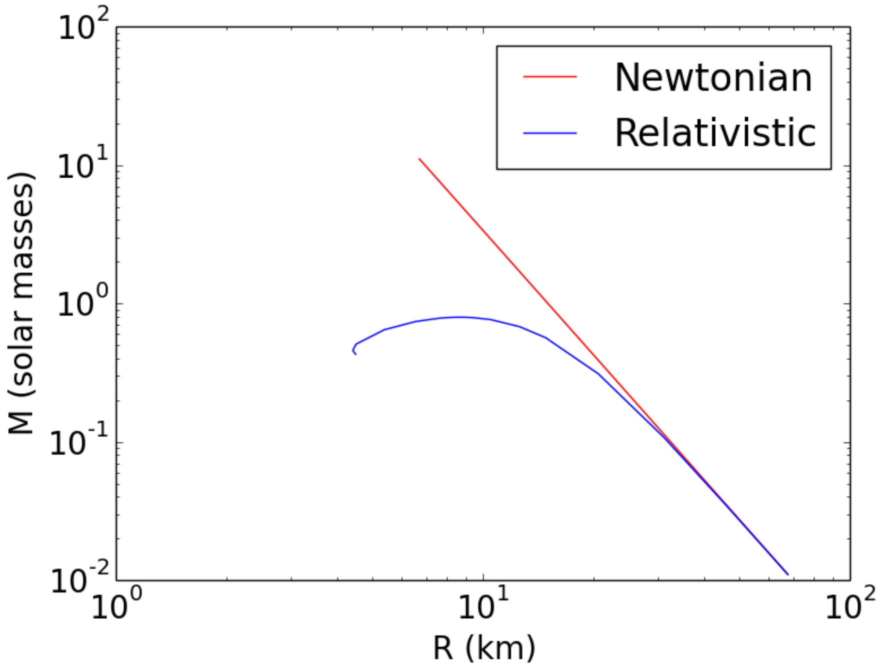

Figure 3: Mass vs Radius.

which can be rewritten in terms of ρ_0 by using the relation $\rho_0 = m_n n$ so then

$$d\left(\frac{\rho}{\rho_0}\right) = -Pd\left(\frac{1}{\rho_0}\right) \tag{1.9}$$

Integrate the expression, choosing the constant of integration so that the expression reduces to $\rho = \rho_0$ in the limit of no pressure (all of the energy is in the rest mass term)

$$\begin{aligned} \rho =& \rho_0 + \frac{K\rho_0^\Gamma}{\Gamma - 1} \\ \rho =& \rho_0 + \frac{3}{2}P \end{aligned} \tag{1.10}$$

This equation is the last equation needed to evolve the relativistic equations of stellar structure.

Part (d)

The relativistic equations of stellar structure are

$$\begin{aligned}\frac{dm}{dr} =& 4\pi r^2 \rho \\ \frac{dP}{dr} =& -\frac{Gm}{r^2}\rho\left(1+\frac{P}{\rho c^2}\right)\left(1+\frac{4\pi P r^3}{mc^2}\right)/\left(1-\frac{2Gm}{rc^2}\right)\end{aligned} \tag{1.11}$$

Note that here ρ is used instead of ρ_0 and that all of the dimensionful constants have been included. Again, choose dimensionless variables as before, adding a new dimensionless variable $\tilde{\rho} = \frac{\rho}{10^{15}\mathrm{g/cm^3}}$ for the energy density. Then the relevant equations to integrate are

$$\begin{aligned}\frac{d\tilde{m}}{d\tilde{r}} =& 6.318\tilde{r}^2\tilde{\rho} \\ \frac{d\tilde{P}}{d\tilde{r}} =& -2.4666\frac{\tilde{m}}{\tilde{r}^2}\tilde{\rho}\left(1+.59864\frac{\tilde{P}}{\tilde{\rho}}\right)\left(1+.3782\frac{\tilde{P}\tilde{r}^3}{\tilde{m}}\right)/\left(1-.2953\frac{\tilde{m}}{\tilde{r}}\right) \\ \tilde{\rho}_0 =& \tilde{P}^{1/\Gamma} \\ \tilde{\rho} =& \tilde{\rho}_0 + 0.089796\tilde{P}\end{aligned} \tag{1.12}$$

The method of calculation the initial and termination conditions are done in the same manner as the Newtonian case and the attached script has an example.

(i)

See Figs. 1–3 for the mass and radius profiles.

(ii)

Only the configurations with $\frac{dM}{d\rho_c} > 0$ are radially stable. Looking at the table printed out from the script, the maximum occurs around $\rho_{c,max} = 4.2245 \times 10^{15}\mathrm{g/cm^2}$ so all configurations below this are stable.

(iii)

The maximum stable mass is found at $\rho_{c,max}$ which according to the table is approximately $0.784908 M_\odot$.

(iv)

The maximum redshift of a photon emitted from the surface is

$$z_{max} = \left(1-\frac{2m_{max}}{R_{max}}\right)^{-1/2} - 1 \tag{1.13}$$

Putting in values of $m_{max} \approx .8M_\odot$ and $R_{max} \approx 8.5\mathrm{km}$ gives a value for $z_{max} \approx$ .18

Problem 2

Part (a)

Because the particle is being held by the string, it is not in free-fall. Rather the string is exerting a force on the particle which has a 4-acceleration given by

$$\begin{aligned}\vec{a} &= \nabla_{\vec{u}}\vec{u}\\ a^\beta &= u^\alpha u^\beta{}_{;\alpha} = u^\alpha(u^\beta{}_{,\alpha} + \Gamma^\beta{}_{\delta\alpha}u^\delta)\end{aligned} \tag{2.1}$$

Because the particle is held stationary, it's 4-velocity must be $\vec{u} = (u^0, \underset{\sim}{0})$ since its spatial coordinates are not moving. Therefore,

$$\begin{aligned}\vec{u}\cdot\vec{u} = -1 &= -u^0\left(1 - \frac{2M}{r}\right)\\ u^0 &= \left(1 - \frac{2M}{r}\right)^{-1/2}\end{aligned} \tag{2.2}$$

Now $\vec{a}\cdot\vec{u} = 0 \Rightarrow a^0 = 0$. The acceleration is then

$$\begin{aligned}a^i =& u^\alpha(u^\beta{}_{,\alpha} + \Gamma^\beta{}_{\delta\alpha}u^\delta)\\ =& \Gamma^i{}_{00}{u^0}^2\\ =& \Gamma^i{}_{00}\left(1 - \frac{2M}{r}\right)^{-1}\end{aligned} \tag{2.3}$$

Since using a coordinate basis and assuming a static metric ($g_{\mu\nu,0} = 0$)

$$\begin{aligned}\Gamma^i{}_{00} =& \frac{1}{2}g^{i\mu}(2g_{\mu 0,0} - g_{00,\mu})\\ =& -\frac{1}{2}g^{i\mu}g_{00,\mu}\\ =& -\frac{1}{2}g^{ir}g_{00,r}\\ \Gamma^r{}_{00} =& -\frac{1}{2}g^{rr}g_{00,r} = \frac{M}{r^2}\left(1 - \frac{2M}{r}\right) \qquad (2.4)\\ \Gamma^\theta{}_{00} =& \Gamma^\phi{}_{00} = 0 \qquad (2.5)\end{aligned}$$

Thus, the acceleration is $\vec{a} = (0, \frac{m}{r^2}, 0, 0)$. But the "measured Newtonian acceleration in the local frame of the particle" is given by the invariant quantity

$$\begin{aligned}a_{Newt} = |\vec{a}\cdot\vec{a}| = a^r {g_{rr}}^{1/2} &= \frac{M}{r^2}\frac{1}{(1 - 2M/r)^{1/2}}\\ \Rightarrow F = ma_{Newt} &= \frac{Mm}{r^2}\frac{1}{(1 - 2M/r)^{1/2}}\end{aligned} \tag{2.6}$$

Part (b)

Imagine that the distant observer pulls on the string for a proper distance $d\ell_{obs}$. He does work on the system, $\delta W_{obs} = F_{obs} d\ell_{obs}$. Meanwhile, the string pulls on the particle, doing work $\delta W_{part} = F d\ell_{obs}$ as measured locally (the proper distance move must be the same, assuming the string did not stretch). Energy conservation demands that the work done by the distant observer must be equal to the work done on the particle by the string, after correcting for gravitational redshift factor $(1-2M/r)^{1/2}$.

$$\begin{aligned} \delta W_{obs}(1-2M/r_{obs})^{1/2} =& \delta W_{part}(1-2M/r)^{1/2} \\ \delta F_{obs} d\ell_{obs}(1-2M/r_{obs})^{1/2} =& F d\ell_{part}(1-2M/r)^{1/2} \\ F_{obs} =& F(1-2M/r)^{1/2}(1-2M/r_{obs})^{-1/2} \\ F_{obs} =& \frac{Mm}{r^2}(1-2M/r_{obs})^{-1/2} \end{aligned} \tag{2.7}$$

In the limit where the observer and the particle are at the same place, $r = r_{obs}$, the original formula for the force in part (a) is recovered.

Part (c)

Take the expression just found for F_{obs} and take the limit as $r_{obs} = \infty$. Then the force is $F = \frac{Mm}{r^2}$, which is just the Newtonian expression for the gravitational force acting between two masses.

Example python script for problem 1

```
from __future__ import division

print 'Table of values for Newtonian neutron stars'
print 'rho0_c\t\tR (km)\t\tM (solar masses)'

rho0_c     = [1.e-4,1.e-3,1.e-2,.1,1.0,10.,100.]
M_Newton  = []
R_Newton  = []

#Choosing a smaller delta_r value should result in more accurate reslts
delta_r = .001

# Build a number of stars with different central densities
for i in range(len(rho0_c)):

  # Set up initial conditions
```

```
  r_new    = delta_r
  m_new    = 2.106*rho0_c[i]*(r_new**3)
  P_new    = pow(rho0_c[i],5./3.)
  rho0_new = rho0_c[i]

  # Use simple Euler integration to bulid the star
  while P_new>0:

    # Update the old steps
    m_old    = m_new
    P_old    = P_new
    rho0_old = rho0_new
    r_old    = r_new

    # Step forward in radius by one interval
    r_new    = r_old+delta_r
    m_new    = m_old + 2.106*rho0_old*(r_new**3-r_old**3)
    P_new    = P_old + 2.4666*rho0_old*m_old*(1./r_new - 1./r_old)
    rho0_new = pow(abs(P_new),3./5.)

  # Done with loop.
  # Use simple linear interpolation to find where exactly P=0 (i.e. star surface)
  frac = -P_old/(P_new-P_old) # How far along the last interval P=0
  R_Newton.append(r_old+frac*delta_r ) # Value of Radius and mass at surface
  M_Newton.append(m_old+frac*(m_new-m_old))

  # Rescale these quantities for simplicity of the plot
  rho0_c[i]   = rho0_c[i]*10**15
  R_Newton[i] = R_Newton[i]*10

  print '{}\t\t{:.6f}\t{:.6f}'.format(rho0_c[i],R_Newton[i],M_Newton[i])

print '\n\n'
####################

print 'Table of values for relativistic neutron stars\n'
print 'rho0_c\t\tR (km)\t\tM (solar masses)\trho_c'

rho0_C  = [1.e-4,1.e-3,1.e-2,.1,.5,1.,2.,3.,3.2,3.4,3.5,3.6,3.7,3.8,4.,4.5,5.,6.,10.,20.,50.
rho_C   = []
M_GR    = []
R_GR    = []

#Choosing a smaller delta_r value should result in more accurate reslts
delta_r = .001
```

```
# Build a number of stars with different central densities
for i in range(len(rho0_C)):

  # Set up initial conditions
  rho0_new = rho0_C[i]
  P_new    = pow(rho0_C[i],5./3.)
  rho_new  = rho0_new + .089796*P_new
  r_new    = delta_r
  m_new    = 2.106*rho_new*(r_new**3)

  rho_C.append(rho0_new + .089796*P_new)

  # Use simple Euler integration to bulid the star
  while P_new>0:

    # Update the old steps
    m_old    = m_new
    P_old    = P_new
    rho_old  = rho_new
    rho0_old = rho0_new
    r_old    = r_new

    # Step forward in radius by one interval
    r_new    = r_old+delta_r
    m_new    = m_old + 2.106*rho_old*(r_new**3-r_old**3)
    P_new    = P_old + (2.4666*rho_old*m_old*(1./r_new-1./r_old)) * (1.+.059864*P_old/rho_ol
    rho0_new = pow(abs(P_new),3./5.)
    rho_new  = rho0_new + .089796*P_new

  # Use simple linear interpolation to find where P=0 (i.e. star surface)
  frac = -P_old/(P_new-P_old) # How far along the last interval P=0
  R_GR.append( r_old+frac*delta_r )
  M_GR.append( m_old+frac*(m_new-m_old) )

  # Rescale these quantities for simplicity in the plots.
  rho0_C[i] = rho0_C[i]*10**15
  rho_C[i]  = rho_C[i]*10**15
  R_GR[i]   = R_GR[i]*10

  print '{}\t\t{:.6f}\t{:.6f}\t\t{:01.6g}'.format(rho0_C[i],R_GR[i],M_GR[i],rho_C[i])


# Plot Mass and Radius as a function of central density
"""
import pylab as plt
import matplotlib
```

```
plt.close('all')
matplotlib.rcParams.update({'font.size':20})
plt.figure(1);
plt.axes([0.15,0.15,0.8,0.8])
plt.loglog(rho0_c,M_Newton,'r',label='Newtonian')
plt.loglog(rho0_C,M_GR,'b',label='Relativistic')
plt.legend()
plt.xlabel('rho_0 (g/cm^3)')
plt.ylabel('M (solar masses)')
plt.figure(2);
plt.axes([0.15,0.15,0.8,0.8])
plt.loglog(rho0_c,R_Newton,'r',label='Newtonian')
plt.loglog(rho0_C,R_GR,'b',label='Relativistic')
plt.xlabel('rho_0 (g/cm^3)')
plt.ylabel('R (km)')
plt.legend()
plt.figure(3);
plt.axes([0.15,0.15,0.8,0.8])
plt.loglog(R_Newton,M_Newton,'r',label='Newtonian')
plt.loglog(R_GR,M_GR,'b',label='Relativistic')
plt.ylabel('M (solar masses)')
plt.xlabel('R (km)')
plt.legend()
plt.show()
"""
# maximum occurs for rho0_C ~ 3.5e+15, M ~ 0.784908

print 'The end'
```