Draft version April 7, 2017
Preprint typeset using LaTeX style AASTeX6 v. 1.0

HW 07: EQUATIONS OF STATE AND THE TEMPERATURE-DENSITY PLANE

Bryan Yamashiro[1]
University of Hawaii at Manoa
2500 Campus Road
Honolulu, HI 96822

1. DIFFERENT FORMULAS FOR PRESSURE

Radiation Pressure

$$P_{rad} = \frac{4\sigma T^4}{3c} \tag{1}$$

Ideal Gas Pressure

$$P_{ideal} = \frac{\rho k T}{\mu m_H} \tag{2}$$

Non-Relativistic Electron Degeneracy Pressure

$$P = 10^{13} \left(\frac{\rho}{\mu_e}\right)^{5/3} \tag{3}$$

Extremely Relativistic Electron Degeneracy Pressure

$$P = 1.245 \times 10^{15} \left(\frac{\rho}{\mu_e}\right)^{4/3} \tag{4}$$

2. AREAS OF PREPONDERANCE IN THE LOG(T) - LOG(ρ) PLANE

The four equations 1-4 for the different formulas for pressure were equated against each other for this section. Figure 1 characterizes the four domain regions in in the T - log ρ plane, where radiation pressure (1), ideal gas (2), degenerate gas (3), and relativistic gas (4) dominate. The four regions in figure 1 includes the boundaries radiation-ideal, ideal-degenerate (non-relativistic), ideal-degenerate (relativistic), and degenerate (non-relativistic)-degenerate (relativistic).

Figure 1 also demonstrates the core properties with a temperature at 15,000,000 K and density of 150 g cm^{-3}, which are indicated by the orange dashed lines and, more specifically, the blue circle. The plot shows that the core equation of state is most adequate in the $\boxed{ideal - gas}$ pressure region.

2.1. *Equation of Boundaries*

Ideal Gas and Radiation Pressure Boundary

$$T_{ideal-rad} = \left(\frac{3c\rho k_B}{4\sigma \mu m_H}\right)^{1/3} \tag{5}$$

Ideal Gas and Non-Relativistic Electron Degeneracy Pressure Boundary

$$T_{ideal-nonrel} = \frac{\mu m_H 10^{13} \left(\frac{\rho}{\mu_e}\right)^{5/3}}{\rho k_B} \tag{6}$$

Ideal Gas and Extremely Relativistic Electron Degeneracy Pressure Boundary

$$T_{ideal-rel} = \frac{\mu m_H 1.245 \times 10^{15} \left(\frac{\rho}{\mu_e}\right)^{4/3}}{\rho k_B} \tag{7}$$

Non-Relativistic and Extremely Relativistic Degeneracy Pressure Boundary

$$\frac{\rho^{5/3}}{\rho^{4/3}} = \frac{1.245 \times 10^{15} \mu_e^{5/3}}{10^{13} \mu_e^{4/3}} \tag{8}$$

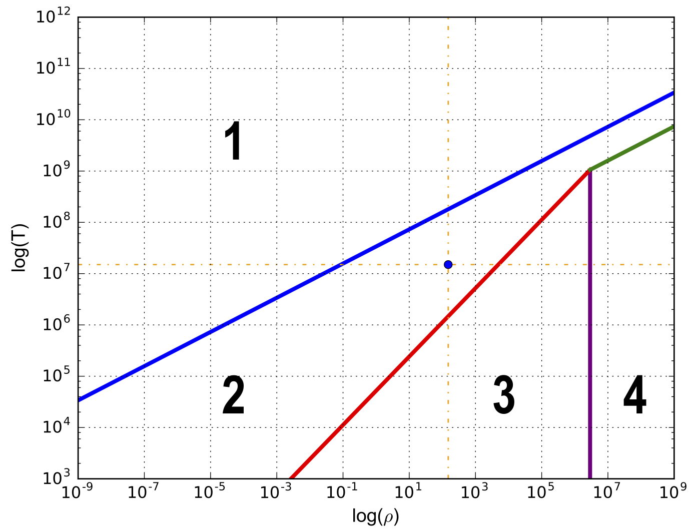

Figure 1. Domains of the validity of the ideal-gas approximation, radiation pressure, degenerate gas, and relativistic degenerate gas. The colors represent the boundaries radiation-ideal [blue], ideal-degenerate (non-relativistic) [red],ideal-degenerate (relativistic) [green], degenerate (non-relativistic)-degenerate (relativistic) [purple]. The blue circle and orange dashed lines represents the temperature at 15,000,000 K and a density of 150 g cm^{-3}. The regimes are labeled are where each pressure scheme dominates, including radiation (1), ideal gas (2), non-relativistic degenerate gas (3), and extremely-relativistic degenerate gas (4).

3. APPENDIX

```python
from astropy import units as u
from astropy.units import imperial as imp
#import astropy.units as u
import numpy as np
from astropy import constants as const
import matplotlib.pyplot as plt
import scipy
from scipy import special

mu_e = 1.5
mu = 0.85
k_1 = 1.00*10.0**7.0
k_2 = 1.24*10.0**11.0

rho = np.arange(10**(-9),10**9,1000.)
rho_non = np.arange(10**(-9),2.89467*10**6,1000.)
rho_rel = np.arange(2.89467*10**6,10**9,1000.)
#temp = np.arange(3,12,0.01)

inner_rho = (rho / mu_e)
inner_rho_non = (rho_non / mu_e)
inner_rho_rel = (rho_rel / mu_e)


T_1 = (((3.0 * const.c * const.k_B * rho)/(4.0 * const.sigma_sb * mu * const.u))**(1./3.)).cgs

#using book approximation (proportionality so ignore this case)
#T_2 = ((mu/(rho * const.k_B))*((rho/mu_e)**(5.0/3.0))).cgs
#T_3 = ((mu/(rho * const.k_B))*((rho/mu_e)**(4.0/3.0))).cgs

#online constants used to derive
#T_2 = k_1 * (rho**(5./3.))
#T_3 = k_2 * (rho**(4./3.))

#online constants used to derive
#T_2 = k_1 * (rho**(5./3.))
#T_3 = k_2 * (rho**(4./3.))

#slide constants
#T_2 = (10.0**13.0)*((rho/mu_e)**(5.0/3.0))
#T_3 = (1.245*(10.0**15.0))*((rho/mu_e)**(4.0/3.0))

T_2 = ((mu * const.u * (10.0**13.0) * (inner_rho_non ** (5.0/3.0)) ) / (rho_non * const.k_B)).cgs
T_3 = ((mu * const.u * (1.245*(10.0**15.0)) * (inner_rho_rel ** (4.0/3.0)) ) / (rho_rel * const.k_B)
    ).cgs


#idx = np.argwhere(np.diff(np.sign(T_3 - T_2)) != 0).reshape(-1) + 0



plt.plot(rho, T_1, color='blue', linewidth = 3)
plt.plot(rho_non, T_2,  color='red', linewidth = 3)
plt.plot(rho_rel, T_3, color='green', linewidth = 3)

#idx = np.argwhere(np.isclose(T_2, T_3, atol=0.1)).reshape(-1)
#plt.plot(rho[idx], T_3[idx], 'ro')
#ax1.axvline(goes_proton_time[max_index], color='black', linewidth=1)
plt.vlines(x=2.89467*10.0**6.0, ymin = 0, ymax = 1*10.0**9.0, color='purple', linewidth = 3)
plt.axvline(x=150, ymin=0, ymax=10., hold=None,linestyle = '-.', color='orange')
plt.axhline(y=15000000.0, hold=None,linestyle = '-.', color='orange')
plt.plot(150, 15000000.0, 'ob')


#plt.plot(lambda_queue, d_gaussian, color='red')
plt.xlabel(r'log($ \rho $)', fontname="Arial", fontsize = 14)
plt.ylabel('log(T)', fontname="Arial", fontsize = 14)
plt.minorticks_on()
plt.ylim([10**3,10**12])
plt.xlim([10**(-9),10**(9)])
plt.grid(True)
plt.yscale('log')
```

```
plt.xscale('log')

plt.savefig('plot.pdf', format='pdf', dpi=900)

plt.show()
```

Listing 1. Python source code.