

HW11-RyanSponzilli

November 16, 2024

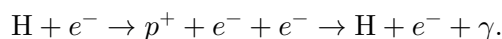
```
[24]: import numpy as np
      from scipy import linalg
      import matplotlib.pyplot as plt
      import scipy.interpolate
```

1 ASTR 310 Homework 11

1.0.1 1. Collisional ionization equilibrium

The setup for this problem is long, but the problem itself is not complex. Read it all the way through before you try to solve it.

At very low densities and high temperatures, gas can exist in a state called collisional ionization equilibrium, in which collisions between atoms and electrons are energetic enough to produce ionization, but recombination of ions and electrons produces photons that escape from the gas without interacting further. For a pure hydrogen gas, the situation looks like this (assuming the initial free electron is “borrowed” from another ionization):



The photons (γ) gradually carry away energy from the gas, but the rate of energy loss is much slower than the rate at which the ratio of ionized (p^+) to neutral (H) hydrogen achieves a steady value.

\$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$

Suppose we have a hydrogen plus helium gas at density ρ and temperature T with a hydrogen mass fraction X . The number density of hydrogen atoms, whether ionized or neutral, is $n_{\text{H}} \approx \rho X / m_p$, and that of helium is $n_{\text{He}} \approx \rho(1 - X) / (4m_p)$. Let $x_{\text{H}+}$ represent the fraction of hydrogen atoms that are ionized, and let $x_{\text{He}+}$ and $x_{\text{He}++}$ represent the fractions of helium atoms that are singly or doubly ionized, respectively. Balancing the rates of collisional ionizations with the rates of radiative recombinations yields the equations

$$\begin{aligned} n_{\mathrm{H}}(1-x_{\mathrm{H}+})n_e C_{\mathrm{H}} &= n_{\mathrm{H}}x_{\mathrm{H}+}n_e R_{\mathrm{H}+} \\ n_{\mathrm{He}}(1-x_{\mathrm{He}+}-x_{\mathrm{He}++})n_e C_{\mathrm{He}} &= n_{\mathrm{He}}x_{\mathrm{He}+}n_e R_{\mathrm{He}+} \\ n_{\mathrm{He}}x_{\mathrm{He}+}n_e C_{\mathrm{He}+} &= n_{\mathrm{He}}x_{\mathrm{He}++}n_e R_{\mathrm{He}++} \end{aligned}$$

where n_e is the number density of free electrons, C_i is the temperature-dependent collisional ionization rate coefficient for species i , and R_i is the temperature-dependent radiative recombination

rate coefficient for species i . (For aficionados: we neglect charge exchange and double ionization processes.) The electron number density is given by charge neutrality as

$$n_e = n_{\text{H}} + 2n_{\text{He}},$$

but we will not need it here, since it divides out. (Check this for yourself.)

Rearranging and eliminating common factors, these equations become the following.

$$\begin{aligned} (C_{\text{H}} + R_{\text{H}+}) x_{\text{H}+} &= C_{\text{H}} \\ (C_{\text{He}} + R_{\text{He}+}) x_{\text{He}+} + C_{\text{He}} x_{\text{He}++} &= C_{\text{He}} \\ C_{\text{He}+} x_{\text{He}+} - R_{\text{He}++} x_{\text{He}++} &= 0. \end{aligned}$$

These make a linear system whose solution is the vector $[x_{\text{H}+}, x_{\text{He}+}, x_{\text{He}++}]$. Notice that the coefficients multiplying $x_{\text{He}+}$ and $x_{\text{He}++}$ in the first equation are zero, and the coefficients multiplying $x_{\text{H}+}$ in the second and third equations are also zero. So the equations can be written in matrix form as

$$\begin{bmatrix} C_{\text{H}} + R_{\text{H}+} & 0 & 0 \\ 0 & C_{\text{He}} + R_{\text{He}+} & C_{\text{He}} \\ 0 & C_{\text{He}+} & -R_{\text{He}++} \end{bmatrix} \begin{bmatrix} x_{\text{H}+} \\ x_{\text{He}+} \\ x_{\text{He}++} \end{bmatrix} = \begin{bmatrix} C_{\text{H}} \\ C_{\text{He}} \\ 0 \end{bmatrix}.$$

The rate coefficients are given as functions of temperature approximately by \$ \$ \$

$$\begin{aligned} C_{\text{H}}(T_4) &\approx 5.8 \times 10^{-9} T_4^{1/2} e^{-15.78/T_4} \text{ cm}^3 \text{ s}^{-1} \\ R_{\text{H}+}(T_4) &\approx 4.13 \times 10^{-13} T_4^{-0.7131-0.115 \ln T_4} \text{ cm}^3 \text{ s}^{-1} \\ C_{\text{He}}(T_4) &\approx 2.8 \times 10^{-9} T_4^{1/2} e^{-28.54/T_4} \text{ cm}^3 \text{ s}^{-1} \\ R_{\text{He}+}(T_4) &\approx 1.54 \times 10^{-13} T_4^{-0.486} + 2.72 \times 10^{-13} T_4^{-0.789} \text{ cm}^3 \text{ s}^{-1} \\ C_{\text{He}+}(T_4) &\approx 4.3 \times 10^{-10} T_4^{1/2} e^{-63.13/T_4} \text{ cm}^3 \text{ s}^{-1} \\ R_{\text{He}++}(T_4) &\approx 1.65 \times 10^{-12} \left(\frac{T_4}{4} \right)^{-0.7131-0.115 \ln(T_4/4)} \text{ cm}^3 \text{ s}^{-1} \end{aligned}$$

where $T_4 \equiv T/(10^4\text{K})$. So the ionization fractions $x_{\text{H}+}$, $x_{\text{He}+}$, and $x_{\text{He}++}$ are functions of temperature: given a temperature, we compute the C and R coefficients, then substitute for them in the matrix equation above and solve the corresponding linear system to find the ionization fractions at that temperature.

a) Write a function that, given the temperature in K, returns the equilibrium coefficient matrix and the right-hand side column vector in the matrix equation, in cgs units. Evaluate and print the coefficient matrix for $T_4 = 10$.

[14 pts]

```
[25]: def create_matrix(T4):
    C_H = 5.8e-9 * (T4**0.5) * np.exp(-15.78/T4)
    R_H_plus = 4.13e-13 * (T4**(-0.7131 - 0.115*np.log(T4)))
    C_He = 2.8e-9 * (T4**0.5) * np.exp(-28.54/T4)
    R_He_plus = 1.54e-13 * (T4**-0.486) + 2.72e-13 * T4**-0.789
    C_He_plus = 4.3e-10 * (T4**0.5) * np.exp(-63.13/T4)
    R_He_plus_plus = 1.65e-12 * (T4/4)**(-0.7131-0.115*np.log(T4/4))

    A = np.array([
        [C_H+R_H_plus, 0, 0],
        [0, C_He+R_He_plus, C_He],
        [0, C_He_plus, -R_He_plus_plus]
    ])

    b = np.array([
        C_H,
        C_He,
        0
    ])

    return A, b

create_matrix(10)
```

```
[25]: (array([[ 3.78543930e-09,  0.00000000e+00,  0.00000000e+00],
               [ 0.00000000e+00,  5.10225350e-10,  5.10130840e-10],
               [ 0.00000000e+00,  2.46471888e-12, -7.79436283e-13]]),
       array([3.78539585e-09, 5.10130840e-10, 0.00000000e+00]))
```

b) Construct a 1D array containing 100 logarithmically spaced temperatures between 3000 K and 3×10^5 K. For each temperature value, use your function from part (a) to set up the matrix equation and solve it to obtain the three ionization fractions at that temperature. Store the three ionization fractions corresponding to each temperature in three other 1D arrays.

[14 pts]

```
[26]: x = 3*np.logspace(3, 5, 100, base=10)

x_H_plus = []
x_He_plus = []
x_He_plus_plus = []

for T in x:
    A, b = create_matrix(T/1e4)
    X = linalg.solve(A, b)
    x_H_plus.append(X[0])
    x_He_plus.append(X[1])
    x_He_plus_plus.append(X[2])
```

```
x_H_plus = np.array(x_H_plus)
x_He_plus = np.array(x_He_plus)
x_He_plus_plus = np.array(x_He_plus_plus)
```

c) Interpolate to find at what temperatures the ionization fractions of those three species are equal to 50%. Possible pitfall: note that the abundance of singly ionized helium is not monotonic with temperature.

[10 pts]

```
[40]: # find x for y=0.5
interp = scipy.interpolate.interp1d(x_H_plus, x, kind='linear')
x_H_plus_half = interp(0.5)

interp = scipy.interpolate.interp1d(x_He_plus[0:50], x[0:50], kind='linear')
x_He_plus_half = interp(0.5)

interp = scipy.interpolate.interp1d(x_He_plus_plus, x, kind='linear')
x_He_plus_plus_half = interp(0.5)
```

d) Plot the ionization fractions as functions of temperature on a single log-linear plot using curves of different colors. Include a proper legend and axis labels, and mark the interpolated 50% points with `plt.axvline`.

[12 pts]

```
[44]: plt.plot(x, x_H_plus, label="$x_{H+}$")
plt.plot(x, x_He_plus, label="$x_{He+}$")
plt.plot(x, x_He_plus_plus, label="$x_{He++}$")

plt.axvline(x_H_plus_half, c='blue', ls='dotted')
plt.axvline(x_He_plus_half, c='orange', ls='dotted')
plt.axvline(x_He_plus_plus_half, c='green', ls='dotted')

plt.xscale('log')
plt.legend()
plt.xlabel("Temperature")
plt.ylabel("Ionization Fraction")
```

```
[44]: Text(0, 0.5, 'Ionization Fraction')
```

