HW 6

Jean and Atmospheric Escape

Jesus Javier Serrano

In [4]:
```
import numpy as np
import scipy as sc

#Define the parameters
T_exo = 1e4 #kelvi
P_exo = 1e-3 #Pa
orb_rad = 0.03*(149.6)*1000
vol_rad = 2.7e7
g_pl = 10
m_H = 1.01/(6.022e23)/(1000)
m_He = 4.003/(6.022e23)/(1000)
k_b = 1.38064852e-23 #J/k
R = 8.314 #J/(K*mol) = Pa *m^3/(k*mol)
G = 6.67408e-11 #grav constant

#Define the Jean's parameter(v_esc/v_o)^2
def lam_esc(m):
    a = m*g_pl*(vol_rad + ((R*T_exo)/(m*(6.022e23)*g_pl)))
    b = k_b*T_exo
    return a/b
#Define the Jean's escape rate
def psi_j(m, N_ex):
    max_vel = (2*k_b*T_exo/m)**0.5
    front = N_ex*max_vel/(2*(np.pi**0.5))
    expo_term = (1 + lam_esc(m))*np.exp(-lam_esc(m))
    return front*expo_term

#Given the above parameters how do we find the number denisty of Hydrogen and
 Helium?
N_exo = P_exo/(R*T_exo)*(6.022e23)
N_H_exo = N_exo/1.09705
N_He_exo = 0.09705*N_H_exo

class GJ436b:
    def __init__(self, H_quality, He_quality, mass):
        self.H_quality = H_quality
        self.He_quality = He_quality
        self.mass = mass
H_prop = {}
He_prop = {}
part_lis = ['H', 'He']
for atom in part_lis:
    mass = 'm_%s'%atom
    plug = '%s_prop'%atom
    dens = 'N_%s_exo'%atom
    jean_param = lam_esc(eval(mass))
    jean_flux = psi_j(eval(mass), eval(dens))
    properties = eval(plug)
    properties['mass'] = eval(mass) #in kilograms
    properties['j_param'] = jean_param
    properties['j_flux'] = jean_flux

GJ436b.H_quality= H_prop
GJ436b.He_quality= He_prop
plan_mass = g_pl*((vol_rad)**2)/G
GJ436b.mass = plan_mass
```

```
#answer
print('The planet GJ436b has a mass of %.3e kg'%GJ436b.mass)
print('The Jean parameter and Jean escape rate for Hydrogen is %.3f and %.3e m
^-2 s^-1'%(GJ436b.H_quality['j_param'], GJ436b.H_quality['j_flux']))
print('The Jean parameter and Jean escape rate for Helium is %.3f and %.3e m^-
2 s^-1'%(GJ436b.He_quality['j_param'], GJ436b.He_quality['j_flux']))

#approximation of total escaped Hydrogen and Helium
dead_m_H =  GJ436b.H_quality['j_flux']*4*np.pi*(vol_rad**2)*(10**10)*m_H
dead_m_He =  GJ436b.He_quality['j_flux']*4*np.pi*(vol_rad**2)*(10**10)*m_He

print('%.3e kg of Hydrogen escaped from the atmosphere of the planet'%dead_m_H
)
print('%.3e kg of Helium escaped from the atmosphere of the planet'%dead_m_He)
ratio_m_H = dead_m_H/GJ436b.mass
ratio_m_He = dead_m_He/GJ436b.mass
print('The ratio of escaped Hydrogen mass to planetary mass is %.3e'%ratio_m_H
)
print('The ratio of escaped Helium mass to planetary mass is %.3e'%ratio_m_He)

#Find the diffusion-limited escape
b = 1e23 #m^-1 s^-1
#mole-fraction of the gases
mf_H = 1/1.09705
mf_He = 0.09705/1.09705
#the flux
def diff_lim(m, chi):
    top = b*m*g_pl*chi
    bott = k_b*T_exo
    return top/bott

lim_esc_H = diff_lim(m_H, mf_H)
ratio_H_esc = lim_esc_H/GJ436b.H_quality['j_flux']
lim_esc_He = diff_lim(m_He, mf_He)
ratio_He_esc = lim_esc_He/GJ436b.He_quality['j_flux']
print('The diffusion-limited flux for Hydrogen is calculated to be: %.3e m^-2
 s^-1 which is %.3f times the Jean flux'%(lim_esc_H, ratio_H_esc))
print('The diffusion-limited flux for Helium is calculated to be: %.3e m^-2 s^
-1  which is %.3f times the Jean flux'%(lim_esc_He, ratio_He_esc))

#approximation of total escaped Hydrogen and Helium
lim_dead_m_H =  lim_esc_H*4*np.pi*(vol_rad**2)*(10**10)*m_H
lim_dead_m_He =  lim_esc_He*4*np.pi*(vol_rad**2)*(10**10)*m_He

print('%.3e kg of Hydrogen escaped from the atmosphere of the planet'%lim_dead
_m_H)
print('%.3e kg of Helium escaped from the atmosphere of the planet'%lim_dead_m
_He)
lim_ratio_m_H = lim_dead_m_H/GJ436b.mass
lim_ratio_m_He = lim_dead_m_He/GJ436b.mass
print('The ratio of escaped Hydrogen mass to planetary mass is %.3e'%lim_ratio
_m_H)
print('The ratio of escaped Helium mass to planetary mass is %.3e'%lim_ratio_m
_He)
```

```
The planet GJ436b has a mass of 1.092e+26 kg
The Jean parameter and Jean escape rate for Hydrogen is 4.280 and 1.747e+18 m
^-2 s^-1
The Jean parameter and Jean escape rate for Helium is 13.999 and 1.454e+13 m^
-2 s^-1
2.684e+17 kg of Hydrogen escaped from the atmosphere of the planet
8.853e+12 kg of Helium escaped from the atmosphere of the planet
The ratio of escaped Hydrogen mass to planetary mass is 2.457e-09
The ratio of escaped Helium mass to planetary mass is 8.105e-14
The diffusion-limited flux for Hydrogen is calculated to be: 1.107e+16 m^-2 s
^-1 which is 0.006 times the Jean flux
The diffusion-limited flux for Helium is calculated to be: 4.259e+15 m^-2 s^-
1  which is 292.957 times the Jean flux
1.701e+15 kg of Hydrogen escaped from the atmosphere of the planet
2.594e+15 kg of Helium escaped from the atmosphere of the planet
The ratio of escaped Hydrogen mass to planetary mass is 1.558e-11
The ratio of escaped Helium mass to planetary mass is 2.375e-11
```

After obtaining the proper mass flux of Hydrogen and Helium escaping from the atmosphere we may approximate the atmosphere mass and pressure of GJ 436b. If we assume that the current atmosphere of this planet is 99.9% Helium then the total mass of the current atmosphere may be calculated by using the column density equation. Since we know that for an Ideal gas the column density $c\rho$ with the column base located at z_o is defined as:

$$c\rho = \int_{z_o}^{\infty} N(z)dz = N(z_o)H$$

where $H = \frac{RT}{m_{atm} g_{pl}}$ is the height scale $N(z)$ is the number density of the air molecules defined as:

$$N(z) = N_o e^{\frac{-z}{H}}$$

where N_o is the number density of air molecules from the surface. With this definition we can decipher the column density with the base of the column being the surface by noting that $N(z_o) = N_o e^{-1}$. Let us make z_o be the height of exobase so $N(z_o) = N_{ex}$ thus $N_o = N_{ex} e$. Since H is a function of the mass of the atmosphere then:

$$m_{atm} = m^w_{atm} n_{atm}$$

where $m^w_{atm} = m^W_{He}/0.999$ is the molar mass of air which can be found with the molar mass of Helium, and $n_{atm} = \frac{N_{ex} e H (4\pi R^2_{vol})}{6.022X10^{23}}$ is the number of moles present in the atmosphere, then:

$$m_{atm} = \sqrt{\frac{m^W_{He}}{0.999} \frac{N_{ex} e R T (4\pi R^2_{vol})}{g_{pl} 6.022X10^{23}}}$$

In [9]:
```
#Look into the promordial atmosphere
dead_dif_m_H = lim_esc_H*4*np.pi*(vol_rad**2)*(10**10)*m_H
dead_dif_m_He = GJ436b.He_quality['j_flux']*4*np.pi*(vol_rad**2)*(10**10)*m_He

#mass of the atmosphere as an approximation
present_mass_atm = (((4.003/(1000*(6.022e23)))*N_exo*np.e*4*np.pi*(vol_rad**2)
*R*T_exo/(g_pl))/0.999)**0.5 #kg
print('The present mass of the atmosphere is %.3e kg'%present_mass_atm)
primor_mass_atm = present_mass_atm + dead_dif_m_H + dead_dif_m_He #kg
print('The primordial mass of the atmosphere is %.3e kg'%primor_mass_atm)

#Mass fraction of Hydrogen and Helium in the primordial case:
primor_mass_frac_He = (0.999*present_mass_atm + dead_dif_m_He)/primor_mass_atm
primor_mass_frac_H = (0.001*present_mass_atm + dead_dif_m_H)/primor_mass_atm
print('Primordial mass density ratio is %.2f : %.2f'%(primor_mass_frac_H, prim
or_mass_frac_He))
#assuming ideal gas we can apply the exponential law to the pressure profile o
f the atmoshpere
primor_pressure_surf = primor_mass_atm*np.e*R*T_exo/(((primor_mass_frac_He*m_H
e) + primor_mass_frac_H *m_H) *4*np.pi*(vol_rad**2)*R*T_exo/(g_pl*primor_mass_
atm))
solar_corrected_pressure_surf = primor_mass_atm*np.e*R*T_exo/(((0.1*m_He) + 0.
9 *m_H) *4*np.pi*(vol_rad**2)*R*T_exo/(g_pl*primor_mass_atm))
print('The primordial surface pressure of the atmosphere is %.3e Pa assuming n
o temperature gradient and a time-invariant temperature'%primor_pressure_surf)
print('The primordial surface pressure of the atmosphere with solar compositio
n is %.3e Pa assuming no temperature gradient and a time-invariant temperatur
e'%solar_corrected_pressure_surf)
present_pressure_surf = present_mass_atm*np.e*R*T_exo/(((0.999*m_He) + 0.001*m
_H) *4*np.pi*(vol_rad**2)*R*T_exo/(g_pl*present_mass_atm))
print('The present surface pressure of the atmosphere is %.3e Pa assuming no t
emperature gradient and a time-invariant temperature'%present_pressure_surf)
```

```
The present mass of the atmosphere is 9.989e+04 kg
The primordial mass of the atmosphere is 1.710e+15 kg
Primordial mass density ratio is 0.99 : 0.01
The primordial surface pressure of the atmosphere is 5.096e+42 Pa assuming no
temperature gradient and a time-invariant temperature
The primordial surface pressure of the atmosphere with solar composition is
3.992e+42 Pa assuming no temperature gradient and a time-invariant temperatur
e
The present surface pressure of the atmosphere is 4.457e+21 Pa assuming no te
mperature gradient and a time-invariant temperature
```

In [ ]: