Phys 600 HW 4 Josephine Baggen

Recombination

1) Derive:

$$\frac{1-X_e}{X_e^2} = \frac{2\zeta(3)}{\pi^2}\eta\left(\frac{2\pi T}{m_e}\right)^{3/2} e^{E_I/T}$$

recombination $e^- + p \leftrightarrow H + \gamma$

In equilibrium the abundances for e, p, H are

$$n_{eq} = g\left(\frac{m\, k_B T}{2\pi\hbar^2}\right)^{3/2}\exp\left(-\frac{mc^2-\mu}{k_B T}\right) \; *$$

* I use other lecture notes with constants $\hbar, k_B, c$

Because $\mu_\gamma = 0$ we have $\mu_e + \mu_p = \mu_H$

So this will fall out of the exponent when we take the ratios:

$$\left(\frac{n_H}{n_e n_p}\right)_{eq} = \left(\frac{g_H}{g_e g_p}\right)\left(\frac{m_H}{m_e m_p}\right)^{3/2}\left(\frac{k_B T}{2\pi\hbar^2}\right)^{-3/2}\exp\left(\frac{(-m_H+m_p+m_e)c^2}{k_B T}\right)$$

Simplifying $g_H = 2\times 2 = 4$

$g_e = 2$

$g_p = 2$

$m_H \approx m_p$

call $E_I = (m_p + m_e - m_H)c^2 = 13.6\text{ eV}$

$$\left(\frac{n_H}{n_e n_p}\right)_{eq} = \left(\frac{2\pi\hbar^2}{k_B T\, m_e}\right)^{3/2}\exp\left(E_I/k_B T\right)$$

Call $X_e = \frac{n_e}{n_b}$ fraction of free electrons $\rightarrow$

$n_b = n_H + n_p$ (neglecting He)

$n_e = n_p = n_b X_e$

$n_H = n_b - n_e = n_b(1-X_e)$

So we have $n_H = n_b(1-X_e)$

$n_e = n_p = n_b X_e$

plugging this in:

$$\left(\frac{n_H}{n_p n_e}\right)_{eq} = \left(\frac{2\pi\hbar^2}{k_B T m_e}\right)^{3/2} \exp\left(E_I/k_B T\right)$$

$$\frac{1-X_e}{X_e^2} = n_b \left(\frac{2\pi\hbar^2}{k_B T m_e}\right)^{3/2} \exp\left(E_I/k_B T\right)$$

Relating the number density of baryons to photons. $n_b = \eta n_\gamma = \eta \frac{2\zeta(3)}{\pi^2(\hbar c)^3}(k_B T)^3$

plugging this in gives

$$\frac{1-X_e}{X_e^2} = \eta \frac{2\zeta(3)}{\pi^2(\hbar c)^3}(k_B T)^3 \left(\frac{2\pi\hbar^2}{m_e}\right)^{3/2}(k_B T)^{-3/2} \exp\left(E_I/k_B T\right)$$

Setting constants to 1 $\hbar = c = k_B = 1$ to get the expression:

$$\frac{1-X_e}{X_e^2} = \eta \frac{2\zeta(3)}{\pi^2}\left(\frac{2\pi T}{m_e}\right)^{3/2} \exp\left(E_I/T\right)$$

2) Solve for Xe as a function of z

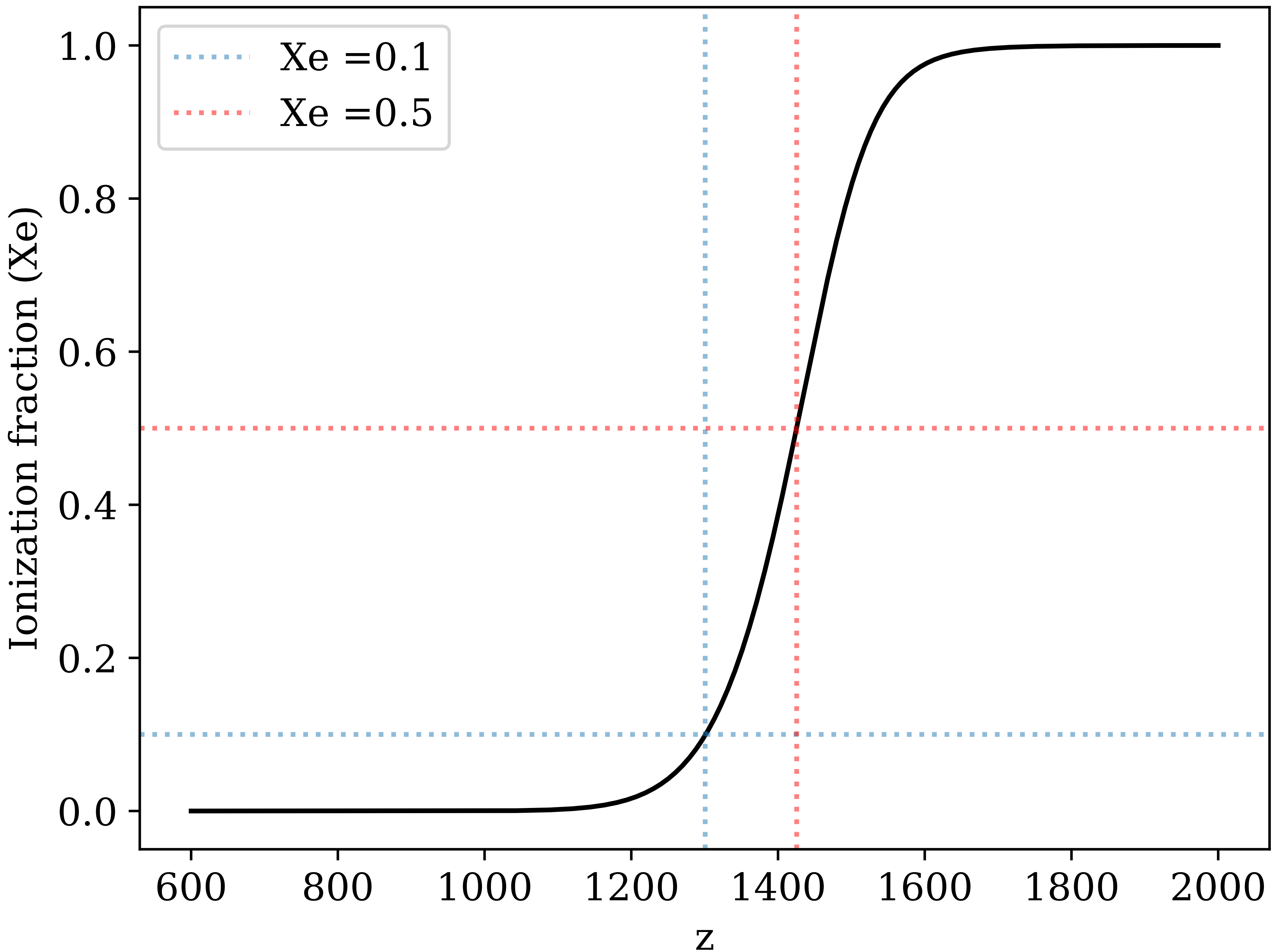
$a \propto \frac{1}{T}$ $\quad 1+z \propto T$

$$\frac{1+z}{1} = \frac{T}{T_0}$$

$T_0 = 2.73\ \mathrm{K}$

$= 0.23\ \mathrm{meV}$

$T = T_0(1+z)$


Xe =0.1
Xe =0.5
1.0
0.8
0.6
0.4
0.2
0.0
Ionization fraction (Xe)
600
800
1000
1200
1400
1600
1800
2000
z

```
import numpy as np
import matplotlib.pyplot as plt
#from ccdproc import wcs_project
import matplotlib
matplotlib.use('TKAgg')
import sys
import os
import io
import math
import scipy
import astropy.io.fits as fits
import astropy.units as u
import astropy.coordinates as coord
import pandas as pd
import matplotlib.lines as mlines
from astropy.cosmology import Planck13, z_at_value
from astropy.coordinates import SkyCoord
from astropy.coordinates import SkyCoord
from astropy.cosmology import FlatLambdaCDM
from astropy import constants as const
from scipy.integrate import quad, dblquad
from scipy.special import zeta

plt.rcParams["font.family"] = "serif"
plt.rcParams["mathtext.fontset"] = "dejavuserif"
path ='/Users/josephine/Desktop/Yale_2/COSM/'


#(1-Xe)/Xe^2 = 2 zeta(3)/pi^2 eta (2piT/me)^3/2 e^E_I/T


def calculate_left_side(z):
        return 1e-9 * 2 *zeta(3)/np.pi**2 * (2*np.pi *
0.23e-3*(1+z)/511e3)**(3/2) * np.exp(13.6/(0.23e-3*(1+z)))

def calc_Xe(C):
        return (-1 + np.sqrt(1+4*C))/(2*C)

def find_nearest(array, value):
        array = np.asarray(array)
        idx = (np.abs(array - value)).argmin()
        return idx

z_range = np.linspace(600,2000,1000)
Xe = np.zeros(len(z_range))

for i in range(len(z_range)):
        outcome = calculate_left_side(z_range[i])
        Xe[i] = calc_Xe(outcome)

index = find_nearest(Xe, 0.1)
z_01 = z_range[index]
index = find_nearest(Xe, 0.5)
z_05 = z_range[index]
```

```
print('z=', z_01)
print('z=', z_05)
plt.plot(z_range, Xe, color='black')
plt.axhline(0.1, ls='dotted', alpha=0.5)
plt.axvline(z_01, ls='dotted', alpha=0.5, label='Xe =0.1')
plt.axhline(0.5, ls='dotted', alpha=0.5, color='red')
plt.axvline(z_05, ls='dotted', alpha=0.5, label='Xe =0.5',
color='red')
plt.xlabel('z', fontsize=12)
plt.ylabel('Ionization fraction (Xe)', fontsize=12)
plt.legend(fontsize=12)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.savefig(path+'Ionization_fraction_redshift.pdf')
plt.show()



def calculate_left_side2(T):
        return 1e-9 * 2 *zeta(3)/np.pi**2 * (2*np.pi * T/
511e3)**(3/2) * np.exp(13.6/(T))

def calc_Xe2(C):
        return (-1 + np.sqrt(1+4*C))/(2*C)

#For kB T =0.01 the exponent blows up, I need to start at 0.1 ev.
#Trying out different ranges, I see that the answer converges to
~0.3 eV so that is why I made my array specific for 0.1-0.5.
kBT_range = np.linspace(0.1,0.5,1000) #(eV)
left_side = np.zeros(len(kBT_range))
Xes = np.zeros(len(kBT_range))
for i in range(len(kBT_range)):
        outcome = calculate_left_side2(kBT_range[i])
        Xes[i] = calc_Xe2(outcome)
        left_side[i] = (calc_Xe2(outcome) * kBT_range[i]**(3) *
(kBT_range[i]/0.23e-3)**(-3/2))

index = find_nearest(left_side, (1.9e-9))
Tdec = (kBT_range[index])
Xe_at_Tdec = Xes[index]
print('Tdec (eV)', Tdec)
print('Xe(Tdec)', Xe_at_Tdec)
```

$$\frac{1-X_e}{X_e^2} = \underbrace{\eta \, 2 \, \frac{\zeta(3)}{\pi^2} \left(\frac{2\pi T_0(1+z)}{m_e}\right)^{3/2} \exp\left(\frac{E_I}{T_0(1+z)}\right)}$$

Plugging this in python with

$T_0 = 0.23$ meV

$m_e = 510$ keV

$E_I = 13.6$ eV

$\eta = 10^{-9}$

call it C

$$C X_e^2 = 1 - X_e$$

$$C X_e^2 + X_e - 1 = 0$$

ABC formula $\quad a = C \quad b = 1 \quad c = -1$

$$X_e = \frac{-b \pm \sqrt{D}}{2a}$$

$$= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$= \frac{-1 \pm \sqrt{1^2 - 4 \cdot C \cdot -1}}{2C}$$

$$= \frac{-1 \pm \sqrt{1 + 4C}}{2C}$$

only + sign here means something (positive X_e)

In python gives plot (next page)

3) For $X_e = 0.1$ I find $z \simeq 1300$

for $X_e = 0.5$ I find $z \simeq 1425$

4) Age of the universe at this epoch (matter dominated)

$z = 1300$

$z = 1425$

$a \propto t^{2/3}$

$1+z \propto t^{-2/3}$

$(1+z)^{3/2} \propto t$

$t = t_0 (1+z)^{-3/2}$ $\quad t_0 \approx$ ~~[illegible]~~ $13.8\ Gyr$

At $z = 1300$

$t = 13.8\ Gyr\ (1+1300)^{-3/2} = 2.9 \cdot 10^{-4}\ Gyr$
$= 2.9 \cdot 10^{5}\ yr$

At $z = 1425$

$t = 13.8\ Gyr (1+1425)^{-3/2} = 2.6 \cdot 10^{5}\ yr$

The difference in time between $X_e = 0.5$ and $X_e = 0.1$ is only $\sim 30000\ yr$, very short on astronomical timescales. Recombination was a fast process.

5) Thompson scattering $\quad e^- + \gamma \leftrightarrow e^- + \gamma$

$\Gamma = n \sigma v$ $\quad v$ relative velocity (so $= c$)

$= n_e \sigma_T c$

$n_e = X_e n_b$ $\quad n_b = \eta n_\gamma = \eta \frac{2\zeta(3)}{\pi^2} T^3$

or with constants:

$$n_b = \eta \frac{2\zeta(3)}{\pi^2 (\hbar c)^3} (k_B T)^3$$

$$\Gamma_{dec} = X_e(T_{dec})\, \eta \frac{2\zeta(3)}{\pi^2 (\hbar c)^3} (k_B T_{dec})^3 c\, \sigma_T$$

$$\frac{H^2}{H_0^2} = \frac{\Omega_{\gamma,0}}{a^4} + \frac{\Omega_{m,0}}{a^3} + \frac{\Omega_{k,0}}{a^2} + \Omega_{\Lambda,0}$$

Assume only matter

$$\frac{H^2}{H_0^2} = \frac{\Omega_{m,0}}{a^3} \qquad H = H_0\sqrt{\Omega_{m,0}}\, a^{-3/2}$$

$$\frac{a}{a_0}$$

$$a \propto \frac{1}{T}$$

$$H = H_0\sqrt{\Omega_{m,0}}\left(\frac{T}{T_0}\right)^{3/2}$$

$$H(T_{dec}) = H_0\sqrt{\Omega_{m,0}}\left(\frac{T_{dec}}{T_0}\right)^{3/2}$$

$\Gamma = H$ gives T_{dec}

$$X_e(T_{dec})\,\eta\,\frac{2\zeta(3)}{\pi^2(\hbar c)^3}(k_B T_{dec})^3\, c\,\sigma_T = H_0\sqrt{\Omega_{m,0}}\left(\frac{T_{dec}}{T_0}\right)^{3/2}$$

$$X_e(T_{dec})\,(k_B T_{dec})^3\left(\frac{k_B T_{dec}}{k_B T_0}\right)^{-3/2} = \underbrace{\frac{H_0\sqrt{\Omega_{m,0}}\,\pi^2(\hbar c)^3}{\eta\, 2\zeta(3)\, c\,\sigma_T}}_{=\text{number}}$$

number units: $\dfrac{s^{-1}\cdot(eV\, s\cdot m\, s^{-1})^3}{m\, s^{-1}\; m^2} = eV^3$

that is exactly what I want, so we for $k_B T_{dec}$ in eV

$$\text{number} = \frac{2.2\cdot 10^{-10}\, s^{-1}\sqrt{0.3}\,\pi^2 \left(\frac{4.136\cdot 10^{-15}}{2\pi}\, eVs \cdot 2.998\cdot 10^{8}\, ms^{-1}\right)^3}{10^{-9}\, \zeta(3)\cdot 2\cdot 2.998\cdot 10^{8}\, ms^{-1}\cdot 6.65\cdot 10^{-29}\, m^2}$$

$$= 1.89\cdot 10^{-9}\ (eV)^3$$

So we have

$$\underbrace{X_e(T_{dec})}_{\text{This is an array in python for a } T_{dec} \text{ range}} (k_B T_{dec})^3 \left(\frac{k_B T_{dec}}{0.23\cdot 10^{-3}}\right)^{-3/2} = 1.89\cdot 10^{-9}\ eV^3$$

Find for which $k_B T_{dec}$ this equation holds

I do this with python and I find

$k_B T_{dec} \simeq 0.26$ eV

$$a \propto \frac{1}{T}$$

$$1+z \propto \frac{1}{T}$$

$$1+z = \frac{T_{dec}}{T_0}$$

$$1+z = \frac{0.26\ eV}{0.23\cdot 10^{-3}\ eV} \simeq 1100$$ (as expected ~CMB!)

Also this justified our choice for assuming matter dominated, it is $\ll z_{mr}$ equality

Het Baarnsch Lyceum

Naam:

Vak:

Datum: Klas:

Age of universe

$$t = t_0 (1+z)^{-3/2}$$

$$= 13.8 \text{ Gyr} (1 + 1100)^{-3/2}$$

$$= \cancel{380} \cdot 10^5 \text{ yr}$$

$$3.77$$

X_e at this z, T_{dec} with python (or by hand works too)

$$X_e(T_{dec}) = 0.004$$

(see code next page)

What-if BBN

$$X_n = \frac{n_n}{n_n + n_p}$$

neutrons/protons coupled through

$$n + \nu_e \leftrightarrow p + e^-$$
$$n + e^+ \leftrightarrow p + \bar{\nu}_e$$

Again we have

$$n_{,eq} = g\left(\frac{m k_B T}{2\pi\hbar^2}\right)^{3/2} \exp\left(-\frac{(mc^2 - \mu)}{k_B T}\right)$$

$$\frac{n_n}{n_p} = \frac{g_n}{g_p}\left(\frac{m_n}{m_p}\right)^{3/2} \exp\left(\frac{-(m_n + m_p)c^2}{k_B T}\right)$$

$\mu_n \approx \mu_p$ because μ_ν, μ_e very small

$$Q = (m_n - m_p)c^2 = 1.3 \text{ MeV}$$

$g_n = g_p = 2 \qquad m_n \approx m_p \qquad$ so

$$\frac{n_n}{n_p} = \exp\left(-Q/k_B T\right)$$

(neglect k_B)

$$= \exp(-Q/T)$$

$$X_n = \frac{n_n}{n_n + n_p} = \frac{1}{1 + \frac{n_p}{n_n}} = \frac{1}{1 + \exp(Q/T)}$$

$$= \frac{\exp(-Q/T)}{\exp(-Q/T) + 1}$$

so we have our wanted expression

Naam:

Vak:

Datum:

Klas:

Cijfer:

$$\cancel{X_e(T_{dec})\left(k_B T_{dec}\right)^{3/2}} = \cancel{\frac{\pi^2 (\hbar c)^3 \sqrt{\Omega_{m,0}} H_0}{2\zeta(3)\eta\,\sigma_T c\,(k_B T_0)^{3/2}}}$$

- neutrinos decouple $\Gamma < H$
  so the neutron-proton ratio freezes out
  $X_n = \frac{n_n}{n_n + n_p}$ the fraction ~~of~~ neutrons of both that are
  at $T = 0.8$ MeV $Q = 1.3$ MeV
  $X_n = \frac{e^{-1.3/0.8}}{1+e^{-1.3/0.8}}$
  $= 0.165$ (16 % of nucleons are neutrons)

- $2n + 2p \rightarrow 1\ He$
  If we convert all neutrons to Helium nuclei, we have $n_{He} = \frac{1}{2} n_n$ because we need 2 neutrons for 1 He atom

$Y_p = 4\frac{n_{He}}{n_H} = 2\frac{n_n}{n_H}$

$n_H = n_n + n_p$

$Y_p = \frac{2n_n}{n_n+n_p} = 2X_n = 0.33$

(Actually some ~~of~~ of the neutrons decayed before they could form He)

- Mass difference $Q = 2.6$ MeV

$$X_n = \frac{e^{-2.6/0.8}}{1 + e^{-2.6/0.8}} = 0.04$$ much less neutrons!

Then

$$y = 2X_n = 0.08$$ (still not taking into account decay)

So it is "lucky" that Q is smaller, otherwise there would not be that much Helium or other higher elements.

Freeze in DM

First I derive the expression for freeze-out:

$$Y = \frac{n}{s}$$

$$\frac{1}{a^3}\frac{d}{dt}\left(na^3\right) = -\langle\sigma v\rangle\left[n^2 - n_{eq}^2\right]$$

$$\frac{1}{a^3}\frac{d}{dt}\left(sYa^3\right) = -\langle\sigma v\rangle s^2\left[Y^2 - Y_{eq}^2\right]$$

$$= \frac{1}{a^3} Y \frac{d}{dt}\left(sa^3\right) + \frac{1}{a^3} sa^3 \frac{d}{dt}(Y) \quad \text{(Chain rule)}$$

↑ this is 0 sa^3 is conserved

$$= s\frac{d}{dt}Y$$

So we find $\frac{dY}{dt} = -\langle\sigma v\rangle s\left[Y^2 - Y_{eq}^2\right]$

$$\frac{dY}{dt} = \frac{dY}{dx}\frac{dx}{dt} \qquad x = \frac{m}{T} \qquad T \propto \frac{1}{a}$$

$$\frac{dx}{dt} = \frac{dx}{dT}\frac{dT}{dt} = \frac{dx}{dT}\frac{dT}{da}\frac{da}{dt} = -\frac{m}{T^2}\cdot -\frac{1}{a^2}\,\dot{a}$$

$$= \frac{m}{T^2}\frac{\dot{a}}{a}T = H\frac{m}{T} = Hx$$

$$s = \frac{2\pi^2}{45} g_{*,s} T^3$$

$$\frac{dY}{dx} = \frac{dY}{dt}\cdot\frac{dt}{dx} = \frac{-\langle\sigma v\rangle s\left[Y^2 - Y_{eq}^2\right]}{Hx}$$

$$= -\langle\sigma v\rangle \frac{2\pi^2}{45} g_{*,s} T^3 \frac{1}{Hx}\left[Y^2 - Y_{eq}^2\right]$$

$$T^3 = \frac{m^3}{x^3} \rightarrow$$

$$\frac{dY}{dx} = -\langle\sigma v\rangle \frac{2\pi^2}{45} g_{*,s} \frac{m^3}{x^4 H}\left[Y^2 - Y_{eq}^2\right]$$

$\lambda = \langle\sigma v\rangle \frac{2\pi^2}{45} g_{*,s} \frac{m^3}{H}$

Because $H \propto x^{-2}$, we want λ to be a constant, we redefine $H = H/x^2$ then $\frac{dY}{dt} = \frac{-\lambda}{x^2}\left[Y^2 - Y_e\right.$

$$\left[H^2 \propto \rho_r \propto T^4 \propto x^{-4} \quad \text{so } H \propto x^{-2}\right]$$

- now same method for freeze-in:

$$\frac{1}{a^3}\frac{d}{dt}(na^3) = 2\Gamma h(t)\, n_{\sigma,eq}(t)$$

$$y = \frac{n}{s} \qquad y_{eq} = \frac{n_{eq}}{s}$$

$$\frac{1}{a^3}\frac{d}{dt}(s y a^3) = 2\Gamma h(t)\, s\, y_{eq}(t)$$

$$\frac{1}{a^3} s a^3 \frac{dy}{dt} = 2\Gamma h(t)\, s\, y_{eq}(t)$$

$$\frac{dy}{dt} = 2\Gamma h(t)\, y_{eq}(t)$$

$$\frac{dy}{dt} = \frac{dy}{dx}\frac{dx}{dt} \qquad x = \frac{m_\sigma}{T} \qquad T \propto \frac{1}{a}$$

$$\frac{dx}{dt} = \frac{dx}{dT}\frac{dT}{dt} = \frac{dx}{dT}\frac{dT}{da}\frac{da}{dt} = -\frac{m_\sigma}{T^2}\cdot\frac{-1}{a^2}\dot{a} = \frac{m_\sigma}{T}H$$

$$= x \cdot H$$

$$\frac{dy}{dx} = \frac{dy}{dt}\frac{dt}{dx} = 2\Gamma h(t)\, y_{eq}(t)\frac{1}{xH}$$

$$= \underbrace{\frac{2\Gamma}{x^2 H}}_{=\lambda_1} x\, h(x)\, y_{eq}(x)$$

Again $H \propto x^{-2}$ so λ_1 is a constant

$$\lambda_1 = \frac{2\Gamma}{x^2 H} \qquad \frac{dy}{dx} = \lambda_1\, x\, h(x)\, y_{eq}(x)$$

- plot $y(x)$ for $\lambda_1 = 10^{-6}, 10^{-8}, 10^{-10}$

$$\frac{dy}{dx} = \lambda_1 \, x \, h(x) \, y_{eq}(x)$$

Assume $y_{eq} = $ const.

~~$y_{eq}(x) \propto e^{-x}$~~ $n_0 \propto e^{-m_0/T} = e^{-x}$

So $y_{eq} \propto e^{-x}$

$$n = g \left(\frac{mT}{2\pi}\right)^{3/2} e^{-m/T}$$ non-rel limit equilibrium number density

$y = \frac{n}{s}$ $\quad s = \frac{2\pi^2}{45} g_{*,s} T^3$

At $m=T$ $\quad n/s = \underbrace{\frac{g}{g_{*,s}} \frac{(2\pi)^{-3/2}}{2\pi^2 \, 45}}_{\text{const } C} e^{-m/T} = y_{eq}$

$g = 2$
$g_{*,s} \approx 10$??

$$y_{eq} = C \cdot e^{-x}$$

$$\frac{dy}{dx} = \lambda_1 \cdot C \cdot x \frac{x}{(x+2)} e^{-x} = \lambda_1 C \frac{x^2}{x+2} e^{-x}$$

$$\int_{y(x_0)}^{y(x)} dy = \int_{x_0}^{x} \lambda_1 C \frac{x^2}{x+2} e^{-x} \, dx$$

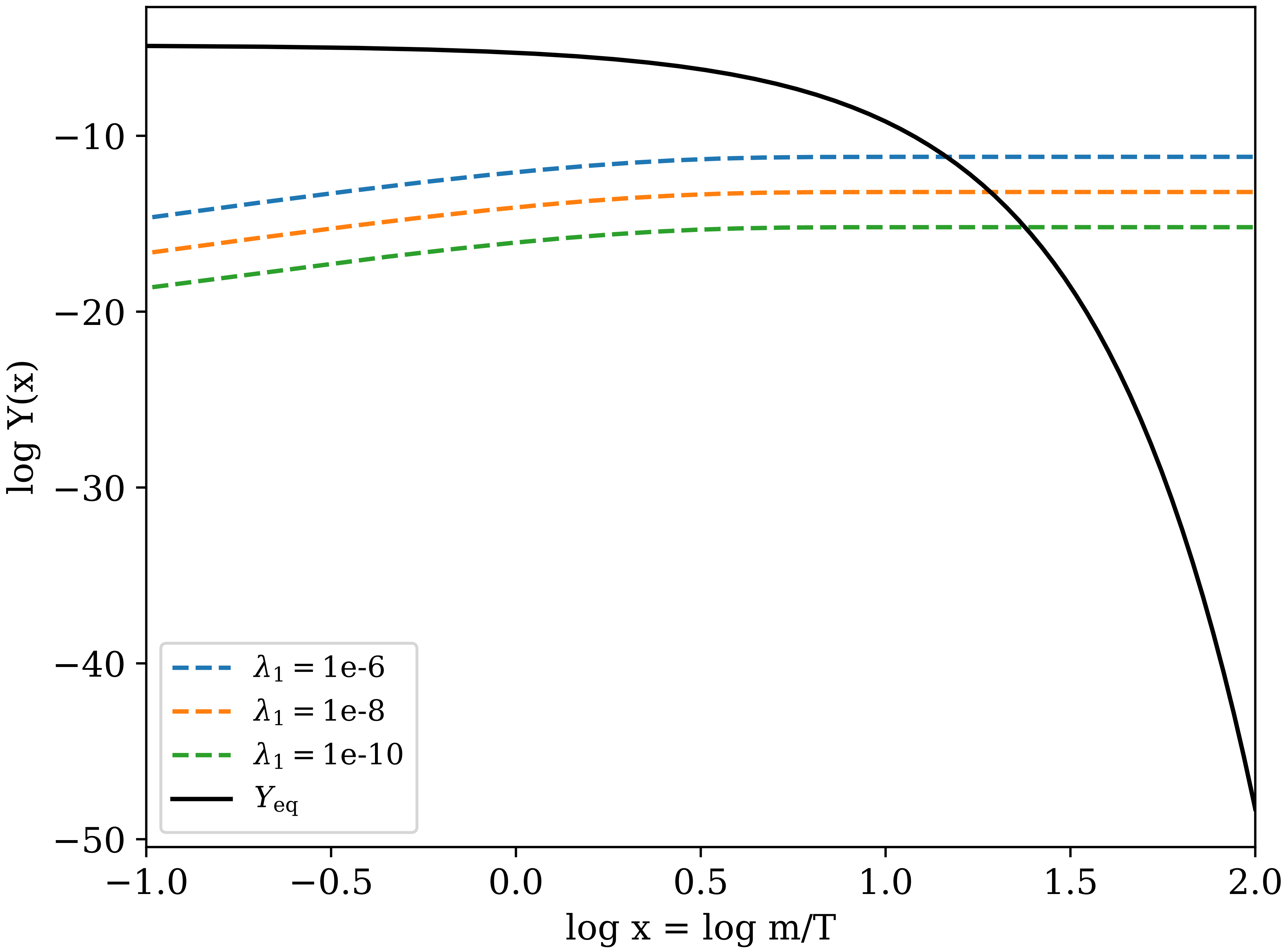
$x_0 = 0.01$ to $x = 100$ only plot $[0.1, 100]$

$y(x_0) = 10^{-20}$

$y(x) - 10^{-20} = $ integral $\rightarrow$ python

```
import numpy as np
import matplotlib.pyplot as plt
#from ccdproc import wcs_project
import matplotlib
matplotlib.use('TKAgg')
import sys
import os
import io
import math
import scipy
import astropy.io.fits as fits
import astropy.units as u
import astropy.coordinates as coord
import pandas as pd
import matplotlib.lines as mlines
from astropy.cosmology import Planck13, z_at_value
from astropy.coordinates import SkyCoord
from astropy.coordinates import SkyCoord
from astropy.cosmology import FlatLambdaCDM
from astropy import constants as const
from scipy.integrate import quad, dblquad
from scipy.special import zeta

plt.rcParams["font.family"] = "serif"
plt.rcParams["mathtext.fontset"] = "dejavuserif"
path ='/Users/josephine/Desktop/Yale_2/COSM/'



x0 = 0.01
x= np.linspace(0.01,100, 10000)
C=2/10* (2*np.pi)**(-3/2)/(2*np.pi**2*45)
#I am not quite sure about g=2 and g*,s = 10
print(C)

def right_side(x, lambda1):

        lower_limit = x0
        upper_limit = x
        outcome = np.nan*np.ones(len(x))

        for i in range(len(x)):
                term1, _ =      quad(_integrand,
lower_limit,upper_limit[i])
                outcome[i] = term1
        return outcome*lambda1*C


def _integrand(x):
        return x**2 *np.exp(-x)/(x+2)

lambdas =[1e-6, 1e-8, 1e-10]
lambdas_=['1e-6', '1e-8', '1e-10']
for i in range(len(lambdas)):
```

```
        lambda1 = lambdas[i]
        integral_x = right_side(x, lambda1)
        Y_x =  integral_x +1e-20
        plt.plot(np.log10(x), np.log10(Y_x), label=r'$\lambda_1 =$'
+lambdas_[i], ls='dashed')
plt.plot(np.log10(x), np.log10(C*np.exp(-x)),
label=r'$Y_{\mathrm{eq}}$', color='black')
plt.xlim(-1, 2)
#plt.yscale('log')
#plt.xscale('log')
plt.xlabel('log x = log m/T',fontsize=12)
plt.ylabel('log Y(x)',fontsize=12)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.legend()
plt.tight_layout()
plt.savefig('Yeq_lambda1.pdf')
plt.show()
```



log Y(x)
−10
−20
−30
−40
−50
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
log x = log m/T
λ_1 = 1e-6
λ_1 = 1e-8
λ_1 = 1e-10
Y_{eq}