ASTR 600 - Cosmology

HW 4

Iver Warburton

October 2023

In [1]:
```
# Our lovely imports

import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import quad
from scipy.special import zeta
```

Problem 1

part ii.)

In [2]:
```
# Constants

# Masses
m_p = 938.272 # MeV/c^2
m_n = 940.6 # MeV/c^2
m_e = 0.511 # MeV/c^2
m_phot = 0
m_H = 938.8  #939.0 # MeV/c^2

# Number densities
#n_p =
#n_n =
#n_e =
#n_phot =

#n_H = n_p - n_n
#n_He = (1/2)*n_n
#n_B = n_p + n_H

# Defined
E_I = 0.0000136 # MeV, so 13.6 eV  #m_p + m_e - m_H
print(E_I)
eta = 6 * 10**-10  # ___  #n_B/n_gamma
```

```
1.36e-05
```

In [3]:
```
z = np.linspace(1000, 2000, 1000)

T_o = 2.3525 * 10**-10  #MeV  #2.73 K  ###0.8 MeV
T = T_o * (1 + z)
```

In [4]:
```
# Build the solution function
def build_Xe_solution(T, eta, E_I, m_e):
    f = ((2*zeta(3))/(np.pi**2)) * eta * (((2*np.pi*T)/m_e)**(3/2)) *
    X_e = (-1 + np.sqrt(1 + (4*f))) / (2*f)
    return X_e
```

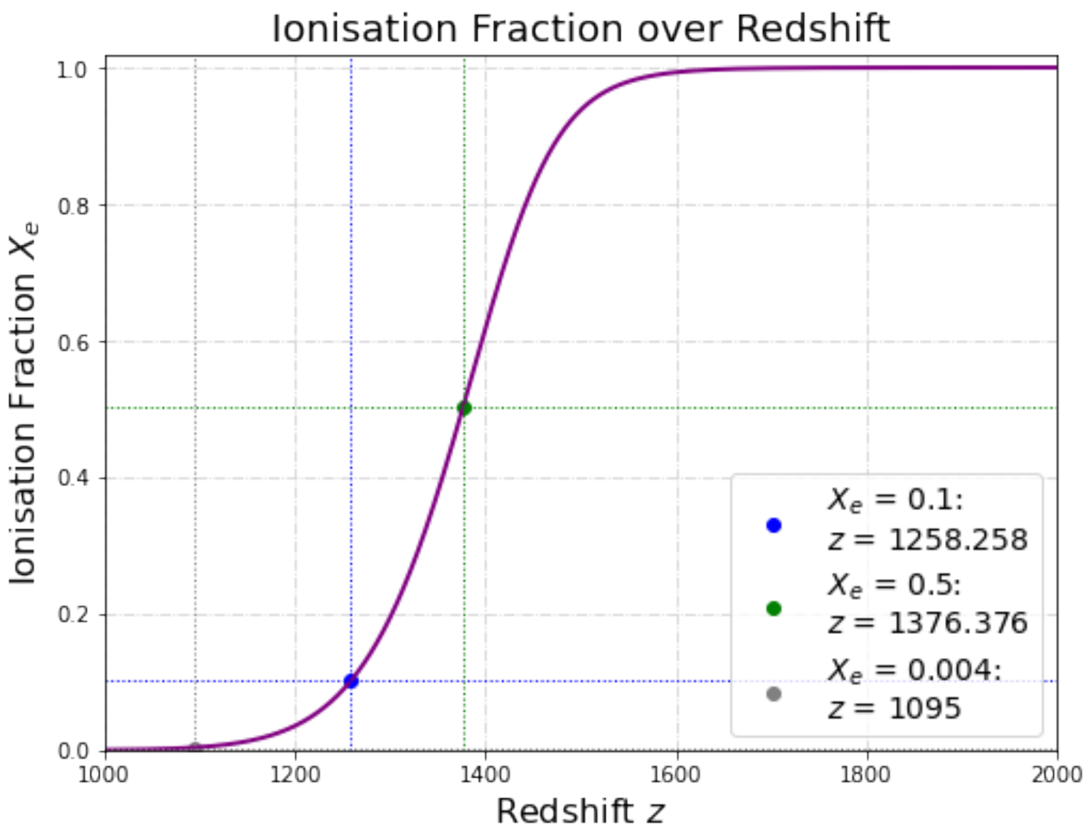
In [5]:
```
# Plot results
lef Xe_solution_plot(z, X_e):

   ## PLOT ##
   # Set figure
   fig, ax = plt.subplots(figsize=(8,6))

   # Plot
   plt.plot(z, X_e, ls='-', color='purple', lw=2)


   ## POINTS ##
   ## For X_e = 0.1  ##
   # Add point
   index_Xe01 = np.where(X_e >= 0.1)[0][0]
   Xe01 = X_e[index_Xe01]
   z_Xe01 = z[index_Xe01]
   print("z at X_e = 0.1:", z_Xe01.round(3))
   plt.scatter(z[index_Xe01], X_e[index_Xe01], color='blue', label='$X

   # Add pointer lines
   plt.axhline(Xe01, ls=':', c='blue', lw=1)
   plt.axvline(z_Xe01, ls=':', c='blue', lw=1)


   ## For X_e = 0.5 ##
   # Add point
   index_Xe05 = np.where(X_e >= 0.5)[0][0]
   Xe05 = X_e[index_Xe05]
   z_Xe05 = z[index_Xe05]
   print("z at X_e = 0.5:", z_Xe05.round(3))
   plt.scatter(z[index_Xe05], X_e[index_Xe05], color='green', label='$

   # Add pointer lines
   plt.axhline(Xe05, ls=':', c='green', lw=1)
   plt.axvline(z_Xe05, ls=':', c='green', lw=1)


   ## Part 1e ##
   ## For z=1095 ##
   # Add point
```

```
index_z1095 = np.where(z >= 1095)[0][0]
z_1095 = z[index_z1095]
Xe_z1095 = X_e[index_z1095]
print("X_e at z = 1095:", Xe_z1095.round(3))
plt.scatter(z[index_z1095], X_e[index_z1095], color='grey', label='

# Add pointer lines
plt.axhline(Xe_z1095, ls=':', c='grey', lw=1)
plt.axvline(z_1095, ls=':', c='grey', lw=1)
## End of part 1e addition ##



## LABELLING ##
# Label axes
plt.xlabel('Redshift $z$', fontsize=16)
plt.ylabel('Ionisation Fraction $X_e$', fontsize=16)

# Increse axis numbering text size
ax.tick_params(axis='both', which='major', labelsize=10)
ax.tick_params(axis='both', which='minor', labelsize=10)

# Plot features
plt.xlim(1000, 2000)
plt.ylim(0, 1.02)
plt.legend(fontsize=14)
plt.grid(True, color='lightgrey', ls='-.')
plt.title('Ionisation Fraction over Redshift', fontsize=18)


## SAVE ##
# Save and show
plt.savefig("HW4Q1ePlot.pdf", format="pdf", bbox_inches="tight", ov
plt.show()
```

In [6]:
```
Xe_solution_plot(z, build_Xe_solution(T, eta, E_I, m_e))
```

```
z at X_e = 0.1: 1258.258
z at X_e = 0.5: 1376.376
X_e at z = 1095: 0.004
```

```
/var/folders/1w/ktxtfrr91bj5bztz50dqm0fr0000gn/T/ipykernel_12415/3203
391629.py:74: MatplotlibDeprecationWarning: savefig() got unexpected
keyword argument "overwrite" which is no longer supported as of 3.3 a
nd will become an error in 3.6
  plt.savefig("HW4Q1ePlot.pdf", format="pdf", bbox_inches="tight", ov
erwrite=True)
```




```
build_Xe_solution(T, eta, E_I, m_e)
```

In [7]:
```
zeta(3)
```

Out[7]: 1.2020569031595942

In [ ]:

In [ ]:

In [ ]:

Problem 3

part ii.)

Numerical integration

In [8]:
```
# Define function to build integrand
def integrand(x):
    # Integrand expression for comoving distance that was provided in
    dY = ((x**2)/(x+2)) * (x/(2*np.pi))**(3/2) * np.exp(-x)
    return dY
```

```
lambdas = [10**-6, 10**-8, 10**-10]
x = np.linspace(0.01, 100, 1000)
Y_numint_list = {}

for lambda_val in lambdas:
    # Calculate the integral for comoving distance numerically
    Y_numint = lambda_val * quad(integrand, 0.1, 100)[0]
    #Y_numint = lambda_val * Y_numint
    Y_numint_list.append(Y_numint)
    print(np.shape(Y_numint))
```

In [30]:
```
lambda1 = 10**-6
lambda2 = 10**-8
lambda3 = 10**-10

x = np.linspace(0.01, 100, 1000)
#Y_numint_list = {}


# Calculate the integral numerically
Y1 = [lambda1 * quad(integrand, 0, x_val)[0] for x_val in x]
Y2 = [lambda2 * quad(integrand, 0, x_val)[0] for x_val in x]
Y3 = [lambda3 * quad(integrand, 0, x_val)[0] for x_val in x]

Yeq = (x/(2*np.pi))**(3/2) * np.exp(-x)

#Y_numint_list.append(Y_numint)
#print(np.shape(Y1))
#print(Y1)
```

In [43]:
```
# Plot results
def Y_plot(x, Y1, Y2, Y3):

    ## PLOT ##
    # Set figure
    fig, ax = plt.subplots(figsize=(8,6))

    # Plot
    plt.loglog(x, Y1, ls='-', color='green', lw=2, label='$\lambda = 1
    plt.loglog(x, Y2, ls='-', color='blue', lw=2, label='$\lambda = 10
    plt.loglog(x, Y3, ls='-', color='purple', lw=2, label='$\lambda = 
    plt.loglog(x, Yeq, ls='-', color='red', lw=2, label='$Y_{eq}$')
    #for Y in Y_list:
        #plt.plot(x, Y, ls='-', color='purple', lw=2)


    ## LABELLING ##
    # Label axes
    plt.xlabel('Time $x = \dfrac{m}{T}$', fontsize=16)
    plt.ylabel('$Y = \dfrac{n}{s} \sim \dfrac{n}{T^3}$', fontsize=16)

    # Increse axis numbering text size
    ax.tick_params(axis='both', which='major', labelsize=12)
    ax.tick_params(axis='both', which='minor', labelsize=12)

    # Plot features
    plt.xlim(0.1, 100)
    #plt.ylim(0, 1.02)
    plt.legend(fontsize=14)
    plt.grid(True, color='lightgrey', ls='-.')
    plt.title('Y for Freeze-In', fontsize=18)


    ## SAVE ##
    # Save and show
    plt.savefig("HW4Q3Plot.pdf", format="pdf", bbox_inches="tight", ov
    plt.show()
```

In [44]:
```
Y_plot(x, Y1, Y2, Y3)
```

```
/var/folders/1w/ktxtfrr91bj5bztz50dqm0fr0000gn/T/ipykernel_12415/6295
57688.py:36: MatplotlibDeprecationWarning: savefig() got unexpected k
eyword argument "overwrite" which is no longer supported as of 3.3 an
d will become an error in 3.6
  plt.savefig("HW4Q3Plot.pdf", format="pdf", bbox_inches="tight", ove
rwrite=True)
```