

```

#generic import and constant definition list
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as colors
import h5py
import astropy.constants as cons
from matplotlib.colors import LogNorm
import astropy.units as u
import pandas as pd
import scipy.optimize as opt
#all of the important fundamental constants are put into cgs units
#just for convenience
c=cons.c.cgs.value
G=cons.G.cgs.value
h=cons.h.cgs.value
hbar=cons.hbar.cgs.value
Msun=cons.M_sun.cgs.value
Rsun=cons.R_sun.cgs.value
Rearth=cons.R_earth.cgs.value
mp=cons.m_p.cgs.value
me=cons.m_e.cgs.value
mn=cons.m_n.cgs.value
kB=cons.k_B.cgs.value
mu_e=2 #mean mass per electron for He-core or C/O core composition
m_u = 1/cons.N_A.cgs.value #atomic mass unit in grams
from astropy.io import fits

def ionization_frac(T, eta):
    s_val = s(T,eta)
    return (np.sqrt(1+4*s_val)-1)/(2*s_val)
def s(T,eta):
    return eta*3.84*((kB*T/(me*c**2))**(3/2))*np.exp(13.6*1.602e-
12/(kB*T))

```

$$T(t_e) = T(t_0)(1+z) = 2.73*(1+z)$$

```

fig = plt.figure()
ax1 = fig.add_subplot(111)
ax2 = ax1.twiny()
t = np.linspace(2000,5000,3000)
model1 = ionization_frac(t, 1e-9)
model2 = ionization_frac(t, 6.5e-10)
model3 = ionization_frac(t, 5.8e-10)
model4 = ionization_frac(t, 1e-10)
ax1.plot(t, model1, label='$\eta=1e-9$', c='blue')
ax1.plot(t, model2, label='$\eta=6.5e-10$', c='green')
ax1.plot(t, model3, label='$\eta=5.8e-10$', c='red')
ax1.plot(t, model4, label='$\eta=1e-10$', c='magenta')
ax1.axvline(3056, linestyle='--', color='black', label='Decoupling')

```

```
ax2.plot(t/2.73 - 1, model1, alpha=0)
ax1.set_xlabel('Temperature (K)')
ax2.set_xlabel('Redshift (z)')
ax1.set_xlim(2000,5000)
ax1.set_ylabel('Ionization Fraction')
ax1.legend()
plt.show()
```

