

Calculating and Plotting the Dulong-Petit, Einstein and Debye formulas for Specific Heat of Solids

```

import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit

R = 8.314
T = np.linspace(1,300,100)
T_D = 343 # Debye Temperature of Cu in Kelvin
T_E = 0.806*T_D # Relationship between Einstein Temp T_E and Debye Temp T_D (Wikipedia)
#print (T)

def c_DP(T):
    x = T/T
    return 3 *R *x

def c_einstein(T):
    x = T_E / T
    return 3 * R * x**2 * np.exp(x) / (np.exp(x) - 1)**2

#def integrand(y):
#    return y**4 * np.exp(y) / (np.exp(y) - 1)**2

def integrand(dummy,y):
    if y > 20:
        return 0
    return y**4 * np.exp(y) / (np.exp(y) - 1)**2

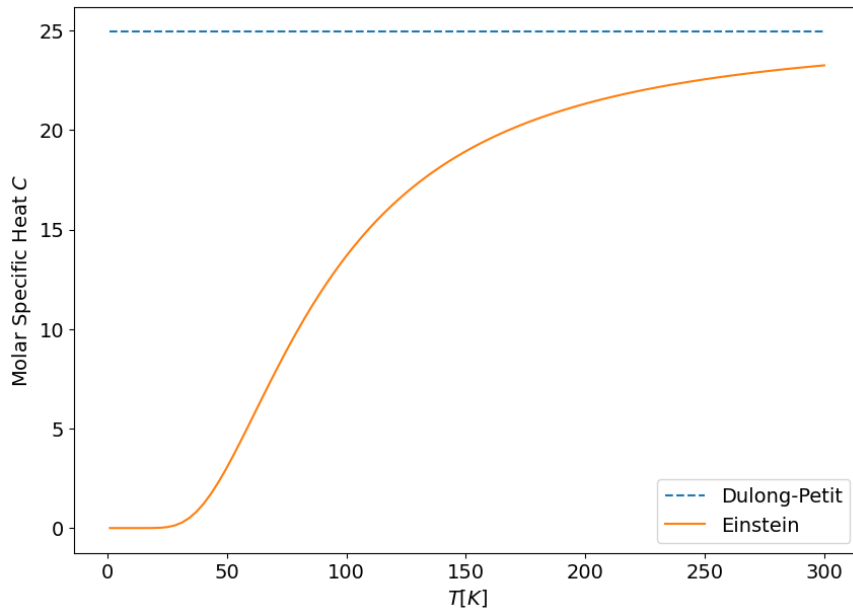
def c_debye(T):
    x = T / T_D
    ans = 9 * R * x**3
    for k in range(len(x)):
        igral = odeint(integrand, 0, [1e-10,1/x[k]])
        ans[k] *= igral[1,0]
    return ans

plt.close()
plt.rcParams.update({'font.size': 14})
plt.rcParams["figure.figsize"] = [10,7]

C_DP = c_DP(T)
C_ES = c_einstein(T)
plt.plot(T, C_DP, label='Dulong-Petit', ls = 'dashed')
plt.plot(T, C_ES, label='Einstein', ls = 'solid')
plt.xlabel('$T$ [K]$')
plt.ylabel('Molar Specific Heat $C$')
plt.legend()
plt.savefig('DP_Einstein.jpg')
plt.show()
plt.close()

```

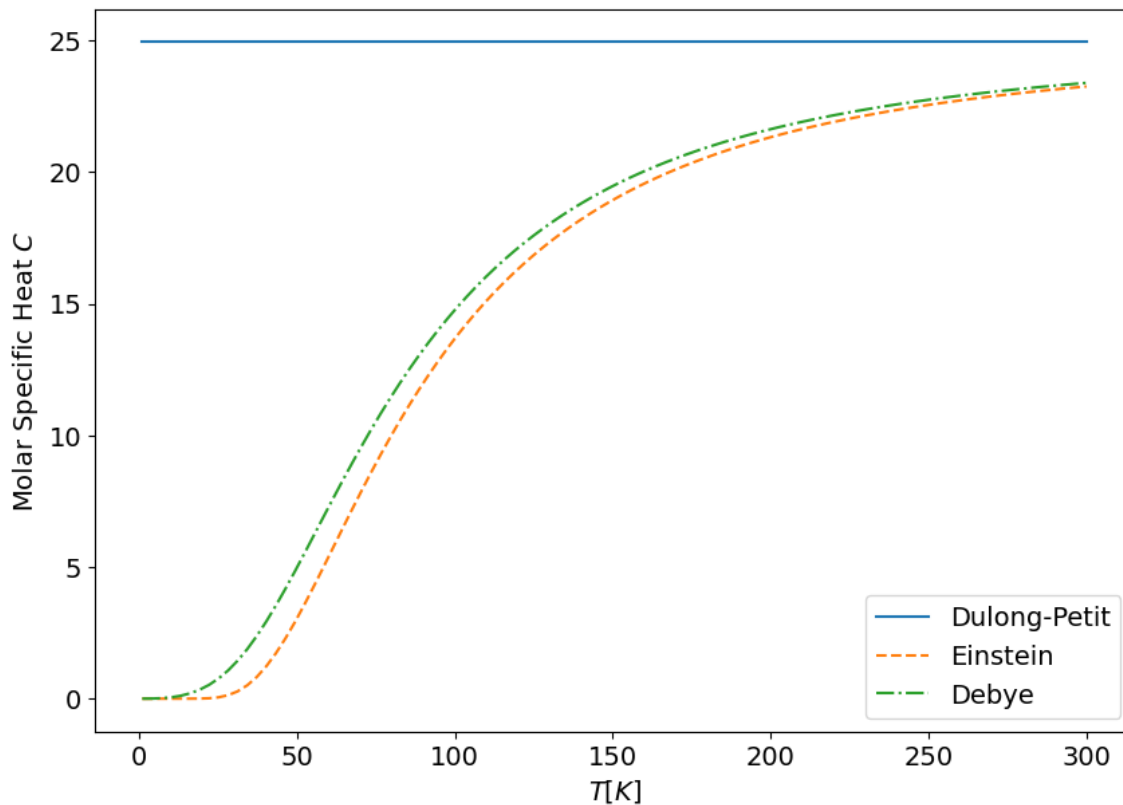
↪



```

N = T.shape[0]
C_DB = c_debye(T)
pl.close()
pl.rcParams.update({'font.size': 14})
pl.rcParams["figure.figsize"] = [10,7]
#C_DB = c_debye(T([0]))
#C = f_integ(np.array([1]))
pl.plot(T, C_DP, label='Dulong-Petit', ls = '-')
pl.plot(T, C_ES, label='Einstein', ls = '--')
pl.plot(T, C_DB, label='Debye', ls = '-.')
pl.xlabel('$T$ [K]$')
pl.ylabel('Molar Specific Heat $C$')
pl.legend()
pl.savefig('DP_Einstein_Debye.pdf')
pl.show()
pl.close()

```



✓ 1s completed at 3:40 PM

