## Plots of Wien Vs. Planck Distribution Law at different cavity temperatures

```
import numpy as np
import matplotlib.pyplot as pl
```

## Defining the constants

```
#h = 4.1357 * 10**(-15)     #Planck Constant in units of eV.s
h = 6.6261 * 10**(-34)     #Planck Constant in units of J.s
c = 2.9979 * 10**(8)       #Speed of light in vacuum in units of m.s^(-1)
#k = 8.6173 * 10**(-5)      #Boltzmann constant in units of eV.K^(-1)
k = 1.3806 * 10**(-23)     #Boltzmann constant in units of J.K^(-1)
b = 2.898 * 10**(-3)       #Wien constant in units of m.K
#T = 5000                   # Temperature of the Blackbody in K
#lmbda_max = b/T            #Wien's Displacement Law
#print ('lambda max in m =', lmbda_max)
```

## Defining the Distribution Functions

```
def Wien(T, lmbda):
    x = h*c/(lmbda*k*T)
    return (2 * h * c**2 / (lmbda**5))* np.exp (-x)

def Planck(T, lmbda):
    x = h*c/(lmbda*k*T)
    return (2 * h *c**2 /(lmbda**5)) * (1/(np.exp(x)-1))
```

## Defining the wavelength scales and the temperature values

```
lmbda = np.arange(0.1,25.0,0.001)   #defining the array of wavelength
lmbda *= 10**(-7)         # Wavelength lambda in micrometers
T = [3000, 4000, 5000, 6000, 7000]    # Range of temperatures in Kelvin
#T_Label = [ 'T = 300 K', 'T = 400 K', 'T = 500 K', 'T = 600 K', 'T = 700 K']
#T_Col = ['red', 'green', 'blue', 'cyan', 'magenta']
lmbda_max = b/T[2]
print ('lambda_max for median temp =', lmbda_max, 'Temperature =', T[2],'K')
```

```
lambda_max for median temp = 5.796e-07 Temperature = 5000 K
```

## Calculating the Distributions Functions as functions of wavelength at various temperatures

```
# Calulating the Distribution Functions
B_Wn = np.zeros([len(T), len(lmbda)])
B_Plnck = np.zeros([len(T), len(lmbda)])
#B_Wn_max = np.zeros([len(T)])
#B_Plnck_max = np.zeros([len(T)])
for i in range(len(T)):
    B_Wn[i,:] = Wien(T[i], lmbda)
    B_Plnck[i,:] = Planck(T[i], lmbda)
```

## Plotting the Distributions Functions as functions of wavelength at various temperatures

```
#ax.set_facecolor(m)
pl.close()
pl.rcParams["figure.figsize"]=[10,7]
pl.subplot(2,2,1)
pl.plot(lmbda*(10**7), B_Wn[0,:], label= 'Wien', color = 'red', ls = 'dashed') # Plotting the absolute B_W
pl.plot(lmbda*(10**7), B_Plnck[0,:], label= 'Planck', color = 'blue') # Plotting the absolute B Wien vs La
```

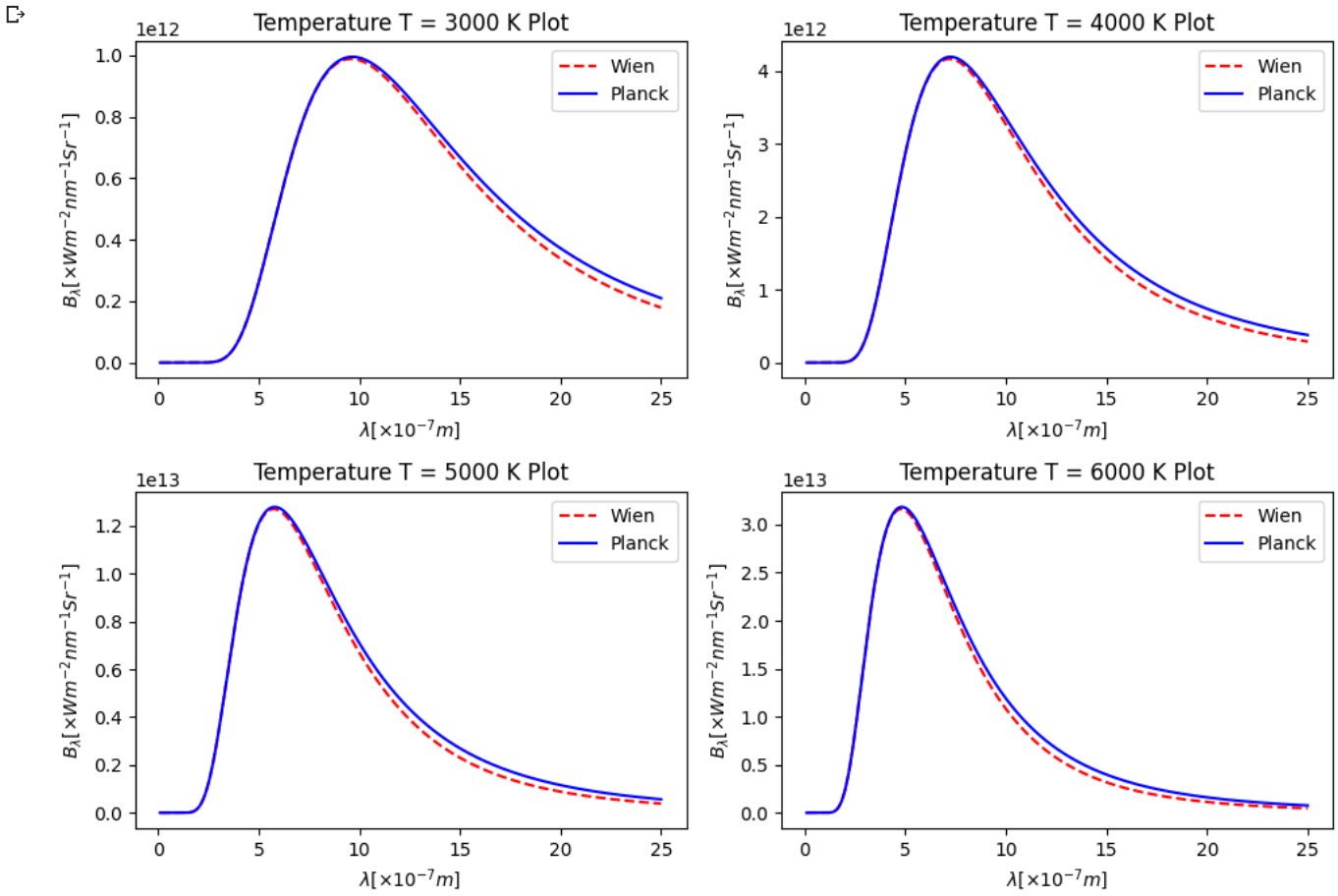
```
pl.xlabel(r'$\lambda [\times 10^{-7} m]$')
pl.ylabel(r'$B_\lambda [\times W m^{-2} nm^{-1} Sr^{-1}]$')
pl.title('Temperature T = 3000 K Plot')
pl.close('Temperature T = 3000 K Plot')
pl.legend()

pl.subplot(2,2,2)
pl.plot(lmbda*(10**7), B_Wn[1,:], label= 'Wien', color = 'red', ls = 'dashed') # Plotting the absolute B_W
pl.plot(lmbda*(10**7), B_Plnck[1,:], label= 'Planck', color = 'blue') # Plotting the absolute B_Wien vs La
pl.xlabel(r'$\lambda [\times 10^{-7} m]$')
pl.ylabel(r'$B_\lambda [\times W m^{-2} nm^{-1} Sr^{-1}]$')
pl.title('Temperature T = 4000 K Plot')
pl.close('Temperature T = 4000 K Plot')
pl.legend()

pl.subplot(2,2,3)
pl.plot(lmbda*(10**7), B_Wn[2,:], label= 'Wien', color = 'red', ls = 'dashed') # Plotting the absolute B_W
pl.plot(lmbda*(10**7), B_Plnck[2,:], label= 'Planck', color = 'blue') # Plotting the absolute B_Wien vs La
pl.xlabel(r'$\lambda [\times 10^{-7} m]$')
pl.ylabel(r'$B_\lambda [\times W m^{-2} nm^{-1} Sr^{-1}]$')
pl.title('Temperature T = 5000 K Plot')
pl.close('Temperature T = 5000 K Plot')
pl.legend()

pl.subplot(2,2,4)
pl.plot(lmbda*(10**7), B_Wn[3,:], label= 'Wien', color = 'red', ls = 'dashed') # Plotting the absolute B_W
pl.plot(lmbda*(10**7), B_Plnck[3,:], label= 'Planck', color = 'blue') # Plotting the absolute B_Wien vs La
pl.xlabel(r'$\lambda [\times 10^{-7} m]$')
pl.ylabel(r'$B_\lambda [\times W m^{-2} nm^{-1} Sr^{-1}]$')
pl.title('Temperature T = 6000 K Plot')
pl.close('Temperature T = 6000 K Plot')
pl.legend()

pl.tight_layout()
pl.show()
```



✓ 2s completed at 4:37 PM