TFY4230
Exercise 5

Vegard Stenhjem Hagen

October 29, 2012

Problem 9

Simulation

Partition Function:

$$Z = \sum_{n=0}^{2} \exp(-\beta E_n)$$

```python
from __future__ import division
import sys
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
from scipy.misc import derivative as ddx
from scipy.integrate import quad

kB = 1;

def partfunc(T,E=[]):
    #Partition Function
    return np.sum(np.exp(-E/(kB*T)))
```

Heat Capacity:

$$\begin{aligned} C &= \frac{\mathrm{d} < E >}{\mathrm{d}T} \\ < E > &= \frac{1}{Z}\sum_{j=0}^{\infty} \exp(-\beta E_j) * E_j \end{aligned}$$

```python
def meanE(T,E = []):
    #Mean Energy
    return np.sum(np.exp(-E/(kB*T))*E)/partfunc(T,E)

def heatcap(T=[],E=[]):
    #Heat Capacity
    C = np.zeros(len(T))
    for i in xrange(len(T)):
        C[i] = meanE(T[i],E)
    return np.gradient(C,T[1]-T[0])
```

Entropy:

$$S = k_B \ln Z - \frac{< E >}{T}$$

```python
def entropy(T=[],E=[]):
    #Entropy
    S = np.zeros(len(T))
    for i in xrange(len(T)):
        S[i] = kB*np.log(partfunc(T[i],E))+meanE(T[i],E)/T[i]
    return S
```

Main Part:

```python
def main(argv):
    T = np.linspace(0,3,501)[1:-1]
    E = [np.array([0,0.1,1.]),np.array([0,0.5,1.]),np.array([0,0.9,1.])]
    C = [heatcap(T,E[0]),heatcap(T,E[1]),heatcap(T,E[2])]
    S = [entropy(T,E[0]),entropy(T,E[1]),entropy(T,E[2])]

    plt.figure(1)
    plt.plot(T,C[0],T,C[1],T,C[2])
    plt.title('Heat Capacity')
    plt.legend(('E1','E2','E3'));
    plt.xlabel('T[K]')
    plt.ylabel('Arb. units')

    plt.figure(2)
    plt.plot(T,S[0],T,S[1],T,S[2])
    plt.title('Entropy')
    plt.legend(('E1','E2','E3'),loc='lower right');
    plt.xlabel('T[K]')
    plt.ylabel('Arb. units')

    plt.show()
```

Results

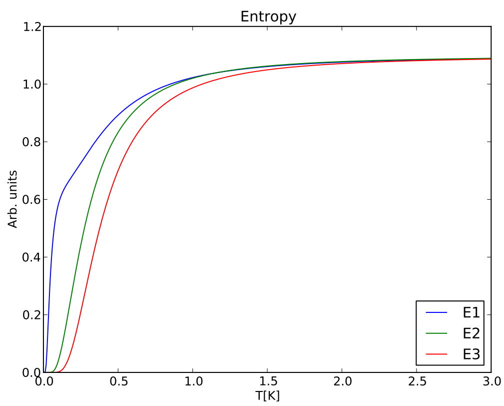

Figure 1: Entropy for $E1 = [0, 0.1, 1]$, $E2 = [0, 0.5, 1]$, $E3 = [0, 0.9, 1]$

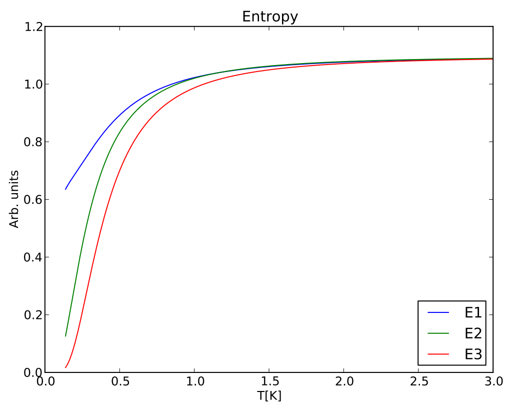

Figure 2: Entropy for $E1 = [100, 100.1, 101]$, $E2 = [100, 100.5, 101]$, $E3 = [100, 100.9, 101]$

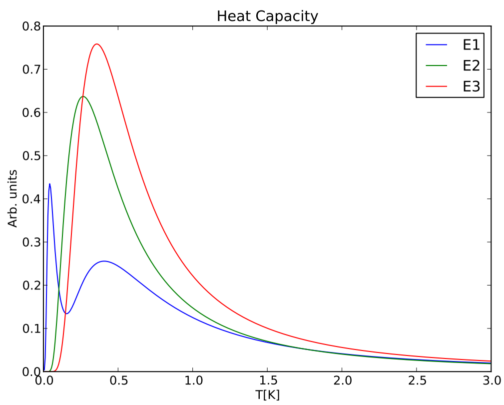

Figure 3: Heat capacity for $E1 = [0, 0.1, 1]$, $E2 = [0, 0.5, 1]$, $E3 = [0, 0.9, 1]$

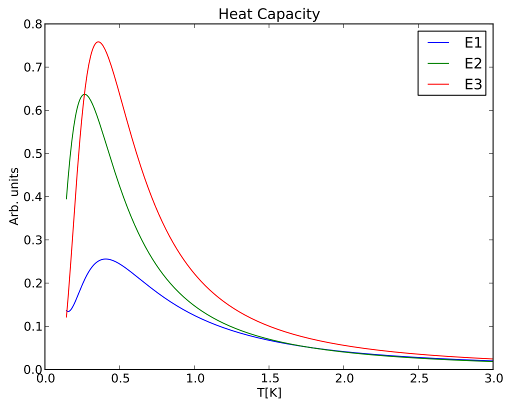

Figure 4: Heat capacity for $E1 = [100, 100.1, 101]$, $E2 = [100, 100.5, 101]$, $E3 = [100, 100.9, 101]$

I could not get python to compute any answers due to errors in the exp-function when $E_n \to E_n + 5000$, so I instead went with $E_n \to E_n + 100$. I still got errors for low values of T, but from figures 1, 2, 3 and 4 one can observe that the entropy and heat capacity have the same dependency on T in both cases.

Problem 10

Simulation

Partition Function:

$$Z = \sum_{s_z=-s}^{s} \exp(-\alpha s_z), \qquad \alpha = g\mu B\beta$$

Heat Capacity:

$$\begin{aligned} C &= \frac{\mathrm{d} < E >}{\mathrm{d}T} \\ < E > &= \frac{1}{Z}\sum_{j=0}^{\infty} \exp(-\beta E_j) * E_j \end{aligned}$$

Entropy:

$$S = k_B \ln Z - \frac{< E >}{T}$$

Magnetization:

$$< s_z > = \frac{\partial}{\partial \alpha} \ln Z$$

```python
def magnetization(s,T=[]):
    #Magnetization
    M = np.zeros(len(T))
    def lnZ(alpha):
        return np.log(np.sum(np.exp(-alpha*np.arange(-s,s+1,1))))
    for i in xrange(len(T)):
        M[i] = ddx(lnZ,1/T[i])
    return M
```

Susceptibility:

$$\chi = \frac{\partial}{\partial B} < s_z >= \frac{\partial}{\partial B}\frac{\partial}{\partial \alpha} \ln Z = \frac{B}{\alpha}\frac{\partial^2}{\partial \alpha^2} \ln Z$$

```
def susceptibility(s,T=[]):
    #Susceptibility
    X = np.zeros(len(T))
    def lnZ(alpha):
        return np.log(np.sum(np.exp(-alpha*np.arange(-s,s+1,1))))
    for i in xrange(len(T)):
        X[i] = ddx(lnZ,1/T[i],n=2)
    return X
```

Main Part:

```
def main(argv):
    T = np.linspace(0,8,501)[1:-1]
    E = ([np.array([-1/2,1/2]),np.array(range(-3,3+1)),
        np.array(range(-10,10+1)),np.array(range(-100,100+1))])
    C = [heatcap(T,E[0]),heatcap(T,E[1]),heatcap(T,E[2]),heatcap(T,E[3])]
    S = [entropy(T,E[0]),entropy(T,E[1]),entropy(T,E[2]),heatcap(T,E[3])]
    M = ([magnetization(1/2,T),magnetization(3,T),magnetization(10,T),
        magnetization(100,T)])
    X = ([susceptibility(1/2,T),susceptibility(3,T),susceptibility(10,T),
        susceptibility(100,T)])

    plt.figure(1)
    plt.plot(T,C[0],T,C[1],T,C[2],T,C[3])
    plt.title('Heat Capacity')
    plt.legend(('s=1/2','s=3','s=10','s=100'));
    plt.xlabel('T[K]')
    plt.ylabel('Arb. units')

    plt.figure(2)
    plt.plot(T,S[0],T,S[1],T,S[2],T,C[3])
    plt.title('Entropy')
    plt.legend(('s=1/2','s=3','s=10','s=100'),loc='lower right');
    plt.xlabel('T[K]')
    plt.ylabel('Arb. units')

    plt.figure(3)
    plt.plot(T,M[0],T,M[1],T,M[2],T,M[3])
    plt.title('Magnetization')
    plt.legend(('s=1/2','s=3','s=10','s=100'));
    plt.xlabel('T[K]')
    plt.ylabel('Arb. units')

    plt.figure(4)
    plt.plot(T,X[0],T,X[1],T,X[2],T,X[3])
    plt.title('Susceptibility')
    plt.legend(('s=1/2','s=3','s=10','s=100'));
    plt.xlabel('T[K]')
    plt.ylabel('Arb. units')
```

```
    plt.show()
```

Results

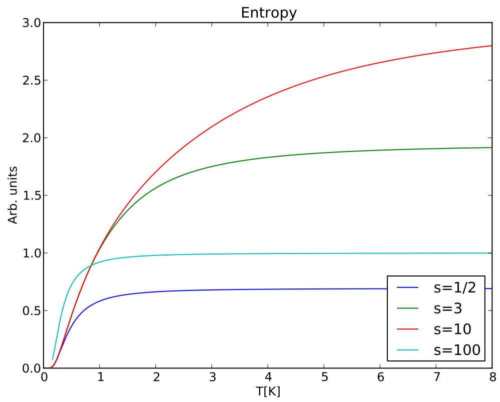

Figure 5: Entropy

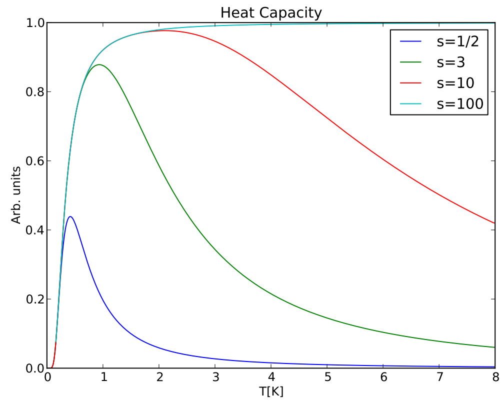

Figure 6: Heat Capacity

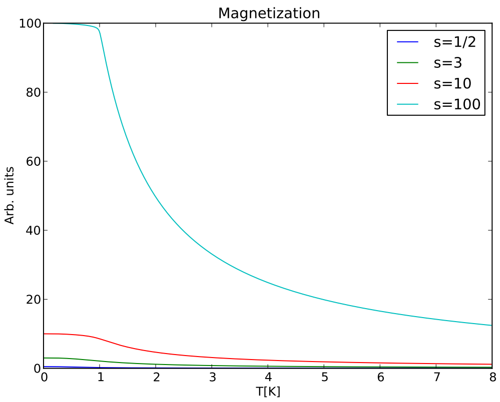

Figure 7: Magnetization

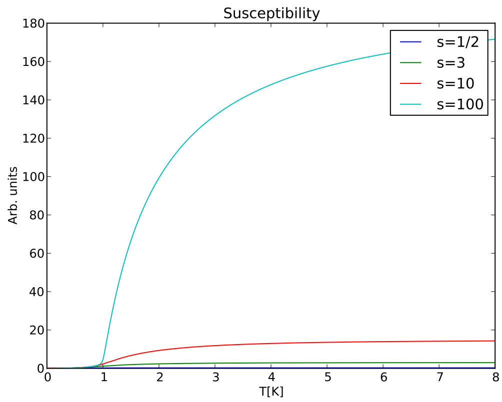

Figure 8: Susceptibility

Problem 11

Partition function:

$$
\begin{aligned}
Z_{vib} &= \sum_{n=0}^{\infty} \exp(-\beta E_n) = (\sinh(\beta\hbar\omega/2))^{-1} \\
Z_{rot} &= \sum_{l=0}^{\infty} (2l+1)\exp(-\beta E_l)
\end{aligned}
$$

Code:

```python
kB = 1
hbar = 1
omega = 1

def Zvib(T=[]):
```

```python
    return 1/(2*np.sinh(hbar*omega/(2*kB*T)))

def meanEvib(T=[]):
    return (1/2)*hbar*omega*(1+(2/(np.exp(hbar*omega/(kB*T))-1)))

def entropyvib(T=[]):
    return kB*np.log(Zvib(T))+meanEvib(T)/T

def heatcapvib(T=[]):
    return np.gradient(meanEvib(T),T[1]-T[0])

def Zrot(T=[]):
    Z = np.zeros(len(T))
    i = 0
    while T[i] < 0.67:
        i += 1
    Z[0:i] = (1 + 3*np.exp(-2/T[0:i]) + 5*np.exp(-6/T[0:i]) +
        7*np.exp(-12/T[0:i]) + 9*np.exp(-20/T[0:i]) + 11*np.exp(-30/T[0:i]))
    Z[i:-1] = T[i:-1]
    return Z

def meanErot(T=[]):
    meanE = np.zeros(len(T))
    i = 0
    while T[i] < 0.67:
        i += 1
    meanE[0:i] = (6*np.exp(-2/T[0:i]) + 30*np.exp(-6/T[0:i]) +
        84*np.exp(-12/T[0:i]) + 180*np.exp(-20/T[0:i]) + 330*np.exp(-30/T[0:i]))
    meanE[i:-1] = T[i:-1]
    return meanE

def entropyrot(T=[]):
    return kB*np.log(Zrot(T))+meanErot(T)/T

def heatcaprot(T=[]):
    tmp = np.gradient(meanErot(T),T[1]-T[0])
    C = np.ones(len(T)) + 1/(45*T**2)
    i = 1
    while tmp[i] <= C[i]:
        i += 1
    C[0:i-2] = tmp[0:i-2]
    return C


def main(argv):
    T = np.linspace(0,3,501)[1:-1]
    C = [heatcapvib(T),heatcaprot(T)]
    S = [entropyvib(T),entropyrot(T)]

    plt.figure(1)
    plt.plot(T,C[0],T[0:-5],C[1][0:-5])
    plt.title('Heat Capacity')
    plt.legend(('Vibrational','Rotational'));
```

```
    plt.xlabel('T[K]')
    plt.ylabel('Arb. units')

    plt.figure(2)
    plt.plot(T,S[0],T,S[1])
    plt.title('Entropy')
    plt.legend(('Vibrational','Rotational'),loc='lower right');
    plt.xlabel('T[K]')
    plt.ylabel('Arb. units')

    plt.show()
```

Results

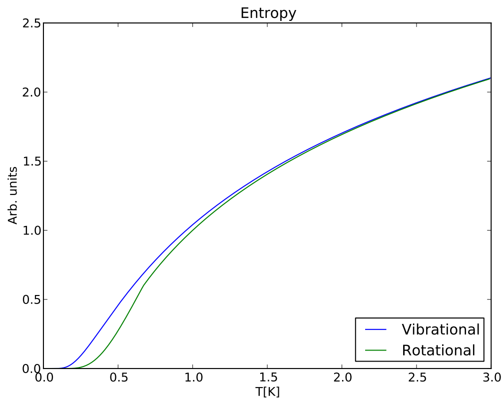

Figure 9: Entropy

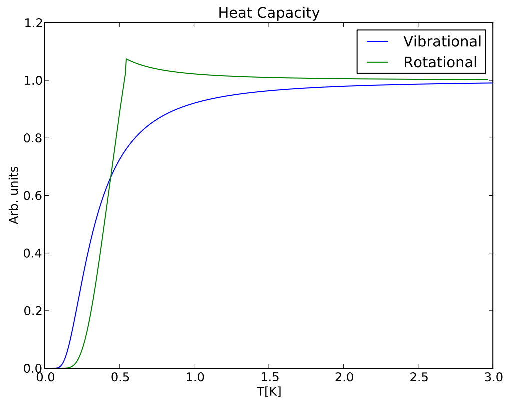

Figure 10: Heat Capacity

The small jumps in the rotator entropy and heat capacity in figure 9 and 10 stems from limited series-expansion of the partition function at "low" temperatures, and a linear approximation at "high" temperatures.

Problem 12

1D

$$\widetilde{E} = \frac{\partial}{\partial \beta}\widetilde{F} = \int_0^{\omega_D} \mathrm{d}\omega \frac{\partial}{\partial \beta} \ln(1 - \exp(-\beta\omega)) = \int_0^1 \mathrm{d}\omega \frac{\omega}{\exp(\omega/k_B T) - 1}$$

Ignoring constant (setting them equal to unity).

$$C = \frac{\partial E}{\partial T} = \frac{1}{T^2} \int_0^1 \mathrm{d}\omega \frac{\omega^2 \exp(\omega/T)}{(\exp(\omega/T) - 1)^2}$$

```
def heatcapLine(T=[]):
    #Heat Capacity for line
    def integral(w,T):
        return (w**2*np.exp(w/T))/((np.exp(w/T)-1)**2)
    wD = 1
    C = np.zeros(len(T))
    for i in xrange(len(T)):
        C[i] = (1/T[i]**2)*quad(integral,0,wD,args=(T[i],))[0]
    return C
```

$$S = -\frac{\partial}{\partial T}\widetilde{F} = \frac{1}{T^2}\int\limits_0^{\omega_D} \mathrm{d}\omega frac\omega\exp(\omega/T) - 1$$

```
def entropyLine(T=[]):
    #Entropy for line
    def integral(w,T):
        return w/(np.exp(w/T)-1)
    wD = 1
    S = np.zeros(len(T))
    for i in xrange(len(T)):
        S[i] = (1/T[i]**2)*quad(integral,0,wD,args=(T[i],))[0]
    return S
```

3D

$$\widetilde{E} = \frac{\partial}{\partial \beta}\widetilde{F} = \int\limits_0^{\omega_D} \omega^2 \mathrm{d}\omega \frac{\partial}{\partial \beta}\ln(1-\exp(-\beta\omega)) = \int\limits_0^1 \mathrm{d}\omega \frac{\omega^3}{\exp(\omega/k_B T) - 1}$$

$$C = \frac{\partial E}{\partial T} = \frac{1}{T^2}\int\limits_0^1 \mathrm{d}\omega \frac{\omega^4 \exp(\omega/T)}{(\exp(\omega/T)-1)^2}$$

```
def heatcap3D(T=[]):
    #Heat Capacity in 3D
    def integral(w,T):
        return (w**4*np.exp(w/T))/((np.exp(w/T)-1)**2)
    wD = 1
    C = np.zeros(len(T))
    for i in xrange(len(T)):
        C[i] = (1/T[i]**2)*quad(integral,0,wD,args=(T[i],))[0]
    return C
```

$$S = -\frac{\partial}{\partial T}\widetilde{F} = \frac{1}{T^2}\int\limits_0^{\omega_D} \mathrm{d}\omega frac\omega^3\exp(\omega/T) - 1$$

```
def entropy3D(T=[]):
    #Entropy in 3D
    def integral(w,T):
        return w**3/(np.exp(w/T)-1)
    wD = 1
    S = np.zeros(len(T))
    for i in xrange(len(T)):
        S[i] = (1/T[i]**2)*quad(integral,0,wD,args=(T[i],))[0]
    return S
```

Results

$$|k_{max}|c_s = \omega_D$$

Entropy decreasing as a function of temperature as seen in figure 11 is peculiar, but no errors in the derivation could be found. It could however have something to do with the simplifications done in the derivation...

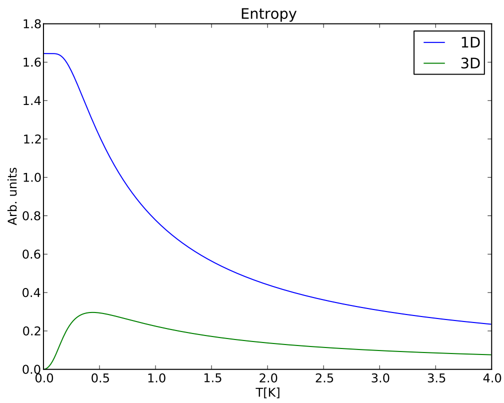

Figure 11: Entropy

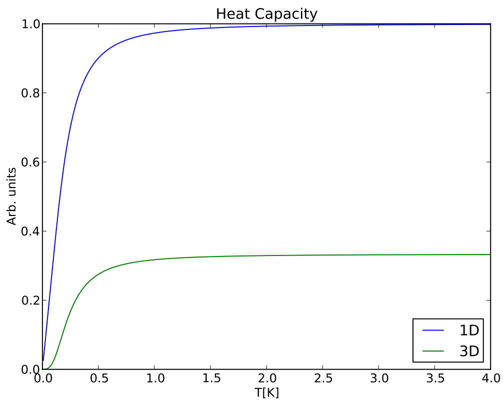

Figure 12: Susceptibility