

FERMI GAS: HEAT CAPACITY APPROXIMATION



Patrick McMillin

Dr. Taheri

PHYS 431

7 December 2017

ASSUMPTIONS AND GOAL

- Test two approximations for a Fermi Gas which:
 - Is trapped such that the energy levels are evenly spaced
 - Assuming spacing such that:
$$U = \eta q$$
 - Where ‘U’ is the total energy, ‘q’ are the number of energy packets, and ‘ η ’ is the spacing of the energy levels.
 - Has nondegenerate energy levels (only 1 fermion per level)
 - The goal is to show that these approximations give the correct relationship for heat capacity for a Fermi Gas, as found analytically in class though the Sommerfeld Expansion.

UNRESTRICTED PARTITIONS

- An “unrestricted partition” is the number of ways that a non-negative integer can be expressed as a sum of other non-negative integers (excluding degeneracies).
- I will define $p(q)$ as the unrestricted partition of the number ‘ q ’.
- This relates to the nondegenerate Fermi Gas with evenly spaced energy levels because the unrestricted partition of the number of energy packets is **the number of ways the available energy can be arranged among the fermions in the Fermi Gas**.
 - This is the number of microstates which correspond to a particular energy level.

UNRESTRICTED PARTITIONS

GENERATING FUNCTION AND THE PENTAGONAL NUMBERS

- There is a generating function for unrestricted partitions which comes from the Pentagonal Number Theorem:

$$p(n) = \sum_{k=1}^n (-1)^{k-1} p(n - g_k)$$

- Where the sum is taken over all non-zero values of k , and g_k is the k^{th} generalized pentagonal number given by:

$$g_k(i) = \frac{i(3i - 1)}{2}$$

GENERATING UNRESTRICTED PARTITIONS

```
def Generate_Unrestricted_Partitions():
    # Finding pentagonal numbers to be used in Euler's formula
    Pentagonal_Numbers = sum([[i*(3*i - 1)/2, i*(3*i - 1)/2 + i] for i in range(1, 250)], [])
    # Initial unrestricted partition (requires initial condition). Unrestricted partition of 0 is 1.
    n = 0
    Unrestricted_Partitions = [1]
    # Sign convention for expanding Euler's formula
    Sign_Convention = [1, 1, -1, -1]
    # Looping over all numbers, finding unrestricted partitions for the first 100 numbers.
    while n < 100:
        n += 1 #iterate
        Unrestricted_Partition_for_n = 0 # To be added to in Euler's formula.
        i = 0 # Iterable for pentagonal numbers summed over in Euler's formula.
        while Pentagonal_Numbers[i] <= n: # Defined only for n-p >=0
            # Adds to the unrestricted partition by Euler's formula. Ensures proper sign convention.
            Unrestricted_Partition_for_n += Unrestricted_Partitions[n - Pentagonal_Numbers[i]] * Sign_Convention[i % 4]
            i += 1 # iterate
            # Add our newly found unrestricted partition to the list.
        Unrestricted_Partitions.append(Unrestricted_Partition_for_n)
    return Unrestricted_Partitions
Unrestricted_Partitions = Generate_Unrestricted_Partitions()
Integer_List = range(0,100)
import numpy as np
np.savetxt('Unrestricted_Partitions.csv', Unrestricted_Partitions, fmt='%10.5f', delimiter=',')
```

Python code which generates the unrestricted partitions of numbers up to 100.

THE FIRST 100 UNRESTRICTED PARTITIONS

To the left are the unrestricted partitions for the numbers 0 through 100. For reference, below is a list of the first few unrestricted partitions from a Wolfram Research website.

n	0	1	2	3	4	5	6	7	8	9	10	...	100
$p(n)$	1	1	2	3	5	7	11	15	22	30	42	...	190569292

The fact that the 100th unrestricted partition is correct is very important. Since the generating function depends on the values of the unrestricted partitions calculated before it, this tell us that all the unrestricted partitions which come before 100 are also correct.

100th Unrestricted Partition :
190569292

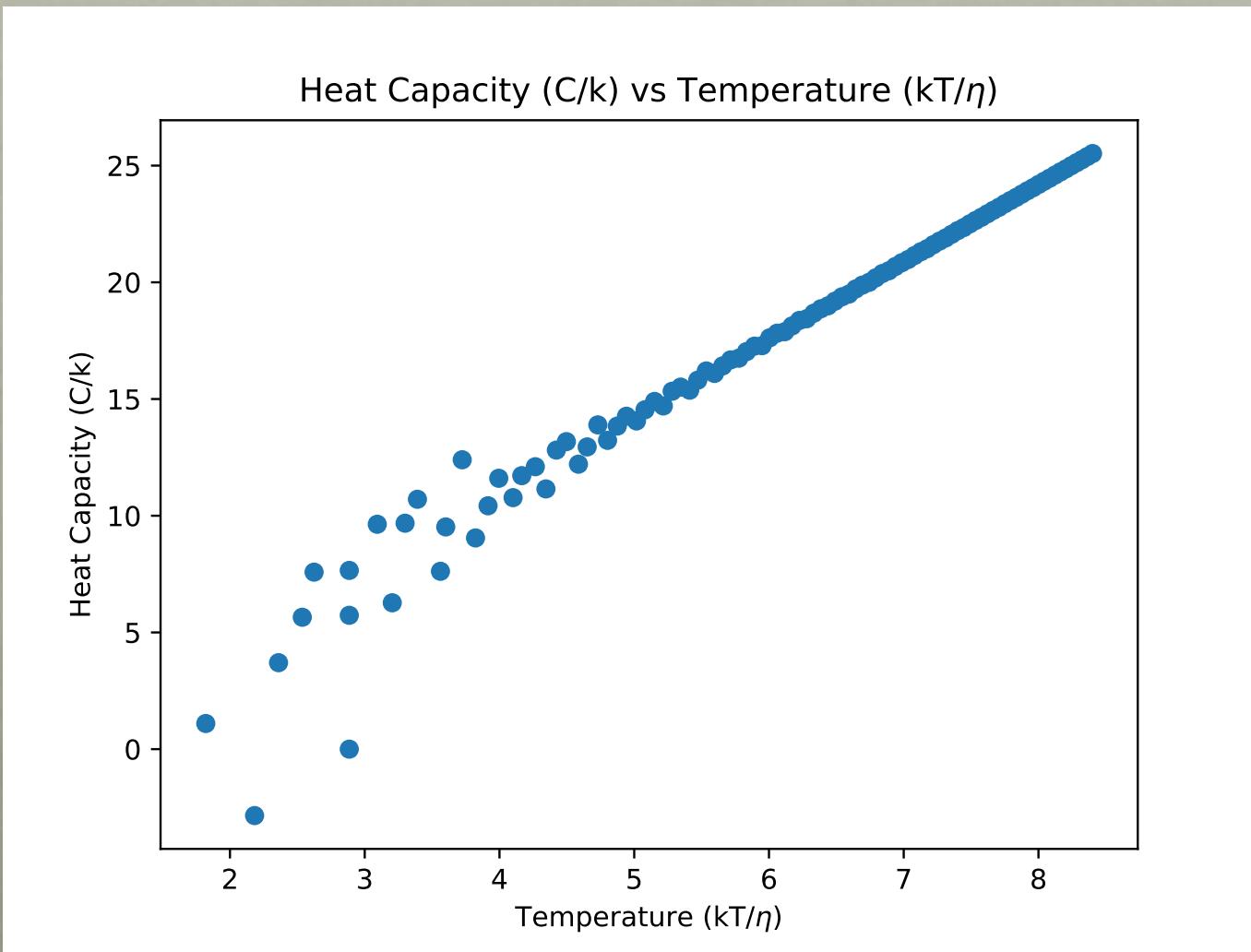
ENTROPY, TEMPERATURE, AND HEAT CAPACITY

$\Omega = p(q)$	Available microstates are equal to the unrestricted partitions.
$S = k \ln(\Omega)$	Entropy is the natural logarithm of the partitions.
$T = \frac{\Delta U}{\Delta S}$	Temperature from the definition.
$C_V = \frac{\Delta U}{\Delta T}$	Heat Capacity from the definition.
$T[i] = \frac{U[i+1] - U[i-1]}{S[i+1] - S[i-1]}$	The temperature and heat capacities are calculated from the “centered difference” method.
$C_V[i] = \frac{U[i+1] - U[i-1]}{T[i+1] - T[i-1]}$	

FINDING HEAT CAPACITY AND PLOTTING

```
import numpy as np
import matplotlib.pyplot as plt
# Script written to find the partitions
import Unrestricted_Partitions
# Calls function to generate partitions and assigns them to a list.
Partitions = Unrestricted_Partitions.Generate_Unrestricted_Partitions()
# Energy levels are just the numerical values we found the
# unrestricted partitions for (i.e. the numbers 0 through 100).
# For simplicity,  $\eta = k = 1$ .
Energy = range(0,101)
Entropy = []
# Takes the logarithm of all microstates to find S/k.
for i in Partitions:
    Entropy.append(np.log(i))
Temperature = [0]
Heat_Capacity = [0]
# Temperatures
for i in range(1,(len(Energy)-1)): # NOTE: len()-1 since we will not have a Energy[101]
    Temperature.append((Energy[i+1]-Energy[i-1]) / (Entropy[i+1]-Entropy[i-1]))
# Heat capacities
for i in range(1,(len(Energy)-2)): # NOTE: len()-1 since we will not have a Temperature[100]
    Heat_Capacity.append((Energy[i+1]-Energy[i-1]) / (Temperature[i+1]-Temperature[i-1]))
del Temperature[0]
plt.figure(1)
plt.scatter(Temperature, Heat_Capacity)
plt.title('Heat Capacity (C/k) vs Temperature (kT/$\eta$)')
plt.xlabel('Temperature (kT/$\eta$)')
plt.ylabel('Heat Capacity (C/k)')
plt.savefig('T_vs_Cv.pdf')
```

HEAT CAPACITY FROM PARTITIONS



RAMANUJAN'S APPROXIMATION

- Ramanujan and Hardy developed an equation which estimates the unrestricted partitions of a number:

$$p(q) \approx \frac{e^{\pi\sqrt{2q/3}}}{4q\sqrt{3}}$$

- This equation works well in the limit of high q (large temperatures compared to the energy level spacing).
 - $p(100) = 190569292$ by the generating function
 - $p(100) = 175236785$ by Ramanujan's approximation.
 - $p(1000)$ is only 1.5% larger than the actual value.

ENTROPY FROM RAMANUJAN'S APPROXIMATION

$$\frac{1}{T} = \frac{\partial S}{\partial U} = \frac{\partial q}{\partial U} \frac{\partial S}{\partial q}$$

From Ramanujan's equation, we can easily find an expression for entropy to differentiate to find the temperature.

$$\frac{kT}{\eta} = \frac{q}{\pi \sqrt{q/6 - 1}}$$

Heat capacity requires energy as a function of temperature, which is what we can obtain if we solve this quadratic equation.

$$C_V = \frac{\partial U}{\partial T} = \frac{\partial U}{\partial q} \frac{\partial q}{\partial T}$$

This is where we make the approximation for $kT \gg \eta$. We can expand the solution to the quadratic equation, and obtain the next equation for energy as a function of temperature.

$$q = \frac{k^2 \pi^2 T^2}{6\eta^2} - \frac{2kT}{\eta}$$

Finally, we differentiate the equation, and obtain our approximation for the heat capacity.

$$C_V = k \left(\frac{\pi^2}{3} \frac{kT}{\eta} - 2 \right)$$

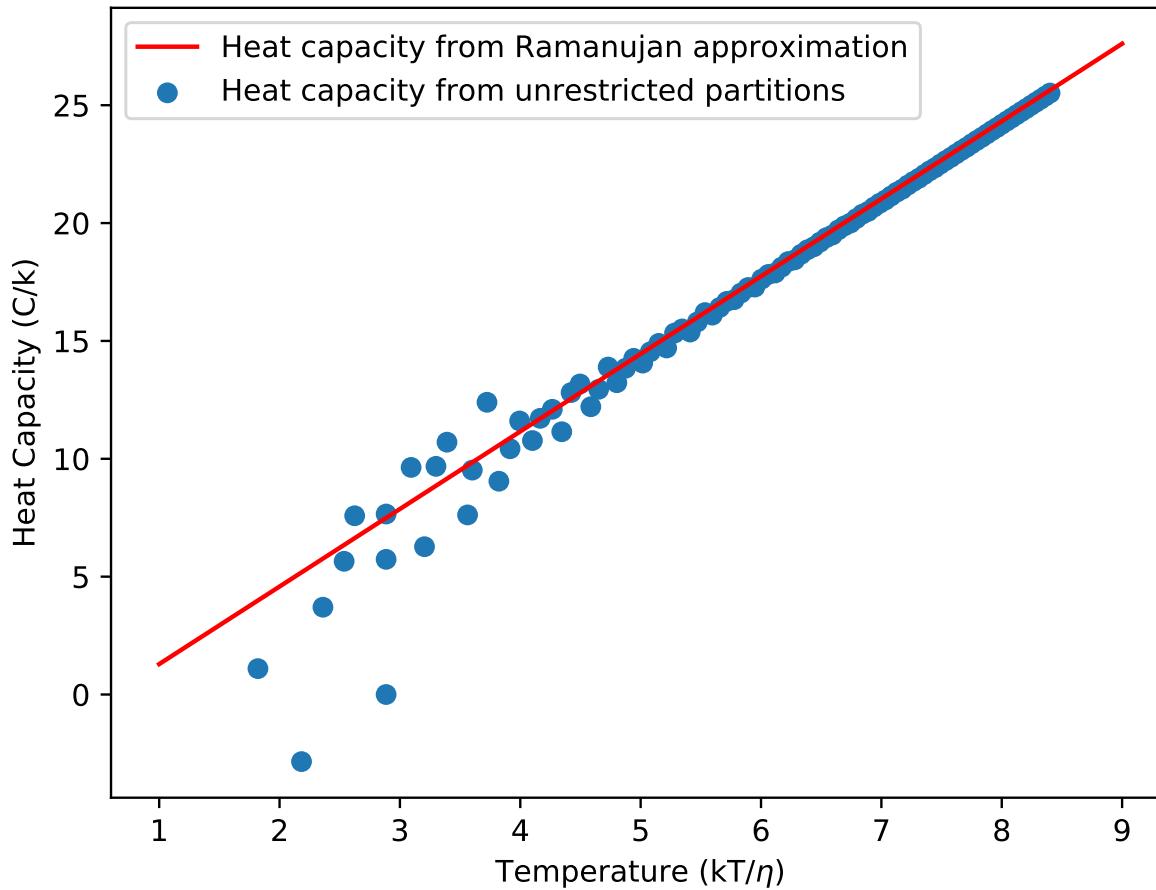
PLOTTING RAMANUJAN'S APPROXIMATION

```
plt.figure(2)
Temp_List = np.linspace(1,9,100)
Ramanujan_Heat_Capacity_Coefficient = ((np.pi)**2)/3
P = plt.scatter(Temperature, Heat_Capacity)
R = plt.plot(Temp_List, (Ramanujan_Heat_Capacity_Coefficient*Temp_List)-2, 'r-')
plt.title('Ramanujans Approximation for Heat Capacity (C/k) vs Temperature (kT/$\eta$)')
plt.xlabel('Temperature (kT/$\eta$)')
plt.ylabel('Heat Capacity (C/k)')
plt.legend(['Heat capacity from Ramanujan approximation','Heat capacity from unrestricted partitions'])
plt.savefig('Ramanujan_T_vs_Cv.pdf')
```

Continued from the last script where the heat capacity of the unrestricted partitions was plotted.

COMPARISON

Ramanujans Approximation for Heat Capacity (C/k) vs Temperature (kT/η)



CONCLUSIONS

- We find that the two approximations used for the heat capacity of a Fermi Gas follow the behavior found from the Sommerfeld expansion:

$$C_V = \frac{\pi^2 N k^2 T}{2\epsilon_F}$$

- We see explicitly the correct dependence in the formula derived from Ramanujan's approximation:

$$C_V = k \left(\frac{\pi^2}{3} \frac{kT}{\eta} - 2 \right)$$

DIFFERENCES

- The main difference between this special case and the case investigated by using the Sommerfeld expansion is the lack of the ‘N’ dependence.
- The approximation does not depend on N because we did not make any assumptions about the volume of the system. Our assumption of evenly spaced energy levels hides the need for a volume and distribution function for the system.

SOURCES

- [http://functions.wolfram.com/IntegerFunctions/
PartitionsP/introductions/Partitions>ShowAll.html](http://functions.wolfram.com/IntegerFunctions/PartitionsP/introductions/Partitions>ShowAll.html)
- J. Arnaud et al., *American Journal of Physics* **67**, 215 (1999).
- Schroeder, Daniel V. *An Introduction to Thermal Physics*. Pearson, 2014.