

1). Consider the outcome of a coin toss experiment as an archetypal two-level system. Examine the outcome of simultaneous tossing of multiple coins at once, where $N_{\text{coin}} = 2, 4, 8, 16, \dots$. By considering a macrostate in this context to be defined/represented by the number of heads in the outcome, plot the probability of various macrostates possible, versus the number of heads in each case.

2. For each value of N_{coin} chosen, find the most probable value of the above distributions. Is there some pattern in the answers for most probable value? Justify.

CODE:

```
// Macrostates and Microstates in Coin-Tossing Experiment

clc;clf();clear;

for j = 1:7
    N_coins = 2^j

    N_H = 0:N_coins

    for i = 1:N_coins+1
        c(i)=factorial(N_coins)/(factorial(N_H(i))*factorial(N_coins-N_H(i)))
        prob(i) = c(i)/2^N_coins
    end

    [MaxVal, Ind] = max(prob);
    disp(sprintf("Max probable value is %f on index no. %d", MaxVal, N_H(Ind)));

    //Plot graph
    subplot(3,3,j)
    plot(N_H',prob,"*")
    xlabel("No of heads")
    ylabel("Probabilities")
    title(sprintf("Plot for coins = 2^%d",j))
end
```

OUTPUT:

"Max probable value is 0.500000 on index no. 1"

"Max probable value is 0.375000 on index no. 2"

"Max probable value is 0.273438 on index no. 4"

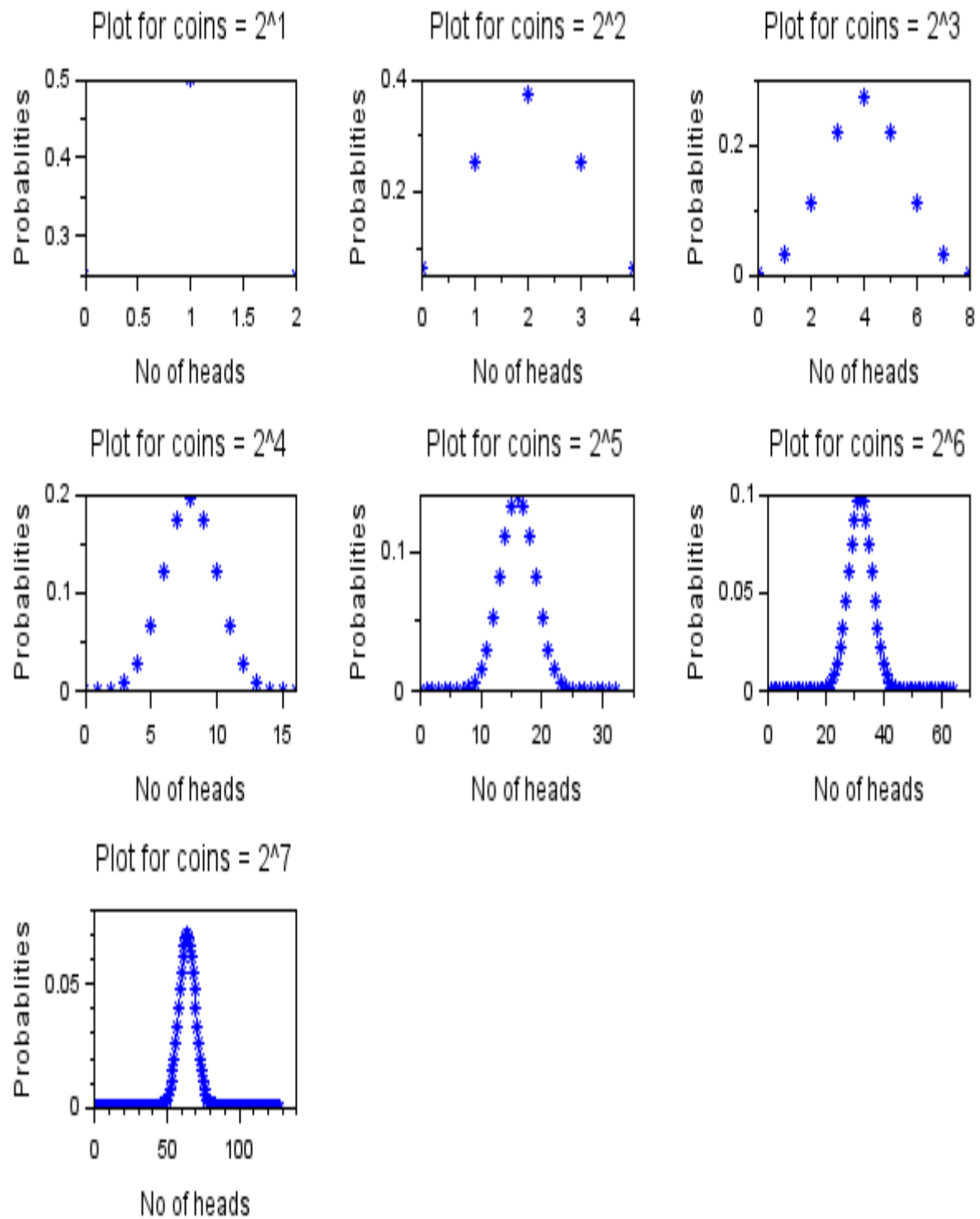
"Max probable value is 0.196381 on index no. 8"

"Max probable value is 0.139950 on index no. 16"

"Max probable value is 0.099347 on index no. 32"

"Max probable value is 0.070386 on index no. 64"

GRAPH:



Q2)(I) Relative temperature-dependence:

For any choice of a solid, compute the Specific Heat Capacity as a function of temperature, in accordance with:

(i) Dulong-Petit Law

(ii) Einstein relation for specific heat capacity of solids

(iii) Debye relation for specific heat capacity of solids

Plot the manner of variation of $C_V / (3R)$ as per the three laws, against temperature.

CODE:

```
//Specific Heat Capacities of Solids
```

```
clc;clear;clf();
```

```
R = 8.314;
```

```
//Define Temperature of Indium
```

```
T_d = 108;
```

```
T_e = 108*0.806;
```

```
n=500;
```

```
T = 0.16:n;
```

```
//Dulong and petit law
```

```
C_v1 = 3*R;
```

```
m1 = C_v1/(3*R)
```

```
//Einstein law
```

```
for i = 1:n
```

```
    C_v2(i) = 3*R*(T_e/T(i))^2 * (exp(T_e/T(i)))/(exp(T_e/T(i))-1)^2)
```

```
    m2(i)=C_v2(i)/(3*R)
```

```
end
```

```
//Debye law
```

```
function y=f(x)
```

```
    y = (x^4*exp(x))/(exp(x)-1)^2
```

```
endfunction
```

```
for i = 1:n
```

```
    y1(i) = intg(0,T_d/T(i),f)
```

```
    C_v3(i) = 9*R*(T(i)/T_d)^3*y1(i)
```

```
    m3(i) = C_v3(i)/(3*R)
```

```
end
```

```
//Plot graph
```

```
plot(T,m1,"--*r")
```

```
plot(T,m2,"g")
```

```
plot(T,m3,"--k")
```

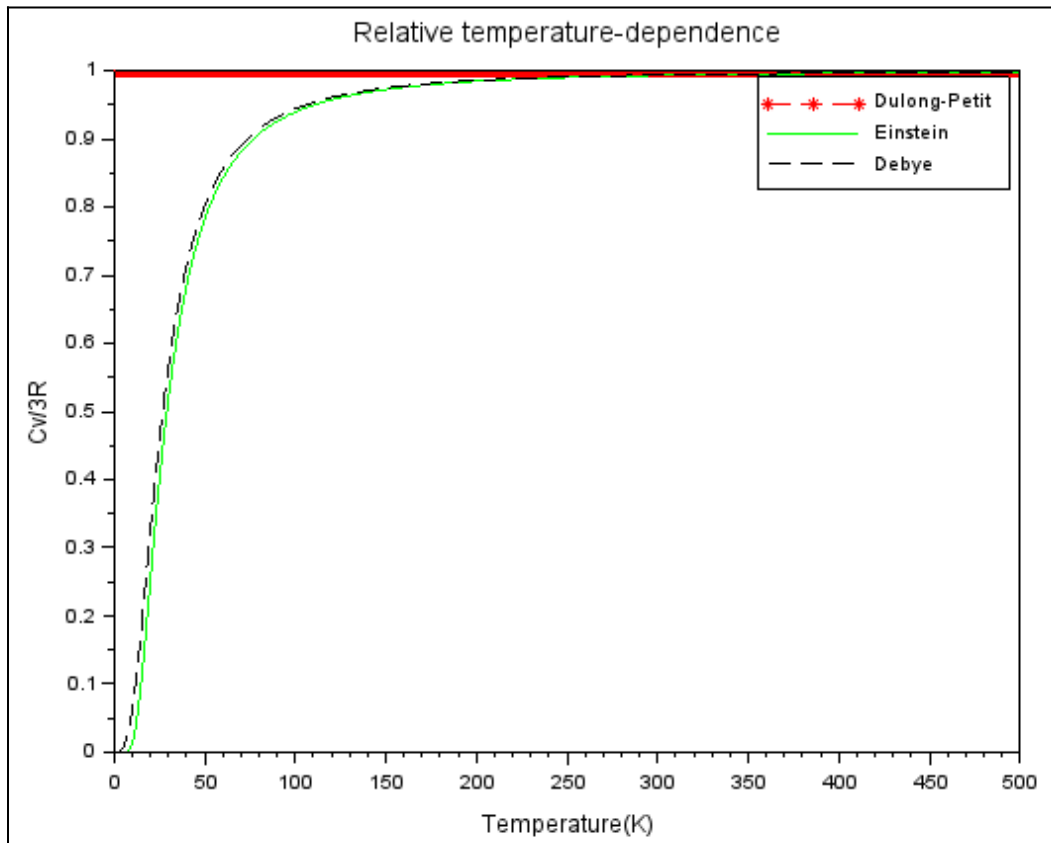
```
xlabel("Temperature(K)")
```

```
ylabel("Cv/3R")
```

```
legend("Dulong-Petit","Einstein","Debye")
```

```
title("Relative temperature-dependence")
```

GRAPH:



Q2)(II) Compare the computed values of CV from the Debye-integral relation, and (the low-temperature limit of the same, i.e.) the Debye T^3 law.

In an appropriate temperature range, co-plot these two computed CV as a function of T^3 , and hence, establish their convergence at low temperatures.

CODE:

```
clc;clear;clf();
R = 8.314;
//Define Temperature of Indium
T_d = 108;
n = 15;
T = 0.16:n;

//Debye integral relation
function y=f(x)
    y = (x^4*exp(x))/(exp(x)-1)^2
endfunction

for i = 1:n
    y1(i) = intg(0,T_d/T(i),f)
    C_v1(i) = 9*R*((T(i)/T_d)^3)*y1(i)
end

//Debye  $T^3$  law
for i = 1:n
```

```

C_v2(i) = ((12/5)*((%pi)^4)*R)*(T(i)/T_d)^3
end

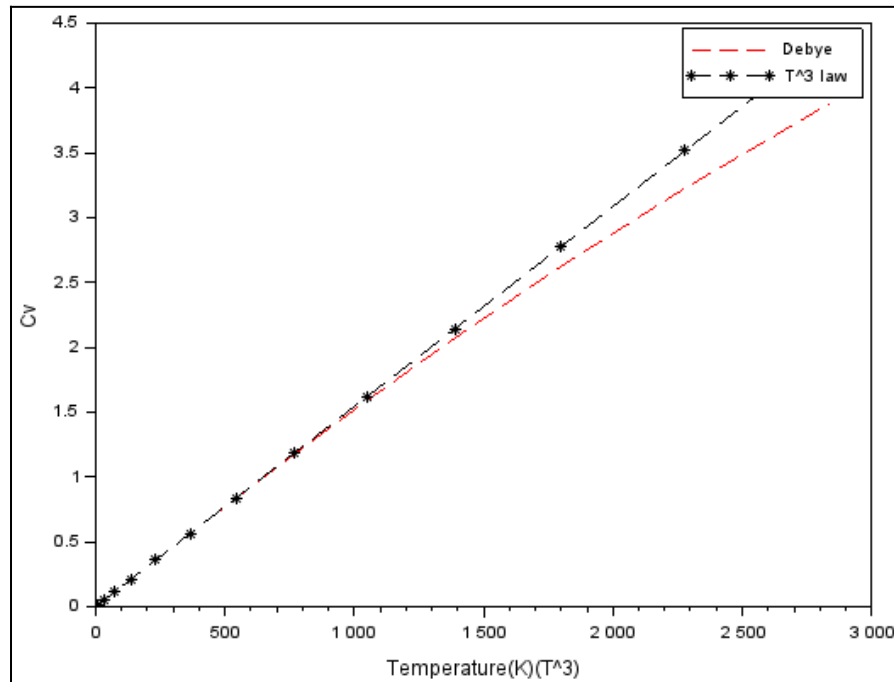
```

```

//Plot graph
plot(T^3',C_v1,"-r")
plot(T^3',C_v2,"*-*k")
xlabel("Temperature(K)(T^3)")
ylabel("Cv")
legend("Debye","T^3 law")

```

GRAPH:



Q2) (III) In some appropriate low-temperature range, co-plot the specific heat capacities computed using Einstein and Debye relations, as a function of T^3 . Hence, establish that the former does not obey the T^3 law.

CODE:

```

// Comparison between Einstein and Debye relations

```

```

clc;clear;clf();

```

```

R = 8.314;

```

```

//Define Temperature of Indium

```

```

T_d = 108;

```

```

T_e = 108*0.806;

```

```

n = 15;

```

```

T = 0.16:n;

```

```

//Einstein law

```

```

for i = 1:n
    C_v1(i) = 3*R*(T_e/T(i))^2 * (exp(T_e/T(i)))/(exp(T_e/T(i))-1)^2
end

//Debye integral relation
function y=f(x)
    y = (x^4*exp(x))/(exp(x)-1)^2
endfunction

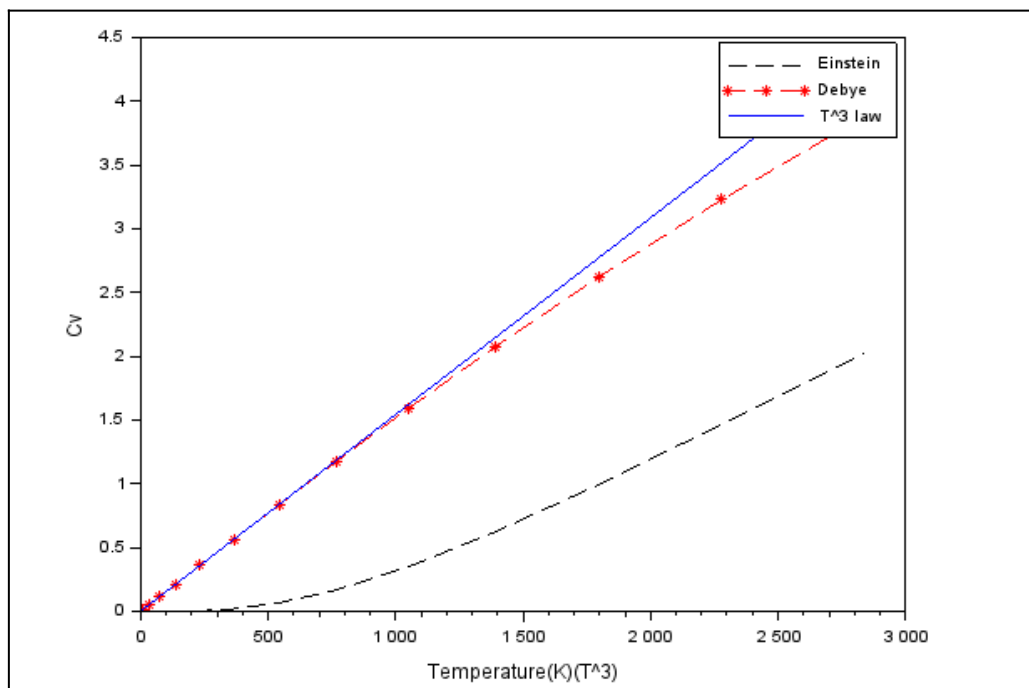
for i = 1:n
    y1(i) = intg(0,T_d/T(i),f)
    C_v2(i) = 9*R*((T(i)/T_d)^3)*y1(i)
end

//Debye T^3 law
for i = 1:n
    C_v3(i) = ((12/5)*((%pi)^4)*R)*(T(i)/T_d)^3
end

//Plot graph
plot(T^3,C_v1,"--k")
plot(T^3,C_v2,"*--*r")
plot(T^3,C_v3,"b")
xlabel("Temperature(K)(T^3)")
ylabel("Cv")
legend("Einstein","Debye","T^3 law")

```

GRAPH:



Q3) Individual Plots:

For each of the following distribution laws, at three suitably-chosen distinct temperatures, plot the energy distribution as a function of (a) frequency, and (b) wavelength.

(i) Rayleigh-Jeans Law

(ii) Wien's Distribution Law

(iii) Planck's Distribution Law

CODE:

```
//Frequency distribution function
```

```
clc;clear;clf();
```

```
// Define constants
```

```
T = [500, 700, 900, 1000]; // Temperatures in Kelvin
```

```
kb = 1.38e-23; // Boltzmann constant
```

```
h = 6.6260e-34; // Planck constant
```

```
c = 3e8; // Speed of light
```

```
// Define frequency range and wavelength range
```

```
n = 500;
```

```
f = linspace(0, 3e14, n);
```

```
// Loop through each temperature
```

```
for j = 1:length(T)
```

```
    for i = 1:n
```

```
        // Calculate Rayleigh-Jeans law
```

```
        E1(i, j) = (8 * %pi * kb * T(j) * f(i)^2) / c^3;
```

```
        // Calculate Planck's law
```

```
        E2(i, j) = ((8 * %pi * h * f(i)^3) / c^3) * (1 / (exp(h * f(i) / (kb * T(j))) - 1));
```

```
        // Calculate Wien's law
```

```
        E3(i, j) = ((8 * %pi * h * f(i)^3) / c^3) * (1 / exp(h * f(i) / (kb * T(j))));
```

```
    end
```

```
end
```

```
// Plot the results
```

```
subplot(3, 2, 1);
```

```
plot(f', E1);
```

```
xlabel("frequency");
```

```
ylabel("Energy");
```

```
title("Rayleigh-Jeans law");
```

```
subplot(2, 2, 2);
```

```
plot(f', E2);
```

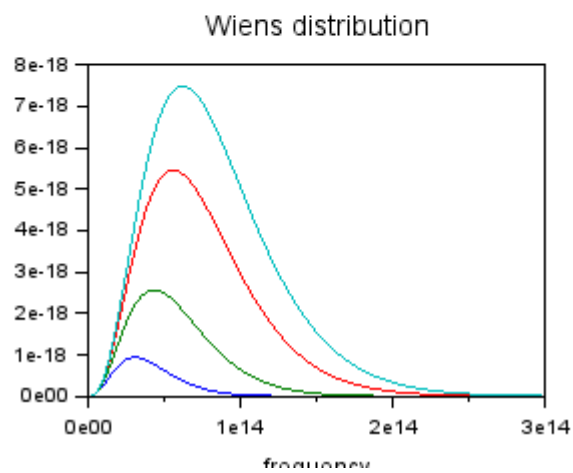
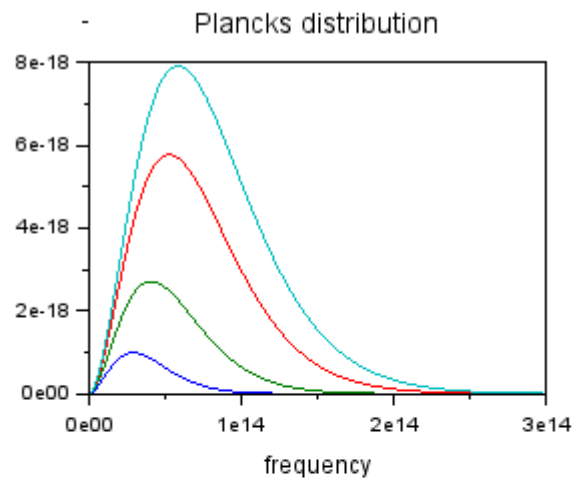
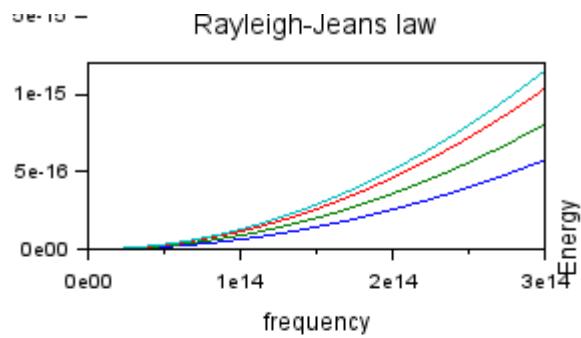
```
xlabel("frequency");
```

```
ylabel("Energy");
```

```
title("Plancks distribution");
```

```
subplot(2, 2, 3);  
plot(f', E3);  
xlabel("frequency");  
ylabel("Energy");  
title("Wiens distribution");
```

OUTPUT:



// Wavelength distribution function

```
clc;clear;clf();
```

```
// Define constants
```

```
T = [2000, 2500, 3000, 3500]; // Temperatures in Kelvin
```

```
kb = 1.38e-23; // Boltzmann constant
```

```
h = 6.6260e-34; // Planck constant
```

```
c = 3e8; // Speed of light
```

```
// Define wavelength range
```

```
n = 500;
```

```
L = linspace(0, 5e-6, n)
```

```
// Loop through each temperature
```

```
for j = 1:length(T)
```

```
    for i = 1:n
```

```
        // Calculate Rayleigh-Jeans law
```

```
        E1(i, j) = ([8 * %pi * kb * T(j)] / L(i)^4);
```

```
        // Calculate Planck's law
```

```
        E2(i, j) = [(8 * %pi * h * c) / L(i)^5] * (1 / (exp(h * c / (kb * T(j) * L(i))) - 1));
```

```
        // Calculate Wien's law
```

```
        E3(i, j) = [(8 * %pi * h * c) / L(i)^5] * (1 / exp(h * c / (kb * T(j) * L(i))));
```

```
    end
```

```
end
```

```
//Plot
```

```
subplot(2, 2, 1);
```

```
plot(L', E1);
```

```
xlabel("wavelength");
```

```
ylabel("Energy");
```

```
title("Rayleigh-Jeans law");
```

```
a = gca()
```

```
a.tight_limits = "on"
```

```
a.data_bounds = [0,0;5e-6,1e4]
```

```
subplot(2, 2, 2);
```

```
plot(L', E2);
```

```
xlabel("wavelength");
```

```
ylabel("Energy");
```

```
title("Plancks distribution");
```

```
subplot(2, 2, 3);
```

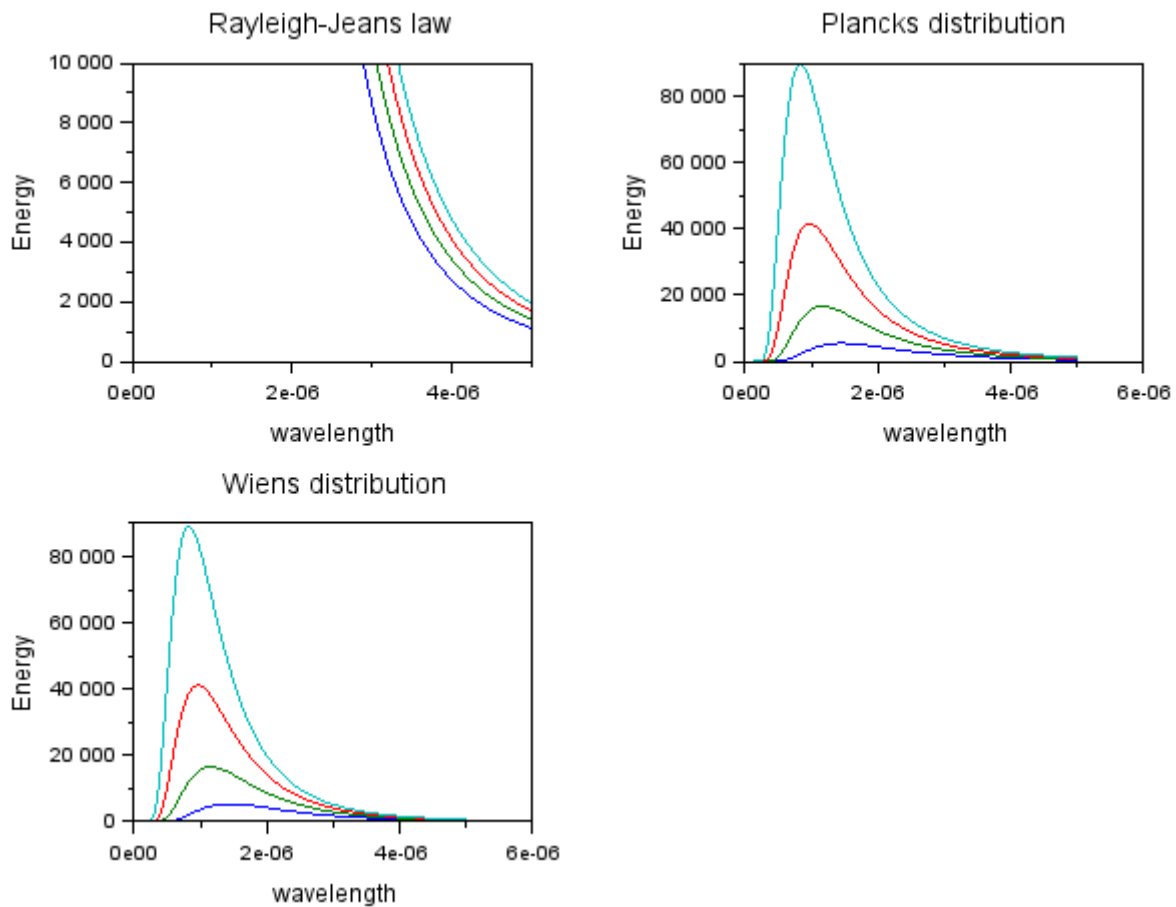
```
plot(L', E3);
```

```
xlabel("wavelength");
```

```
ylabel("Energy");
```

```
title("Wiens distribution");
```

GRAPH:



Q3)(B) Comparison Plots:

At any suitable, fixed temperature, simultaneously plot all three distribution laws, as a function of (a) frequency, and (b) wavelength.

Hence, by showing that the distribution curves coincide in the right regimes of frequency and wavelength, establish that the laws (i) and (ii) are limiting cases of law (iii).

CODE:

```
//Frequency distribution comparison
```

```
clc;clear;clf();
```

```
// Define constants
```

```
T = 1000; // Temperatures in Kelvin
```

```
kb = 1.38e-23; // Boltzmann constant
```

```
h = 6.6260e-34; // Planck constant
```

```
c = 3e8; // Speed of light
```

```
// Define frequency range
```

```
n = 100;
```

```
f = linspace(0, 1e14, n);
```

```
// Loop through each temperature
```

```
for i=1:n
```

```
    // Calculate Rayleigh-Jeans law
```

```
    E1(i) = (8 * %pi * kb * T * f(i)^2) / c^3;
```

```
    // Calculate Planck's law
```

```
    E2(i) = ((8 * %pi * h * f(i)^3) / c^3) * (1 / (exp(h * f(i) / (kb * T)) - 1));
```

```
    // Calculate Wien's law
```

```
    B3(i) = ((8 * %pi * h * f(i)^3) / c^3) * (1 / exp(h * f(i) / (kb * T)));
```

```
end
```

```
// Plot the results
```

```
plot(f', E1,"-*g");
```

```
plot(f', E2,"-*b");
```

```
plot(f', B3,"-k");
```

```
xlabel("frequency");
```

```
ylabel("Energy");
```

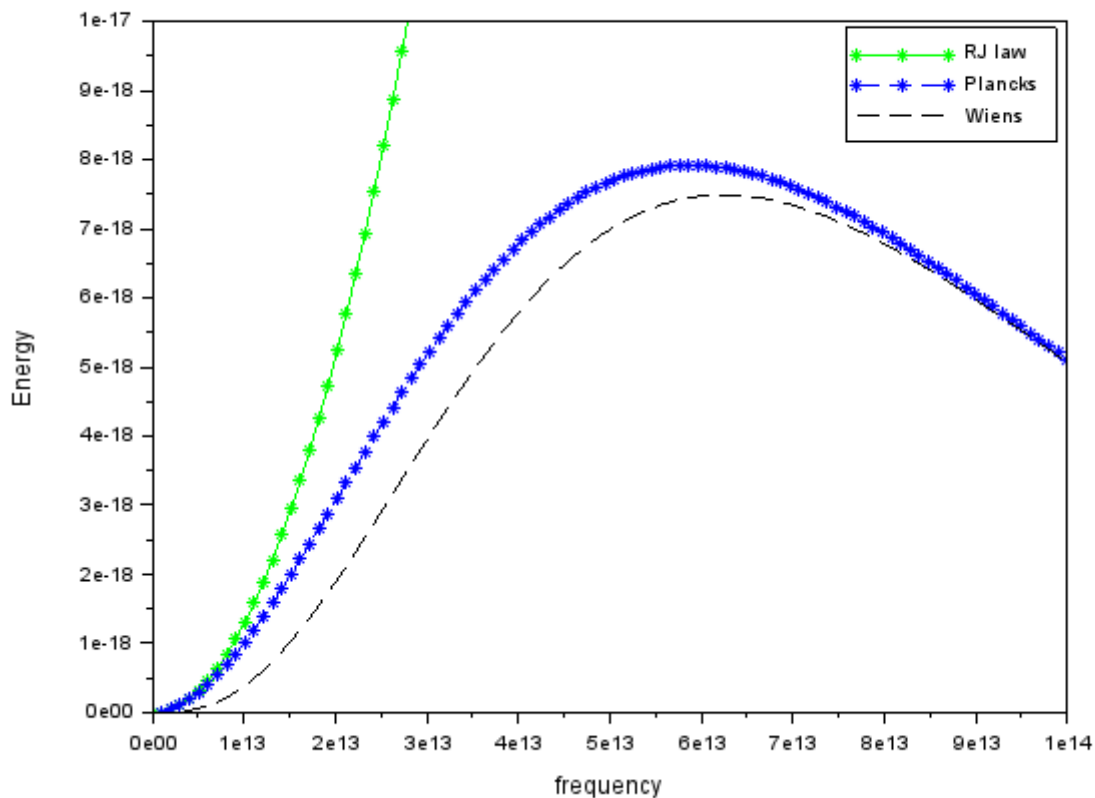
```
a = gca()
```

```
a.tight_limits = "on"
```

```
a.data_bounds = [0,0;1e14,1e-17]
```

```
legend("RJ law","Plancks","Wiens")
```

GRAPH:



//wavelength distribution comparision

```
clc;clear;clf();
```

```
// Define constants
```

```
T = 1000;           // Temperatures in Kelvin
```

```
kb = 1.38e-23;      // Boltzmann constant
```

```
h = 6.626e-34;      // Planck constant
```

```
c = 3e8;            // Speed of light
```

```
// Define wavelength range
```

```
n = 500;
```

```
L = linspace(0, 3e-5, n)
```

```
// Loop through each temperature
```

```
for i = 1:n
```

```
    E1(i) = [8 * %pi * kb * T] / L(i)^4;
```

```
    E2(i) = [(8 * %pi * h * c) / L(i)^5] * (1 / (exp(h * c / (kb * T * L(i))) - 1));
```

```
    E3(i) = [(8 * %pi * h * c) / L(i)^5] * (1 / exp(h * c / (kb * T * L(i))));
```

```
end
```

```
//Plot
```

```
plot(L, E1, "b");
```

```
plot(L, E2, "--k");
```

```
plot(L, E3, "*-g");
```

```
xlabel("wavelength");
```

```
ylabel("Energy");
```

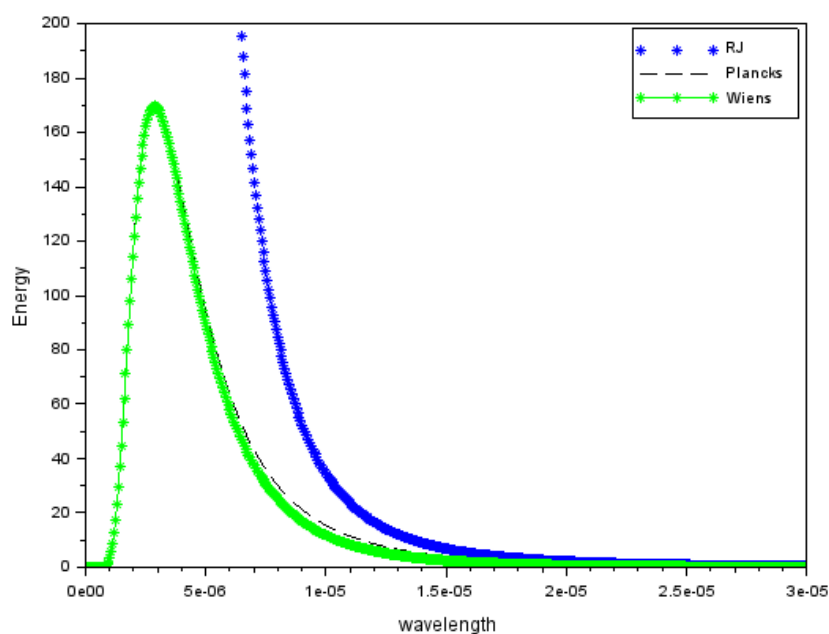
```
a = gca()
```

```
a.tight_limits = "on"
```

```
a.data_bounds = [0,0;3e-5,200]
```

```
legend("RJ","Plancks","Wiens")
```

GRAPH:



Q3)(C) Wien's Displacement Law and Wien's Constant Determination:

Compute the Planck's distribution at several different temperatures. In each case, evaluate the wavelength at which the distribution peaks. From this data, verify the Wien's Displacement Law, and determine the value of Wien's constant.

USE THESE METHODS:

(a) `reglin`

(b) least square fitting.

CODE:

```
clc;clear;clf();

Kb=1.38e-23;
c=3e8
h=6.626e-34
n = 650

L=linspace(1e-10,1e-5,n) //wavelength range
T=[1000,2000,3000,4000]

for j=1:4 //loop for temprature
    for i=1:n //loop for frequency
        E(i,j)=[(8*(%pi)*h*c)/L(i)^5]*(1/(exp(h*c/(Kb*T(j)*L(i)))-1))
    end

    [a,b]=max(E(:,j))
    L1(j)=L(b)
end

x=1./T
plot(x',L1,'r')
[m,c]=reglin(x,L1') //m = slope, c = constant

//Plot
plot(x,m.*x+c) // lambda = b/T
xlabel("1/T")
ylabel("lambda")
title("Weins constant by slope: "+string(m))

disp("WEINS DISPLACEMENT CONSTANT: "+string(m))

//Least square fitting

m1=length(x)
x1=sum(x)
x2=sum(L1)
x3=sum(x*L1)
x4=sum(x^2)
A=[x1,m1;x4,x1]
B=[x2;x3]
```

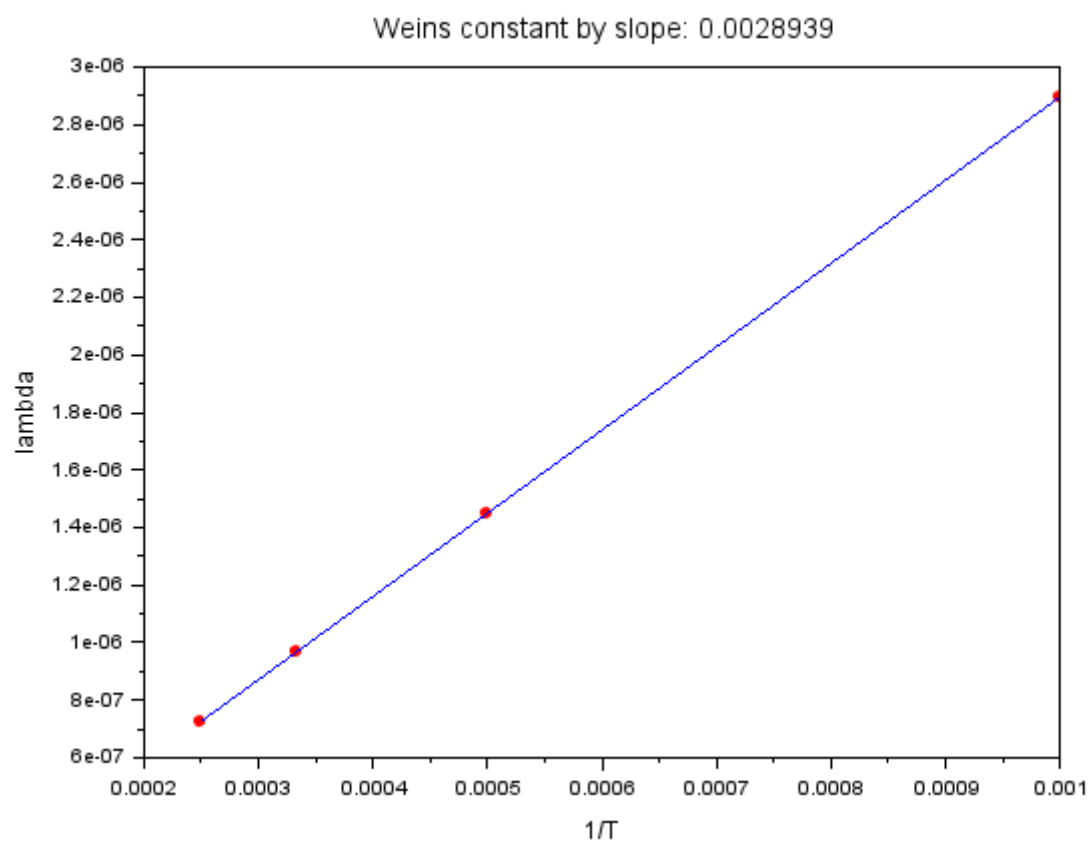
```
X=inv(A)*B  
disp("Weins constant using LSFM : "+ string(X(1)))
```

OUTPUT:

"WEINS DISPLACEMENT CONSTANT: 0.0028939"

"Weins constant using LSFM : 0.0028939"

GRAPH:



3D). Stefan's Constant Determination:

Compute the Planck's distribution at several different temperatures. In each case, evaluate the total energy (all wavelengths/frequencies taken, combined), and total energy radiated out at each temperature. From this data, verify Stefan's Law, and determine the value of Stefan's constant.

Use both methods:

(a) reglin

(b) your own routine for least square fitting.

CODE:

```
//Stefan's constant
```

```
clc;clear;clf();
```

```
c = 3e8;
```

```
Kb = 1.3806*10^(-23);
```

```
h = 6.626*10^(-34);
```

```
T = [1000,1500,2000,2500,4000];
```

```
x = T.^4;
```

```
n = 500;
```

```
L = linspace(1e-10,1e-4,n);
```

```
for j=1:length(T)
```

```
    for i=1:n
```

```
        E1(i,j)=[(8*(%pi)*h*c)/L(i)^5]*(1/(exp(h*c/(Kb*T(j)*L(i)))-1))
```

```
    end
```

```
    Et(j)=inttrap(L,E1(:,j)) //Total Energy
```

```
    Er(j)=(c/4)*Et(j)
```

```
end
```

```
plot(L,E1(:,1))
```

```
[m,c1]=reglin(x,Er')
```

```
//pLOT
```

```
plot(x,((m*x)+c1),'-k') // Er = sigma * T^4
```

```
xlabel("T^4")
```

```
ylabel("Rate of energy")
```

```
title("Stefans constant")
```

```
disp("Stefan constant by using reglin = "+string(m))
```

```
//Least sqaure fitting method
```

```
x2 = (x(j))^2
```

```
xy = x(j).*Er(j)
```

```
x1=sum(x)
```

```
y1=sum(Er)
```

```
x21=sum(x2)
```

```
xy1=sum(xy)
```

```

A=[x1,length(x);x21,x1]
B=[y1;xy1]
X=inv(A)*B
plot(x,((X(1).*x)+X(2)),'*k')
disp("Stefan constant using LSTM = "+string(X(1)))

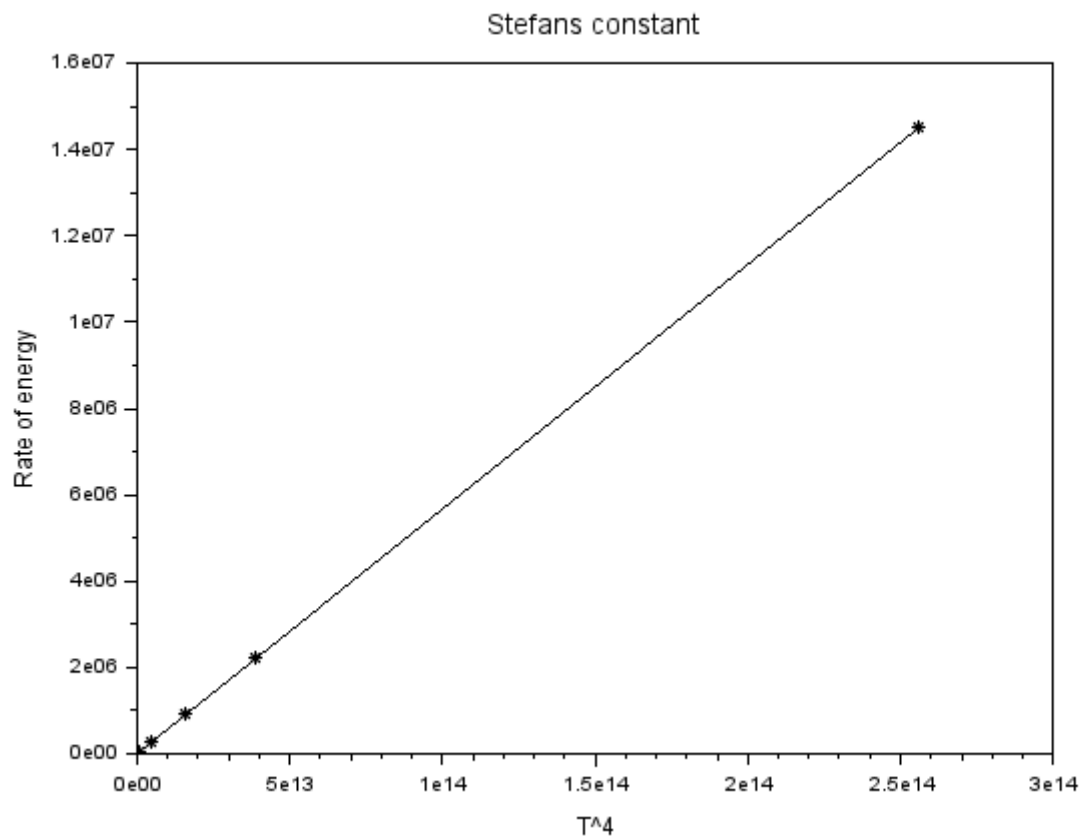
```

OUTPUT:

"Stefan constant by using reglin = 5.667D-08"

"Stefan constant using LSTM = 5.667D-08"

GRAPH:



Speed Distribution

Q 4 For the Maxwellian Speed Distribution:

(a) Compute the average speed, root mean square and most probable speed, for any choice of gas. Compare the results of this computation against the theoretical expressions for these.

(b) For any gas (i.e. m = fixed), compute and plot the distribution function at (at least three) different temperatures. What inference can we draw from this?

Code:

```
//Speed boltzmann distribution
```

```
clc;clear;
```

```
N = 6e23;          // avogadro number
M = 48;            // molecular mass of O3 in g/mole
m = M/(1000*N);    // mass in kg
k = 1.38e-23;      // boltzmann constant
```

```
n=2000;           // number of intervals
V = linspace(0,2500,n); // range of speed
T = [600,900,1500];
```

```
for i=1:length(T);
```

```
    function y=f(v)
        y = 4*%pi*(v^3)*((m/(2*%pi*k*T(i)))^(3/2))*exp(-(m*v^2)/(2*k*T(i)));
    endfunction
```

```
    disp("_____Average Speed _____");
    v_avg=intg(0,2500,f);
    vavg=sqrt((8*k*T(i))/(%pi*m))
    disp("Theoretical value = "+string(vavg)+" m/s")
    disp("Calculated value = "+string(v_avg)+"m/s");
```

```
    function y=f1(v)
        y = 4*%pi*(v^4)*((m/(2*%pi*k*T(i)))^(3/2))*exp((-m*v^2)/(2*k*T(i)))
    endfunction
```

```
    disp("_____RMS speed_____")
    v_rms=intg(0,2500,f1)
    vrms=sqrt(3*k*T(i)/m)
    disp("Theoretical value = "+string(vrms)+" m/s")
    disp("Calculated value = "+string(sqrt(v_rms))+ " m/s")
```

```
    for j=1:n
        E(i,j) = 4*%pi*(V(j)^2)*((m/(2*%pi*k*T(i)))^(3/2))*exp(-(m*V(j)^2)/(2*k*T(i)));
    end
    disp("_____Most probable speed_____")
    [a,b]=max(E(i,:))
```

```

Vmp=sqrt((2*k*T(i))/m)
disp("Theoretical value = "+string(Vmp)+" m/s")
disp("Calculated value = "+string(V(b))+" m/s")
end

```

```

plot(V',E',linewidth,2);
xlabel("Speed")
ylabel("f(v)")
legend("600","900","1500")

```

OUTPUT:

" _____Average Speed _____ "

"Theoretical value = 513.38152 m/s"

"Calculated value = 513.38152m/s"

" _____RMS speed_____ "

"Theoretical value = 557.22527 m/s"

"Calculated value = 557.22527 m/s"

" _____Most probable speed_____ "

"Theoretical value = 454.97253 m/s"

"Calculated value = 455.22761 m/s"

" _____Average Speed _____ "

"Theoretical value = 628.76138 m/s"

"Calculated value = 628.76136m/s"

" _____RMS speed_____ "

"Theoretical value = 682.45879 m/s"

"Calculated value = 682.45874 m/s"

" _____Most probable speed_____ "

"Theoretical value = 557.22527 m/s"

"Calculated value = 557.77889 m/s"

" _____Average Speed _____ "

"Theoretical value = 811.72746 m/s"

"Calculated value = 811.66709m/s"

" _____RMS speed_____ "

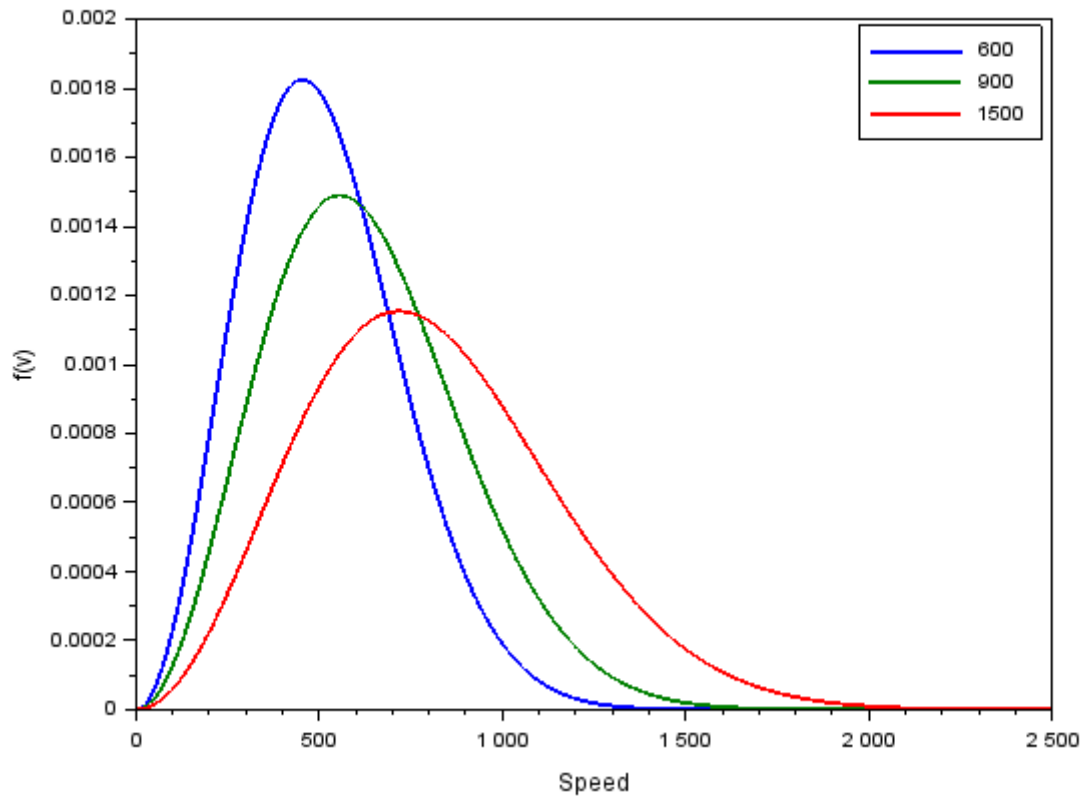
"Theoretical value = 881.05051 m/s"

"Calculated value = 880.96119 m/s"

"_____Most probable speed_____"

"Theoretical value = 719.37473 m/s"

"Calculated value = 719.10955 m/s"



(c) At a fixed temperature, compute and plot the distribution function for (at least three) different gases. What inference can we draw from this?

CODE:

```
//Speed boltzmann distribution
```

```
clc;clear;clf;
```

```
k = 1.38e-23    // boltzmann constant
```

```
N = 6e23        // avogadro numbers
```

```
n = 2000        // number of intervals
```

```
V = linspace(0,2500,n) // range of speed
```

```
T = 1000        // for fix temperature
```

```
M=[4,20,40,132] //He,Ne,Ar,Xe
```

```
for i=1:length(M)
```

```

m(i)=M(i)/(1000*N) // Mass of gases in kg
c(i)=m(i)/(2*k*T)

for j=1:n
    E(j,i)=4*pi*((c(i)/pi)^1.5)*(V(j)^2)*exp(-c(i)*(V(j)^2))
end
end

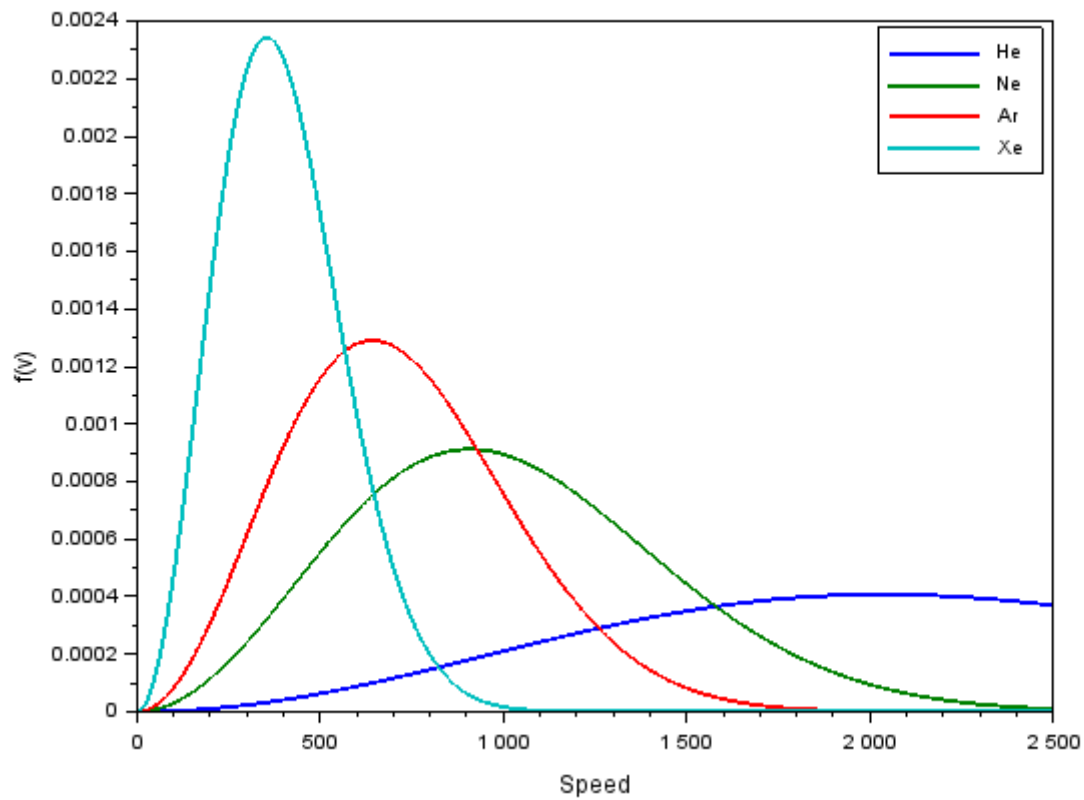
```

```

// Plot
plot(V',E,'linewidth',2)
xlabel("Speed")
ylabel("f(v)")
legend("He","Ne","Ar","Xe")

```

OUTPUT:



Velocity Distribution

Q4) For the Maxwellian Velocity Distribution

(for any individual component of velocity, say v_x):

(a) Compute the average speed, root mean square and most probable values of the x-component of velocity, for any choice of gas. Compare the results of this computation against what is expected on theoretical grounds.

CODE:

```
// Velocity Distribution

clc;clear;

N=6e23;    // avogadro number
M=32;      // molar mass of O2
m=M/(1000*N); // Mass in kg
k=1.38e-23; // boltzmann constant

n=1001;    // number of intervals
V=linspace(-2000,2000,n); // range of speed
T=[1000];

disp(" Velocity for x component:- ")

for i=1:length(T);
    function y=f(v)
        y=((m/(2*(%pi)*k*T(i)))^(1/2))*v*exp((-m*v^2)/(2*k*T(i)));
    endfunction

    for j=1:n
        E(j,i)=(((m/(2*k*T(i)))/%pi)^0.5)*V(j)*exp(-(m/(2*k*T(i))))
    end

    disp("____Average Velocity ____");
    disp("Using inttrap = "+string(inttrap(V,E)))
    [v_avg,err]=intg(-70,70,f);
    disp("calculated value using intg = "+string(v_avg)+string(" m/s"));
    disp("Theoretical value = "+string(0)+string(" m/s"))

    function y=f1(v)
        y=((m/(2*(%pi)*k*T(i)))^(0.5))*v^2*exp((-m*v^2)/(2*k*T(i)))
    endfunction

    v_rms=intg(-2000,2000,f1)
    disp("____RMS Velocity____")
    vrms=sqrt(k*T(i)/m)
    disp("Calculated value = "+string(sqrt(v_rms))+string(" m/s"))
    disp("Theoretical value = "+string(vrms)+string(" m/s"))

    for j=1:n
```

```

    E(j,i)=(((m/(2*k*T(i)))/%pi)^0.5)*exp(-(m/(2*k*T(i)))*(V(j))^2)
end

disp("____Most probable velocity____")
[a,b]=max(E(:,i))
disp("calculated value = "+string(sqrt(V(b)))+string(" m/s"))
disp("Theoretical value = "+string(0)+string(" m/s"))
end

```

OUTPUT:

```

" Velocity for x component:- "

"____Average Velocity ____"

"Using inttrap = 5.596D-14"
"calculated value using intg = 0 m/s"
"Theoretical value = 0 m/s"

"____RMS Velocity____"

"Calculated value = 508.30236 m/s"
"Theoretical value = 508.67475 m/s"

"____Most probable velocity____"

"calculated value = 0 m/s"
"Theoretical value = 0 m/s"

```

(b) For any gas (i.e. m = fixed), compute and plot the distribution function at (at least three) different temperatures. What inference can we draw from this?

CODE:

```

clc;clear;clf;

N=6e23;    // avogadro numbers
M=32;      // molar mass of oxygen
m=M/(1000*N); //molar mass into
k=1.38e-23; // boltzmann constant

n=1000;    // number of intervals
V=linspace(-2500,2500,n); // range of speed
T=[600,900,1300]; // different temperature for single gas

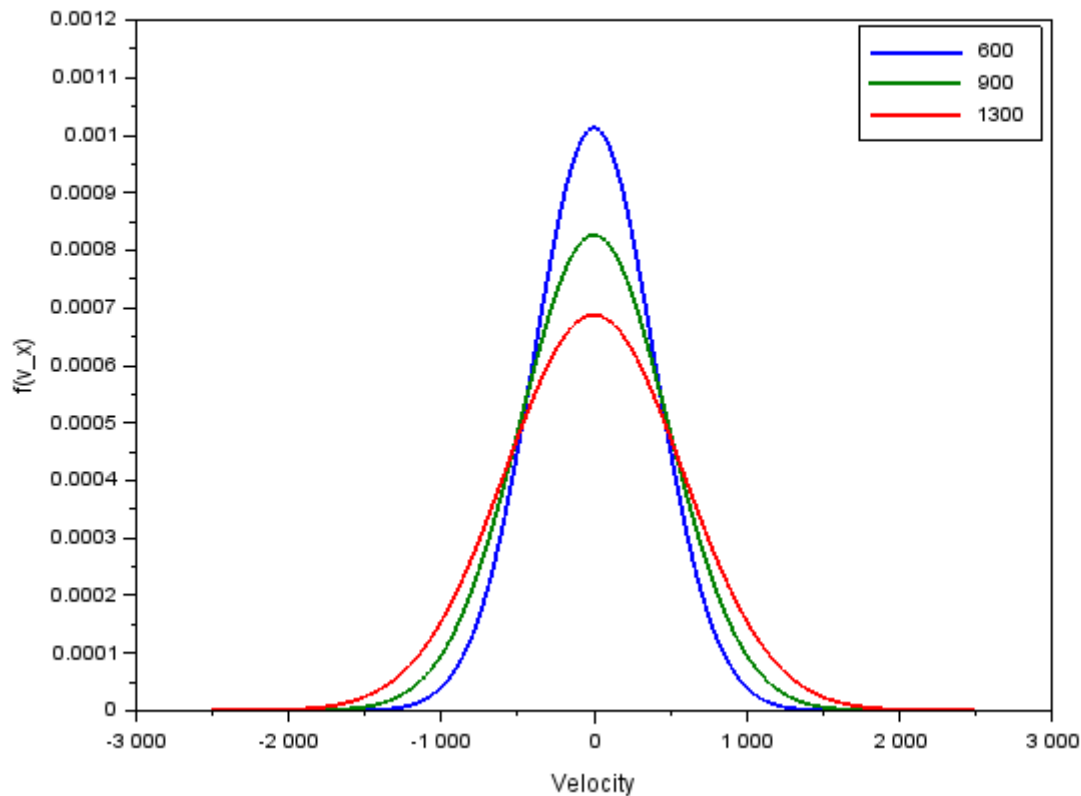
for i=1:length(T) // loop for temperature
    c=m/(2*k*T(i));
    for j=1:n // loop for speed
        E(j,i)=((c/%pi)^0.5)*exp(-c*(V(j))^2)
    end
end

```

end

```
plot(V,E,'linewidth',2)
xlabel("Velocity")
ylabel("f(v_x)")
legend("600","900","1300")
```

OUTPUT:



(c) At a fixed temperature, compute and plot the distribution function for (at least three) different gases. What inference can we draw from this?

CODE:

```
clc;clear;clf;

k=1.38e-23 // boltzmann constant
N=6e23     // avogadro numbers

n=2001     // number of intervals
V=linspace(-2500,2500,n) // range of speed
T=500      // for fix temperature
M=[4,20,40,132] //He,Ne,Ar,Xe

for i=1:length(M)
```

```

m(i)=M(i)/(1000*N) //molar mass into
c(i)=m(i)/(2*k*T)
for j=1:n
    E(j,i)=((c(i)/%pi)^0.5)*exp(-c(i)*(V(j))^2)
end
end

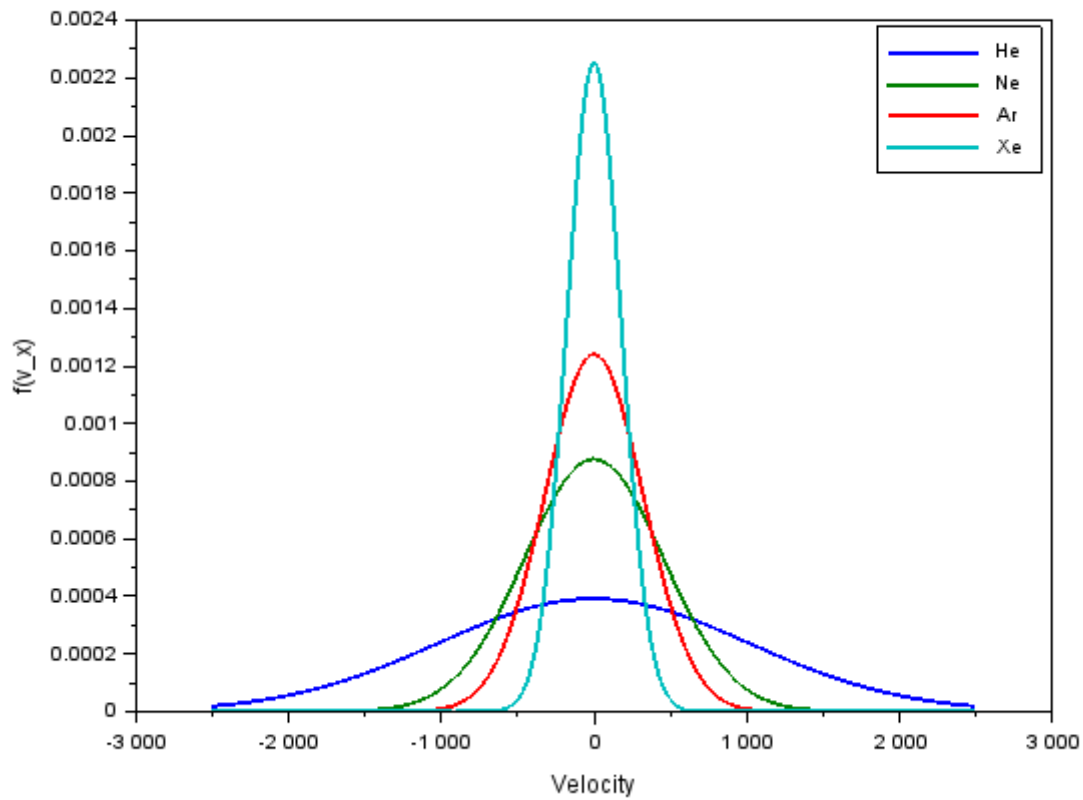
```

```

plot(V,E,'linewidth',2)
xlabel("Velocity")
ylabel("f(v_x)")
legend("He","Ne","Ar","Xe")

```

OUTPUT:



Q5) Comparison of Statistical Distributions, and Effect of Temperature:

Consider each of the following statistical distributions:

- a) Maxwell-Boltzmann Distribution**
- b) Fermi-Dirac Distribution**
- c) Bose-Einstein Distribution**

Code:

```
clc;
clear;
clf();

//Define constants

e = 1.6e-19;
k = 1.38e-23;
u = 0;
E = -0.5: 0.001: 0.5
T = [100,900,1300]
Tc = 900 // Temperature for comparison
a = -1
dist = ["BE","MB","FD"]

for n = 1:3
    for j = 1:length(T)
        if Tc == T(j)
            q = j
        end

        for i = 1:length(E)
            f(j,i) = 1/(exp(((E(i)-u)*e)/(k*T(j)))+a)
        end
    end

    a = a+1;

// Plot of distribution for different temperatures

    subplot(2,2,n)
    plot(E,f')
    xlabel("E")
    ylabel("f(E)")
    legend("T = " + string(T)+"K")
    title(string(dist(n)))

    C(n,:) = f(q,:);
    p = gca()
    p.data_bounds = [-0.5,0;0.5,2]
end

// Plot for comparison of the distributions
```

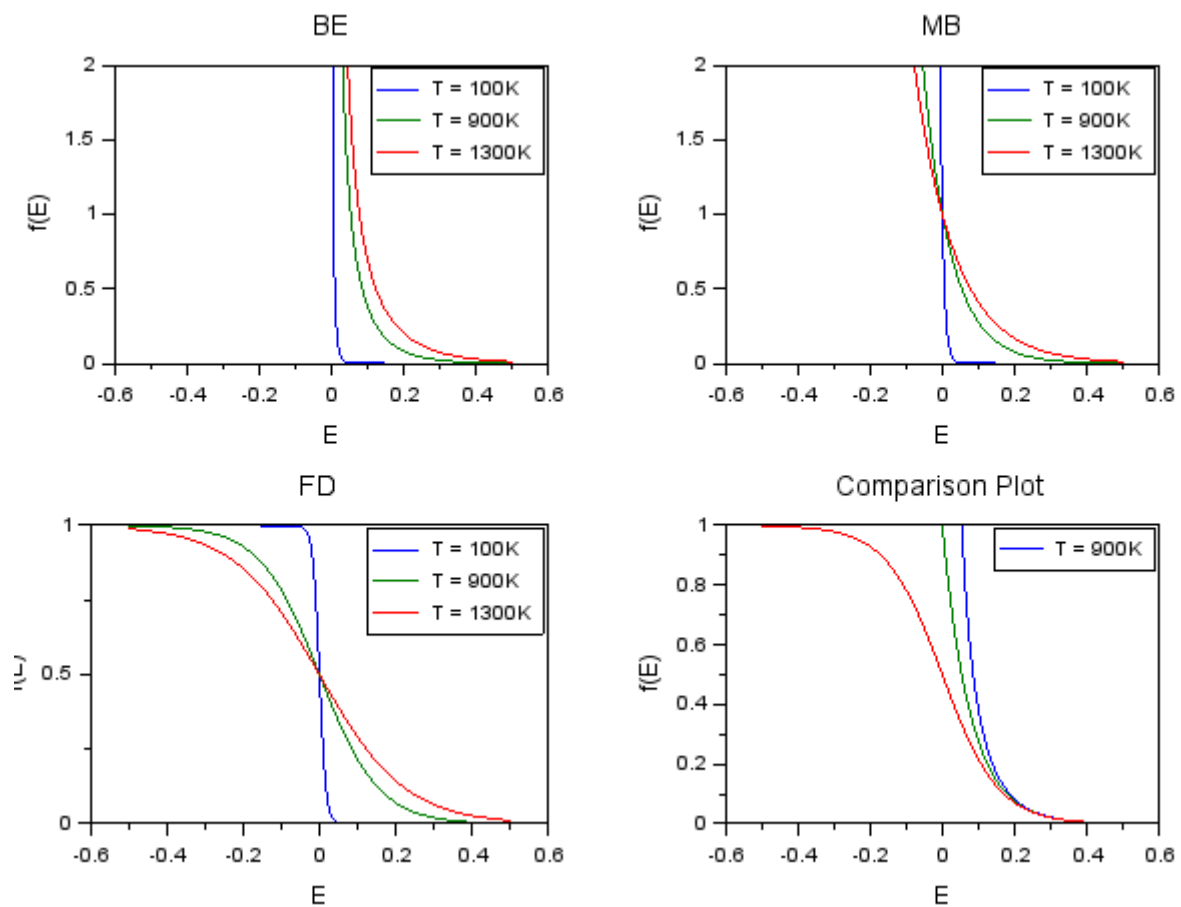
```

subplot(2,2,4)
plot(E',C')
xlabel("E")
ylabel("f(E)")

title("Comparison Plot")
legend("T = "+string(Tc)+"K")
p = gca()
p.data_bounds = [-0.5,0;0.5,1]

```

Output:



Q7) Energy Distribution of Bosons and Fermions: Consider the following four distinct combinations for both high and low temperature:

- 1. Non-relativistic Fermions**
- 2. Non-relativistic Bosons**
- 3. Relativistic Fermions**
- 4. Relativistic Bosons**

1) CODE:

```
clc;clear;clf;

e = 1.6e-19 // Charge of an electron(C)
kb = 1.38e-23 // Boltzmann constant in J/K
h = 6.626e-34 // Planck's constant in J/s
u = 1 // Chemical potential
V = 1 // Volume of the system
m = 9.1e-31 // Mass of an electron(Kg)

// Energy range in electron volts
E = 0: 0.001:2
// Temperatures in Kelvin
T = [100,900,2000]

Cn = 2*(2*3.14*V*(2*m)^1.5)/h^3; // (2*s + 1), s = 1/2

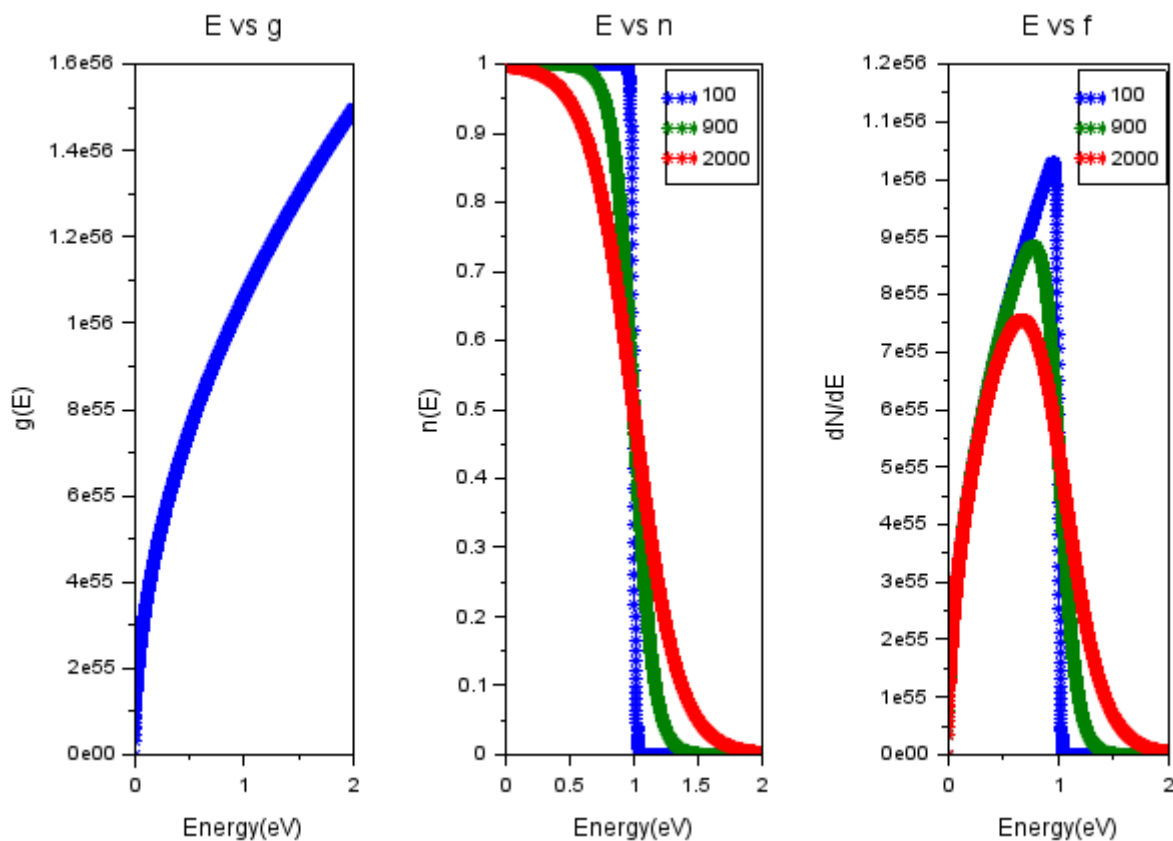
for j = 1:length(T)
    b = 1/(kb*T(j));
    for i = 1: length(E)
        // The density of accessible energy state per unit volume
        g(i) = Cn *(E(i))^0.5; // independent of temp
        // The density of occupied energy state per unit volume
        n(j,i) = 1/(exp((E(i) - u)*e*b)+1)
        // The probability that an accessible energy state is occupied by a particle
        f(j,i) = g(i) * n(j,i)
    end
end

// Plotting the density of accessible energy states
subplot(1,3,1)
plot(E',g, "b")
ylabel("g(E)")
xlabel("Energy(eV)")
title("E vs g")

// Plotting the density of occupied energy states
subplot(1,3,2)
plot(E',n, "b")
ylabel("n(E)")
xlabel("Energy(eV)")
title("E vs n")
legend("100","900","2000")
```

```
// Plotting the probability that an accessible energy state is occupied by a particle
subplot(1,3,3)
plot(E',f',"*")
ylabel("dN/dE")
xlabel("Energy(eV)")
title("E vs f")
legend("100","900","2000")
```

GRAPH:



2) CODE:

```
// Non relativistic bosons
```

```
clc;clear;clf;
```

```
// Define constants
```

```
e = 1.6e-19 // Charge of an electron
```

```
kb = 1.38e-23 // Boltzmann constant
```

```
h = 6.626e-34 // Planck's constant
```

```
s = 1 // Spin quantum number
```

```
u = -1 // Chemical potential
```

```
V = 1 // Volume
```

```
m = 9.1e-31 // Mass of an electron
```

```

// Define energy range
E = 0: 0.001:0.5

// Define temperature values
T = [100,1000]

Cn = (2*s + 1)*(2*3.14*V*(2*m)^1.5)/h^3;

// Loop over temperature values
for j = 1:length(T)
    b = 1/(kb*T(j));
    for i = 1: length(E)
        // Calculate the density of states
        g(i) = Cn *(E(i))^0.5;
        // Calculate the distribution function
        n(j,i) = 1/(exp((E(i) - u)*e*b)-1)
        // Calculate the number of particles
        f(j,i) = g(i) * n(j,i)
    end

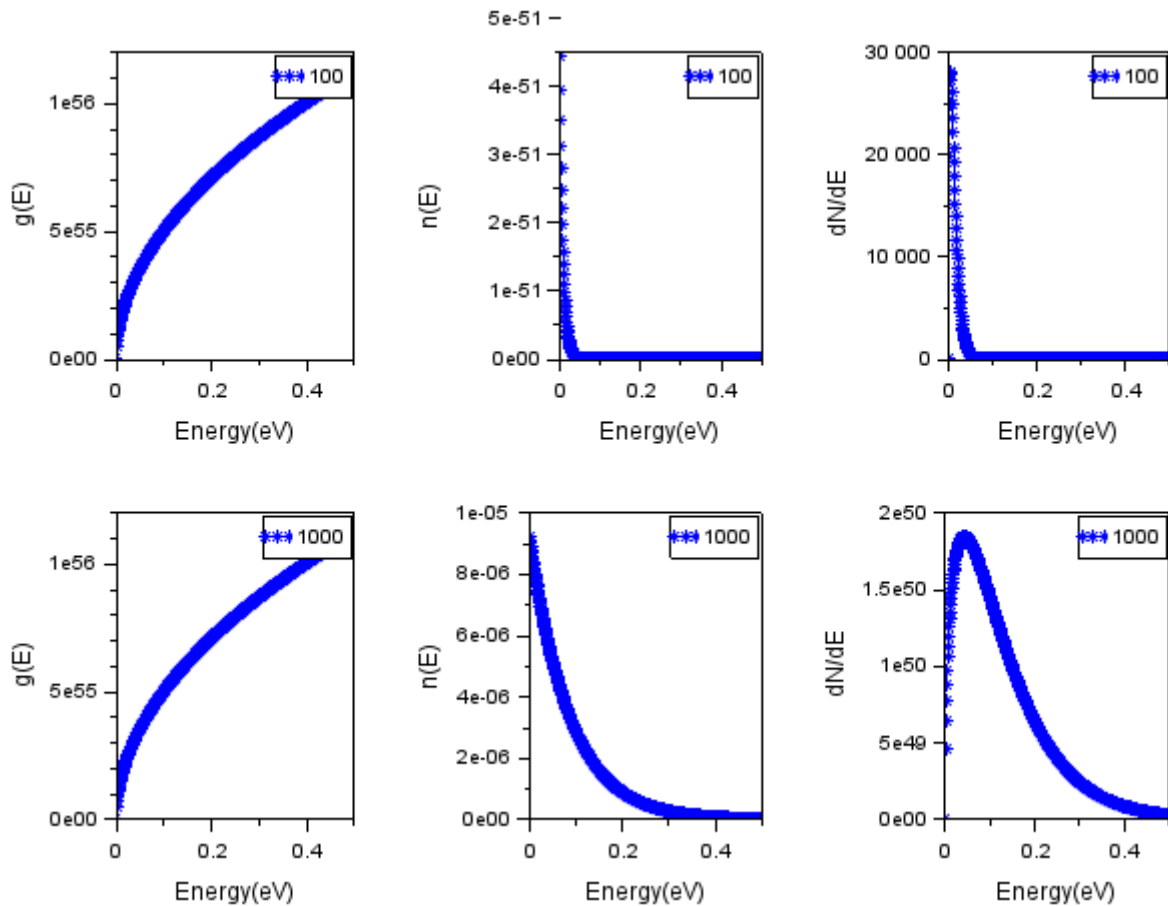
    // Plot the density of states
    subplot(2,3,j*j)
    plot(E',g,"*")
    ylabel("g(E)")
    xlabel("Energy(eV)")
    legend(string(T(j)))

    // Plot the distribution function
    subplot(2,3,j*j+1)
    plot(E',n(j,:),"*")
    ylabel("n(E)")
    xlabel("Energy(eV)")
    legend(string(T(j)))

    // Plot the number of particles
    subplot(2,3,j*j+2)
    plot(E',f(j,:),"*")
    ylabel("dN/dE")
    xlabel("Energy(eV)")
    legend(string(T(j)))
end

```

GRAPH:



3) CODE:

// Relativistic fermions

clc;clear;clf;

e = 1.6e-19; // charge of an electron
kb = 1.38e-23; // Boltzmann constant
h = 6.626e-34; // Planck constant
s = 0.5; // spin
u = 1; // chemical potential
V = 1; // volume
c = 3e8; // speed of light

E = 0: 0.001:2; // energy range
T = [10^8,10^9]; // temperature range

Cr = (2*s*4*3.14*V)*(h^3*c^3); // density of states constant

```
for j = 1:length(T)
    b = 1/(kb*T(j));
    for i = 1:length(E)
        g(i) = Cr *(E(i))^2; // density of states
        n(j,i) = 1/(exp((E(i) - u)*10^6*e*b)+1); // Fermi-Dirac distribution
    end
end
```

```

    f(j,i) = g(i) * n(j,i); // number of fermions
end
end

```

```

subplot(1,3,1)
plot(E',g,"*")
ylabel("g(E)")
xlabel("Energy(MeV)")
title("E vs g")

```

```

subplot(1,3,2)
plot(E',n',"*")
ylabel("n(E)")
xlabel("Energy(MeV)")
title("E vs n")
legend("10^8","10^9")

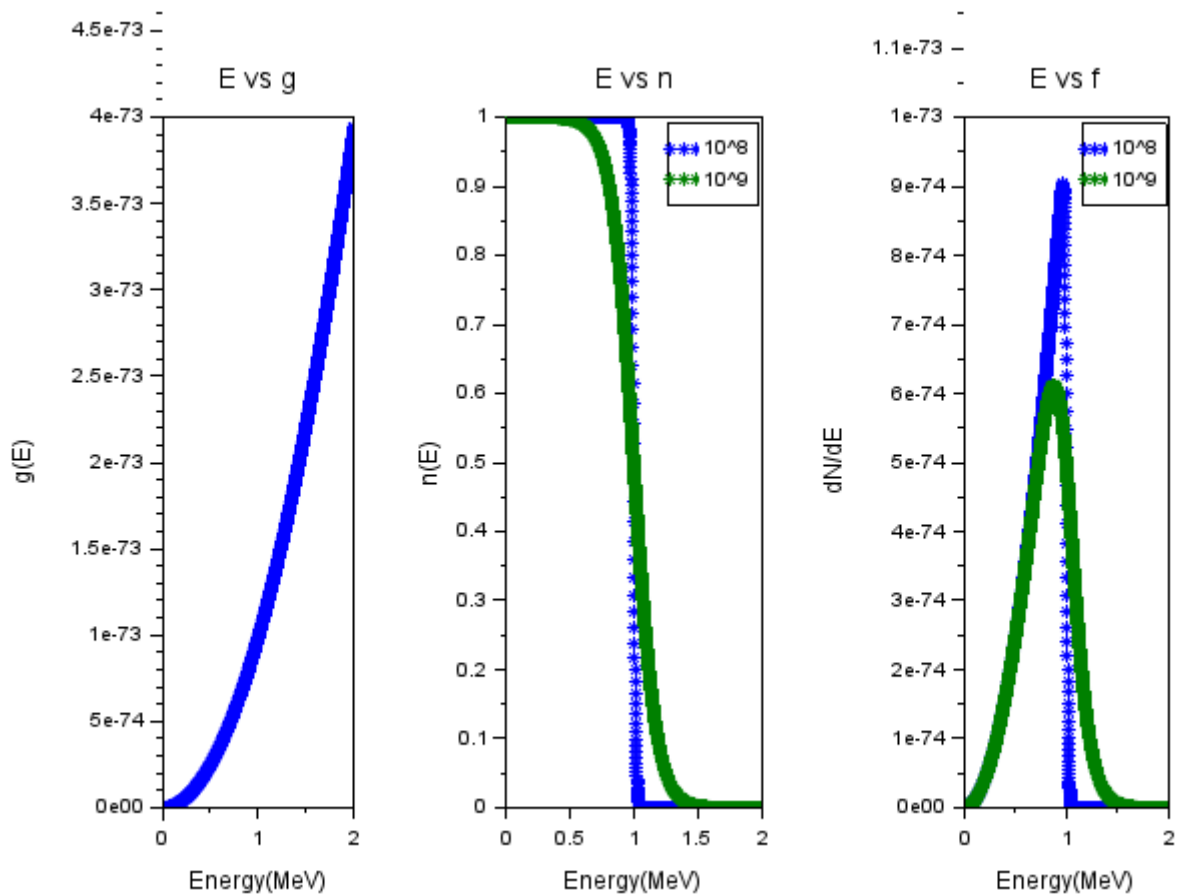
```

```

subplot(1,3,3)
plot(E',f',"*")
ylabel("dN/dE")
xlabel("Energy(MeV)")
title("E vs f")
legend("10^8","10^9")

```

GRAPH:



4) CODE:

// Relativistic bosons

clc;clear;clf;

e = 1.6e-19; // charge of an electron
kb = 1.38e-23; // Boltzmann constant
h = 6.626e-34; // Planck constant
s = 1; // spin
u = -1; // chemical potential
V = 1; // volume
c = 3e8; // speed of light

E = 0: 0.001:6; // energy range
T = [10^9,10^10]; // temperature range

Cr = (2*s*4*3.14*V)*(h^3*c^3); // density of states constant

```
for j = 1:length(T)
    b = 1/(kb*T(j));
    for i = 1:length(E)
        g(i) = Cr *(E(i))^2; // density of states
        n(j,i) = 1/(exp((E(i) - u)*10^6*e*b)-1); // Bose-Einstein distribution
        f(j,i) = g(i) * n(j,i); // number of bosons
    end
```

```
subplot(2,3,j*j)
plot(E',g, "b")
ylabel("g(E)")
xlabel("Energy(MeV)")
legend(string(T(j)))
```

```
subplot(2,3,j*j+1)
plot(E',n(j,:)', "b")
ylabel("n(E)")
xlabel("Energy(MeV)")
legend(string(T(j)))
```

```
subplot(2,3,j*j+2)
plot(E',f(j,:)', "b")
ylabel("dN/dE")
xlabel("Energy(MeV)")
legend(string(T(j)))
end
```

GRAPH:

