Homework 4

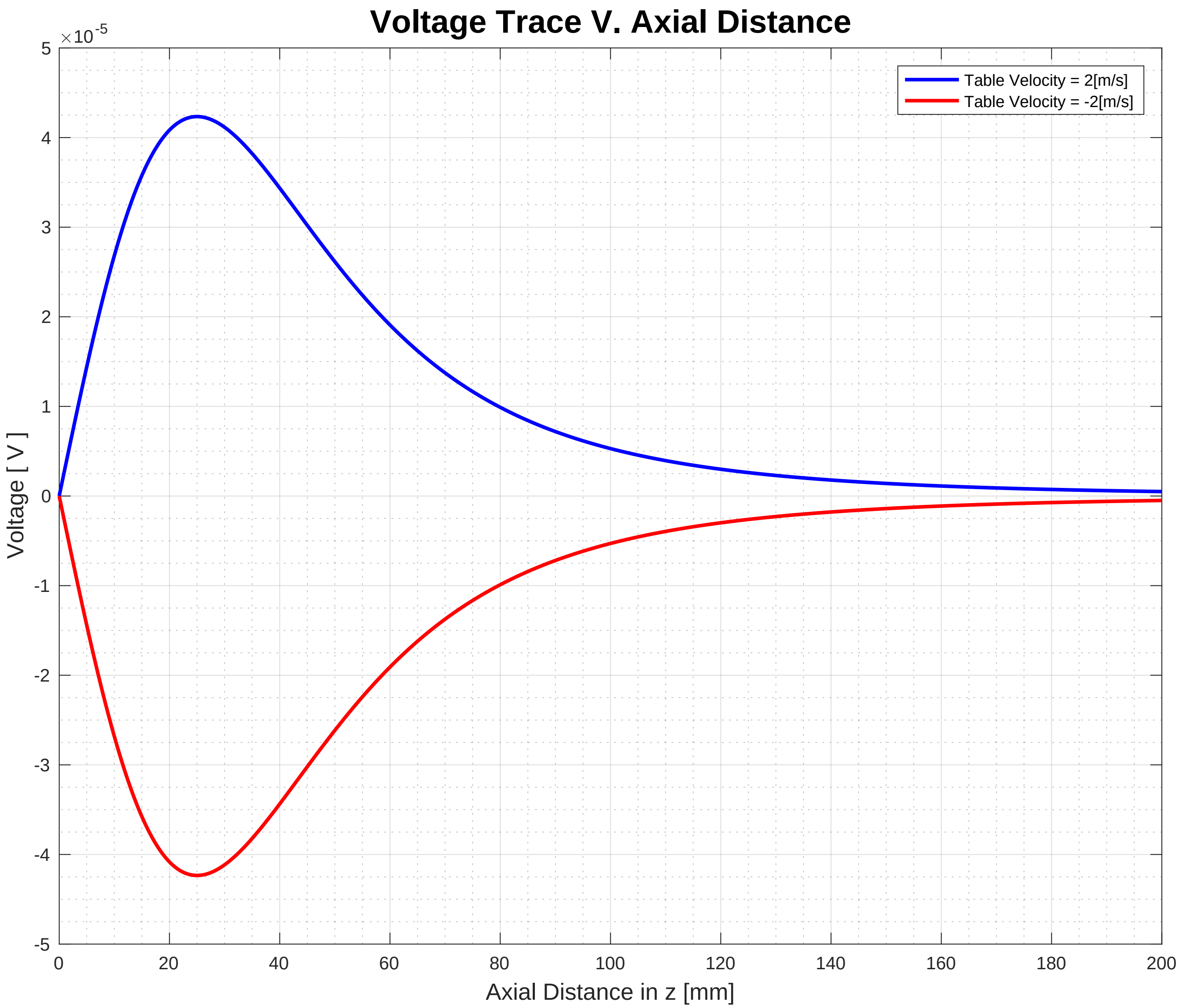
AE453 - Spring 2018

Emilio R. Gordon

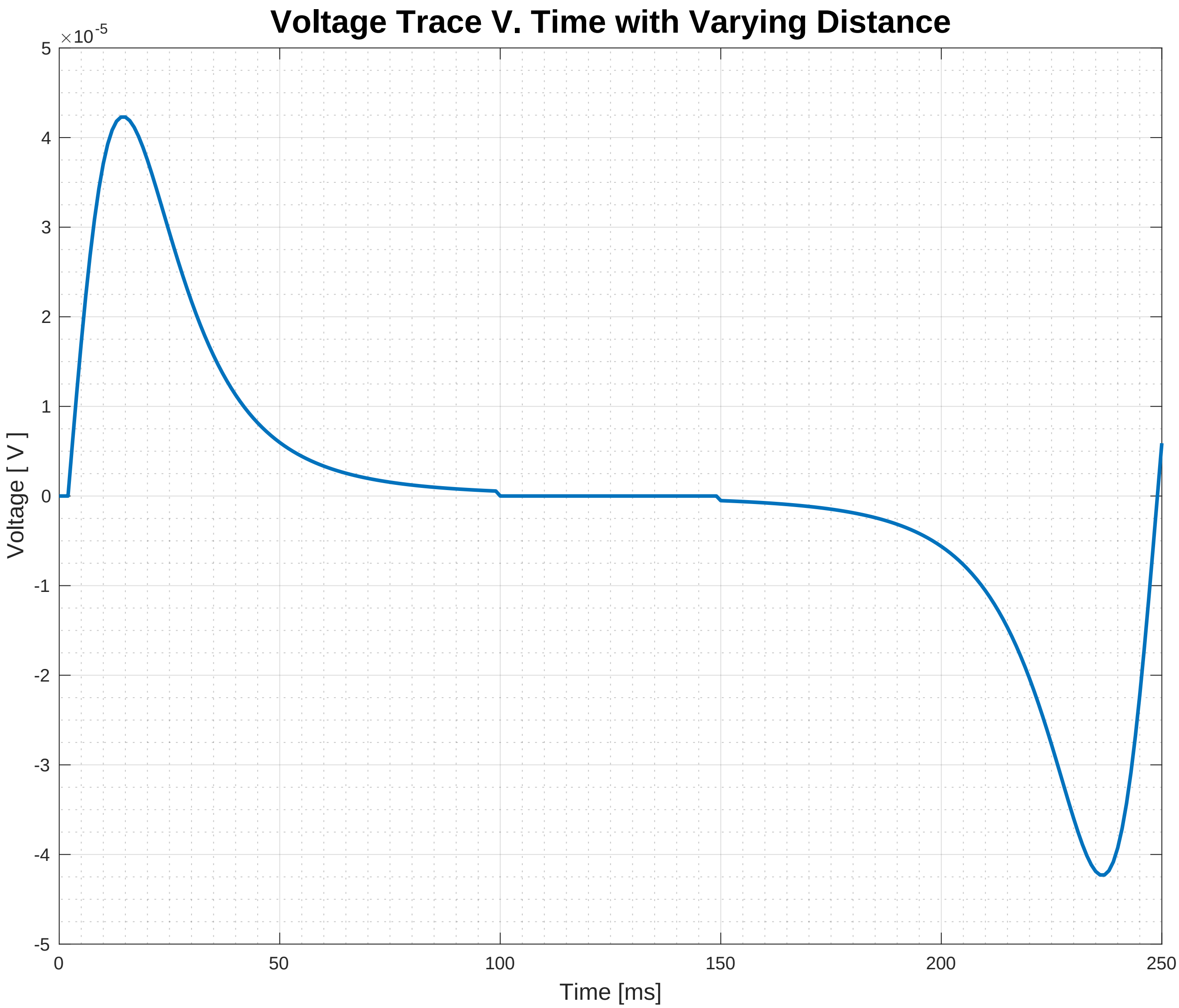
Problem 1:

```matlab
function hw4p1
t = [0:1:250]/1000; i=1; dzdt = 0; z = 0; V = 0;

while i<1+length(t)
    if i<100
        dzdt = [dzdt 2];
        z = [z zloc(dzdt(i),t(i))];
        V = [V Vz(dzdt(i),z(i))];
    elseif (i>100)&(i<151)
        dzdt = [dzdt 0];
        z = [z zloc(dzdt(100),t(100))+zloc(dzdt(i),t(i))];
        V = [V Vz(dzdt(i),z(i))];
    elseif i>150
        dzdt = [dzdt -2];
        z(150)
        zloc(dzdt(i),t(i))
        z = [z z(150)+zloc(dzdt(i),(t(i)-.15))];
        V = [V Vz(dzdt(i),z(i))];
    end
    i=i+1;
end
figure(1)
plot(t*1000,V,'LineWidth',2)

AxialDistance = [0:.0002:.200]
for j=1:length(AxialDistance)
        VV(1,j)=Vz(2,AxialDistance(j));
        VV(2,j)=Vz(-2,AxialDistance(j));
end
figure(2)
plot(AxialDistance*1000,VV(1,:),'-b','LineWidth',2)
hold on
plot(AxialDistance*1000,VV(2,:),'-r','LineWidth',2)
%Plotting code removed for space. Please see Github for full code.
end

function V = Vz(dzdt,z)
    n = 50;
    A = 1.963e-5;
    mu = 1.256e-6;
    JN = 100;
    a = 0.05;
    Constant1 = ((n*A*3*mu*JN*a^2)/2)*dzdt;
    V = Constant1*(z/((a^2+z^2)^(5/2)));
end

function z = zloc(dzdt,t)
    z = dzdt*t;
end
```

Figure 1: Problem 1: Voltage Trace Code



Voltage Trace V. Axial Distance
×10^-5
5
4
3
2
1
0
-1
-2
-3
-4
-5
Voltage [V]
0
20
40
60
80
100
120
140
160
180
200
Axial Distance in z [mm]
Table Velocity = 2[m/s]
Table Velocity = -2[m/s]

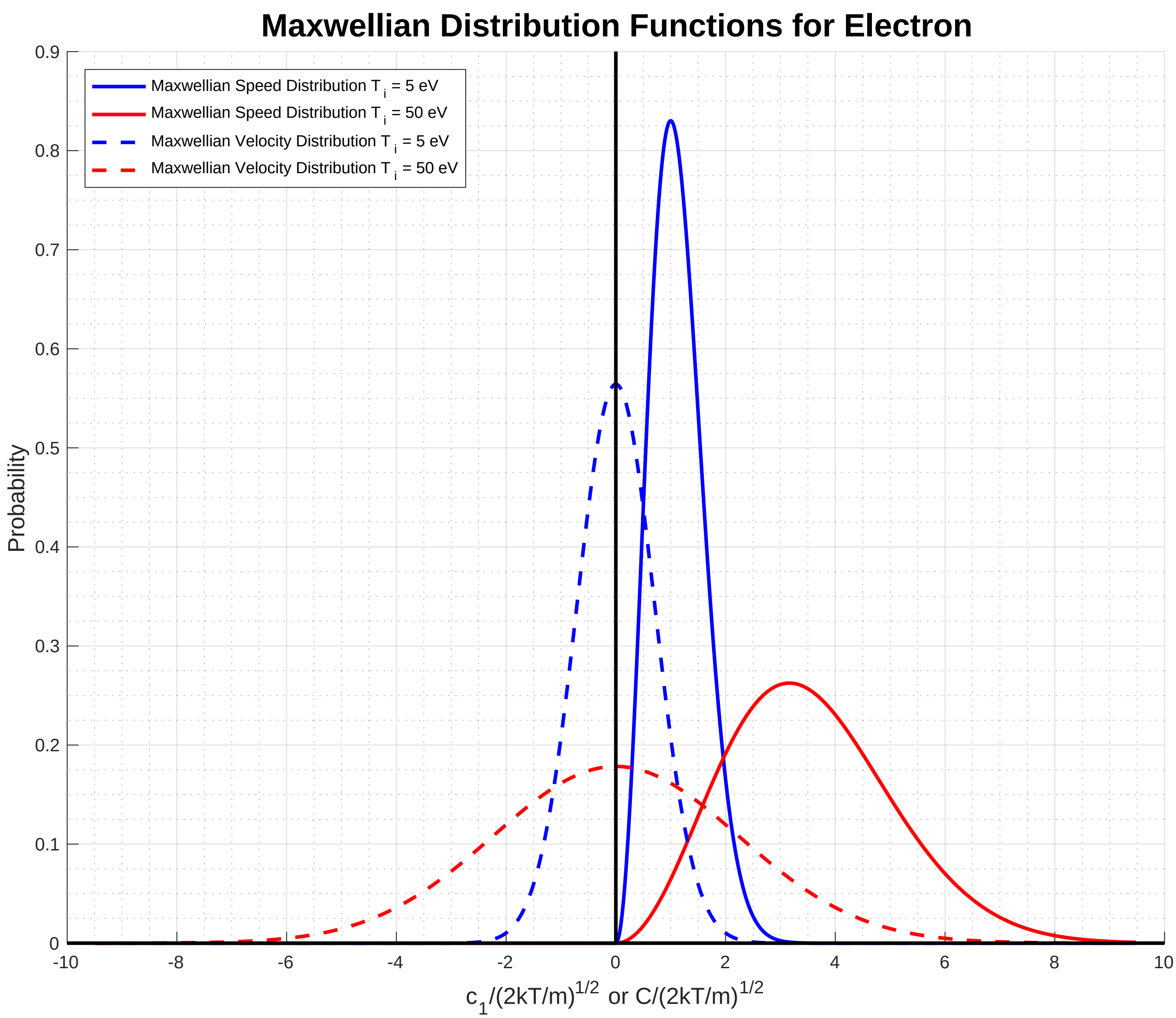


Voltage Trace V. Time with Varying Distance
×10^-5
5
4
3
2
1
0
-1
-2
-3
-4
-5
Voltage [V]
0
50
100
150
200
250
Time [ms]

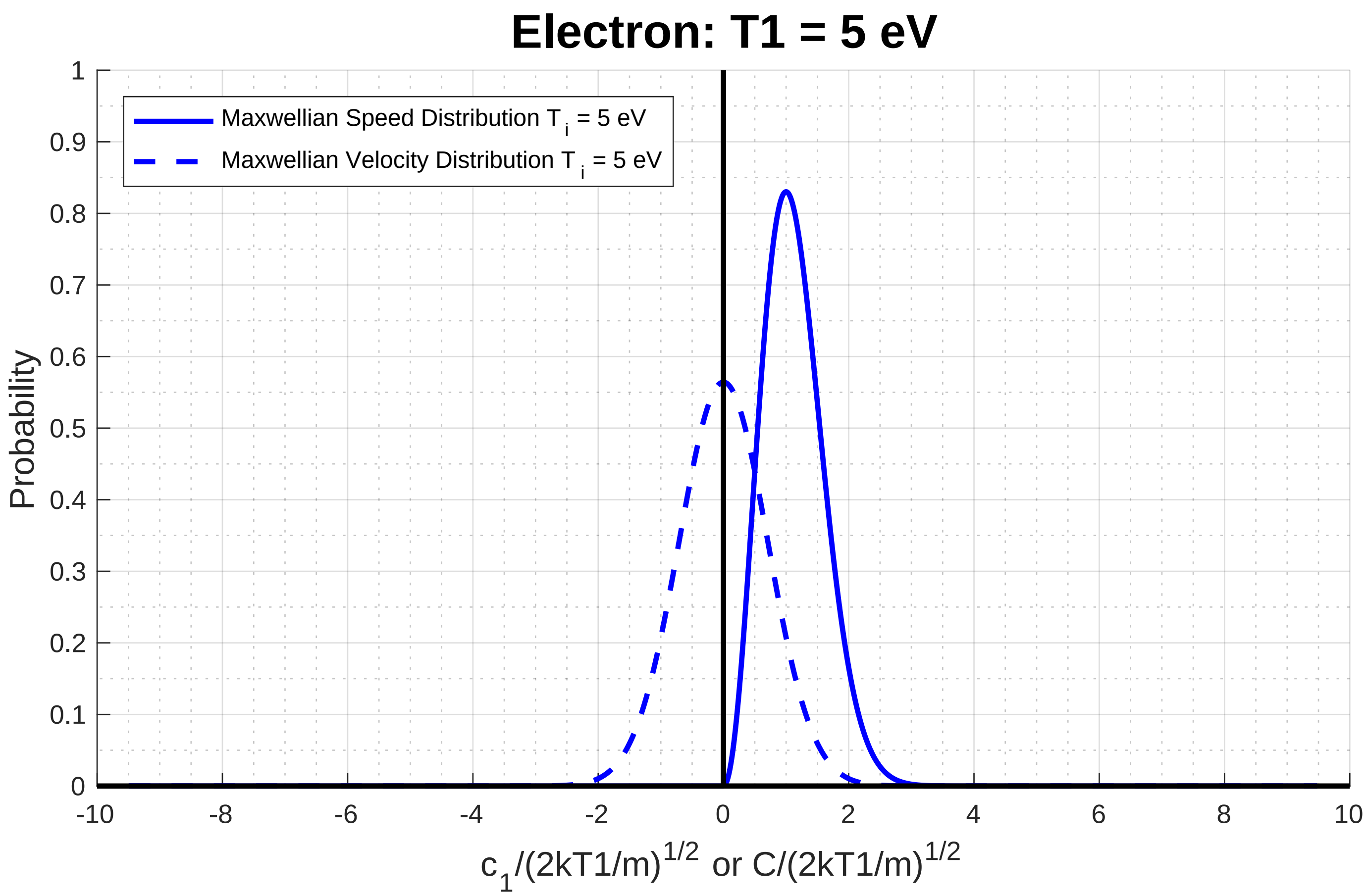
Problem 3: Electron

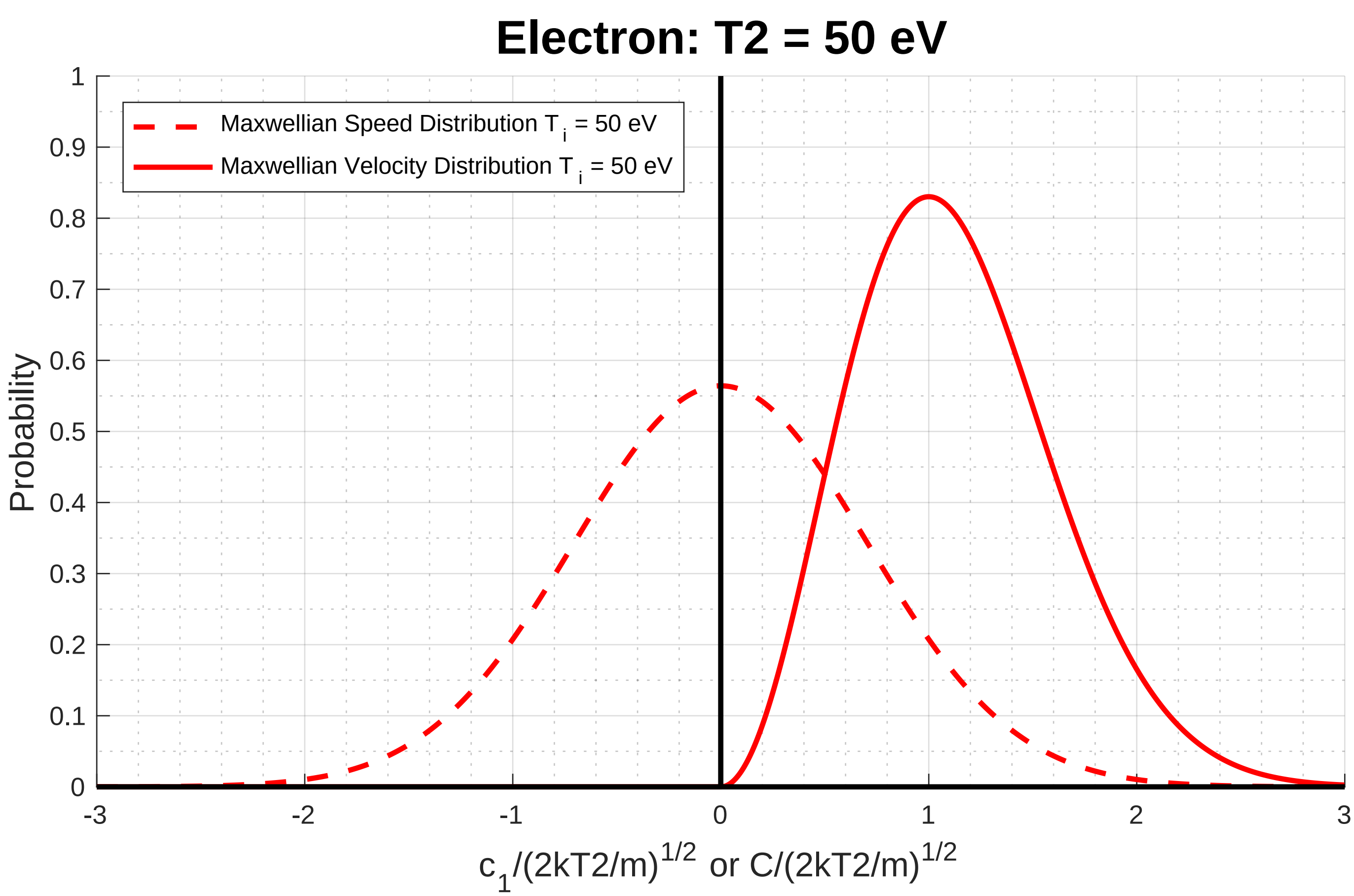
```
clc; clear;

% Initialize variables
m = 4.48e-26;     % Mass of Electron [kg]
k = 1.38e-23;     % Boltzmann's constant [J/K]
Te = [5; 50];           % Particle Temperature [eV]

% Convert eV to K
T = 1.16045221e4*Te; %Temperature [K]

% Compute two frequently used constants
C1 = [4*pi*(m/(2*pi*k*T(1)))^(3/2);4*pi*(m/(2*pi*k*T(2)))^(3/2);]
C2 = [m/(2*k*T(1));m/(2*k*T(2))];
C3 = [(m/(2*pi*k*T(1)))^(1/2);(m/(2*pi*k*T(2)))^(1/2)];

c_mp1 = sqrt((2*k*T(1))/(m)) %Most Probable Speed
c_mp2 = sqrt((2*k*T(2))/(m))

c = [-3*c_mp2:50:3*c_mp2];  % Particle Velocity Magnitude [m/s]
for i=1:length(c)
    %Maxwellian Speed Distribution Function
    if c(i)<0
        chiM(1,i)=0;
        chiM(2,i)=0;
    else
        chiM(1,i) = C1(1) * c(i)^2 * exp(-C2(1) * c(i)^2);
        chiM(2,i) = C1(2) * c(i)^2 * exp(-C2(2) * c(i)^2);
    end

    %Maxwellian Velocity Distribution Function
    fM(1,i) = C3(1) * exp(-C2(1) * c(i)^2);
    fM(2,i) = C3(2) * exp(-C2(2) * c(i)^2);
end

hold on
plot(c/c_mp1,chiM(1,:)*c_mp1,'-b','LineWidth',2) %normalize with cmp.
plot(c/c_mp1,chiM(2,:)*c_mp1,'-r','LineWidth',2) %normalize with cmp.
plot(c/c_mp1,fM(1,:)*c_mp1,'--b','LineWidth',2)
plot(c/c_mp1,fM(2,:)*c_mp1,'--r','LineWidth',2)
hold off

subplot(2,1,1)
hold on
plot(c/c_mp1,chiM(1,:)*c_mp1,'-b','LineWidth',2) %normalize with cmp.
plot(c/c_mp1,fM(1,:)*c_mp1,'--b','LineWidth',2)
hold off
subplot(2,1,2)
hold on
plot(c/c_mp2,fM(2,:)*c_mp2,'--r','LineWidth',2)
plot(c/c_mp2,chiM(2,:)*c_mp2,'-r','LineWidth',2) %normalize with cmp.
```

Figure 2: Problem 3: Normalized Velocity and Speed Distribution Function Code for Electron



Maxwellian Distribution Functions for Electron
Maxwellian Speed Distribution T i = 5 eV
Maxwellian Speed Distribution T i = 50 eV
Maxwellian Velocity Distribution T i = 5 eV
Maxwellian Velocity Distribution T i = 50 eV
Probability
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-10
-8
-6
-4
-2
0
2
4
6
8
10
c1/(2kT/m)1/2 or C/(2kT/m)1/2

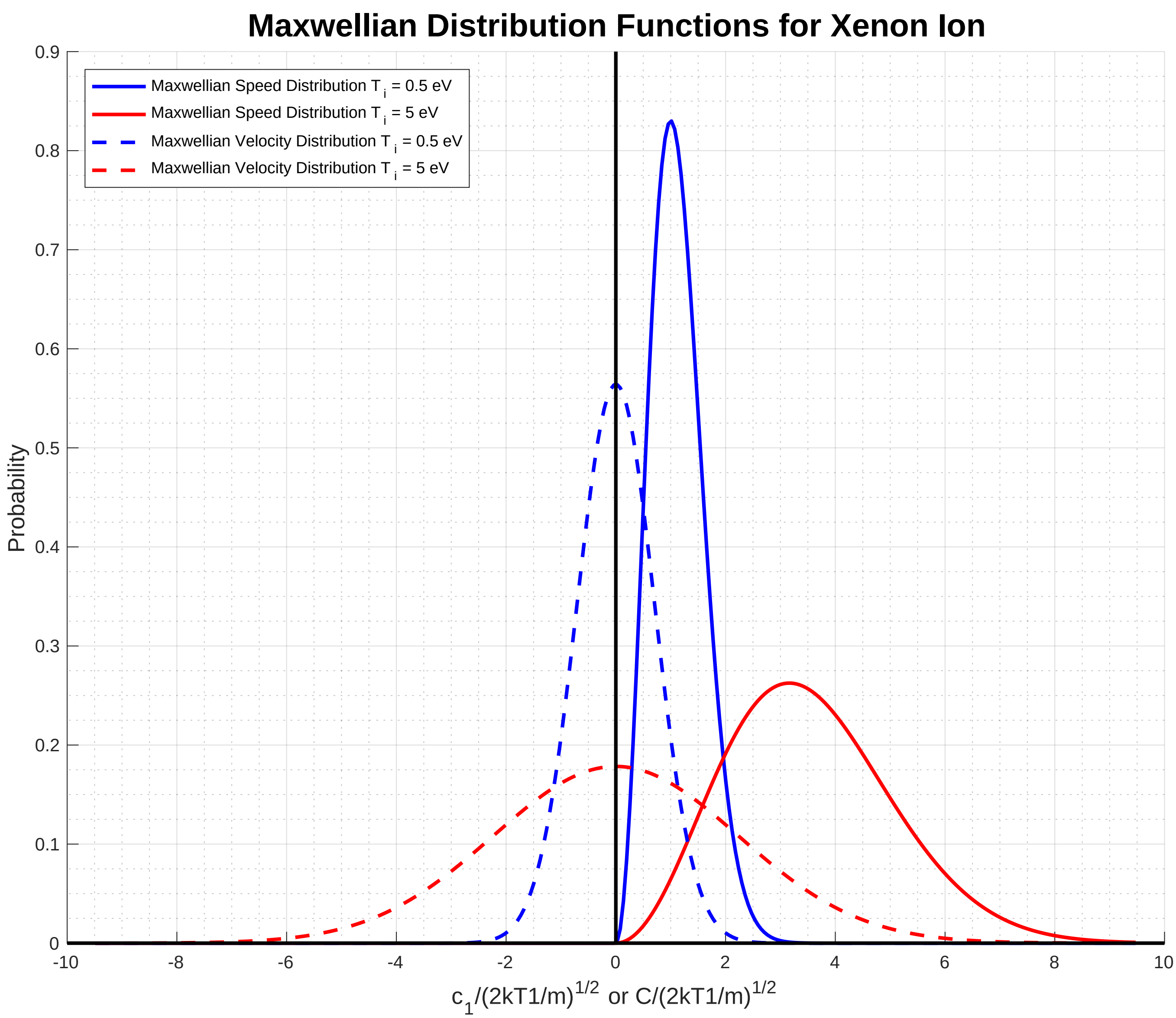

Electron: T1 = 5 eV
Maxwellian Speed Distribution T i = 5 eV
Maxwellian Velocity Distribution T i = 5 eV
Probability
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-10
-8
-6
-4
-2
0
2
4
6
8
10
c1/(2kT1/m)1/2 or C/(2kT1/m)1/2

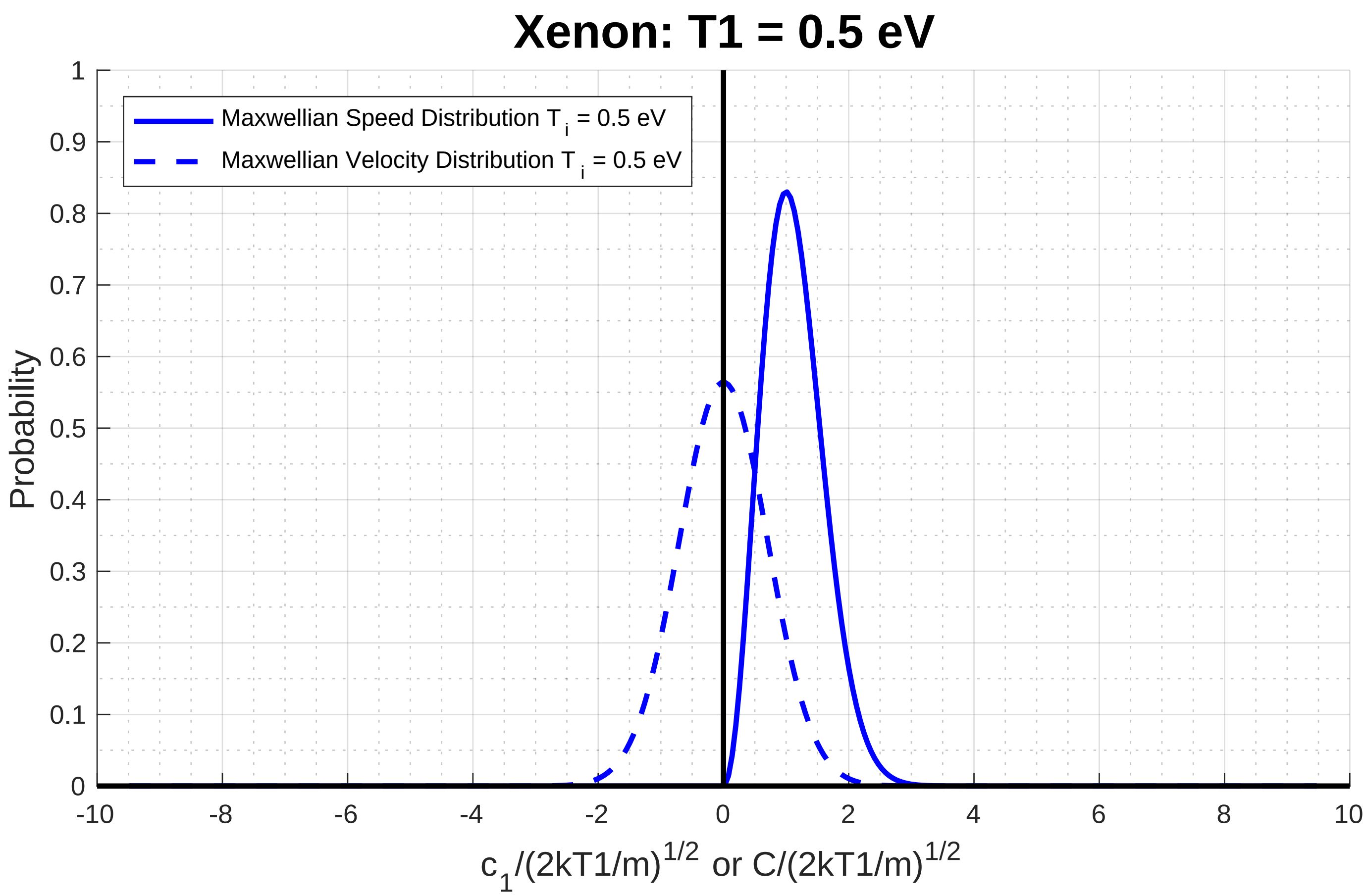

Electron: T2 = 50 eV
Maxwellian Speed Distribution T i = 50 eV
Maxwellian Velocity Distribution T i = 50 eV
Probability
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-3
-2
-1
0
1
2
3
c1/(2kT2/m)1/2 or C/(2kT2/m)1/2

Problem 3:Xenon

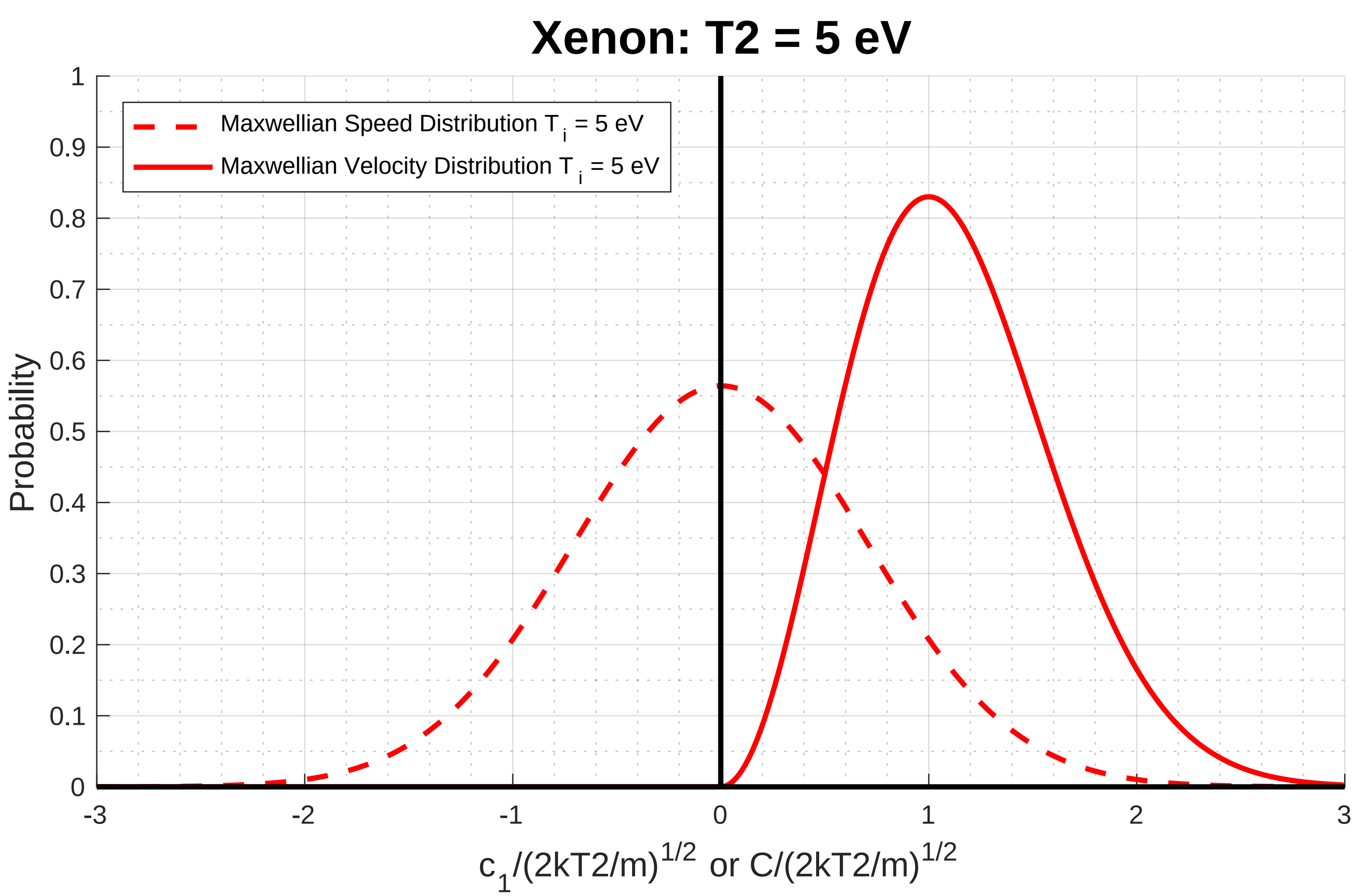
```matlab
clc; clear;

% Initialize variables
m = 2.18e-25;     % Mass of Xe Ion [kg]
k = 1.38e-23;     % Boltzmann's constant [J/K]
Te = [.5; 5];           % Particle Temperature [eV]

% Convert eV to K
T = 1.16045221e4*Te; %Temperature [K]

% Compute two frequently used constants
C1 = [4*pi*(m/(2*pi*k*T(1)))^(3/2);4*pi*(m/(2*pi*k*T(2)))^(3/2);]
C2 = [m/(2*k*T(1));m/(2*k*T(2))];
C3 = [(m/(2*pi*k*T(1)))^(1/2);(m/(2*pi*k*T(2)))^(1/2)];

c_mp1 = sqrt((2*k*T(1))/(m)) %Most Probable Speed
c_mp2 = sqrt((2*k*T(2))/(m))

c = [-3*c_mp2:50:3*c_mp2];  % Particle Velocity Magnitude [m/s]
for i=1:length(c)
    %Maxwellian Speed Distribution Function
    if c(i)<0
        chiM(1,i)=0;
        chiM(2,i)=0;
    else
        chiM(1,i) = C1(1) * c(i)^2 * exp(-C2(1) * c(i)^2);
        chiM(2,i) = C1(2) * c(i)^2 * exp(-C2(2) * c(i)^2);
    end

    %Maxwellian Velocity Distribution Function
    fM(1,i) = C3(1) * exp(-C2(1) * c(i)^2);
    fM(2,i) = C3(2) * exp(-C2(2) * c(i)^2);

end
hold on
plot(c/c_mp1,chiM(1,:)*c_mp1,'-b','LineWidth',2) %normalize with cmp.
plot(c/c_mp1,chiM(2,:)*c_mp1,'-r','LineWidth',2) %normalize with cmp.
plot(c/c_mp1,fM(1,:)*c_mp1,'--b','LineWidth',2)
plot(c/c_mp1,fM(2,:)*c_mp1,'--r','LineWidth',2)
hold off

subplot(2,1,1)
hold on
plot(c/c_mp1,chiM(1,:)*c_mp1,'-b','LineWidth',2) %normalize with cmp.
plot(c/c_mp1,fM(1,:)*c_mp1,'--b','LineWidth',2)

subplot(2,1,2)
hold on
plot(c/c_mp2,fM(2,:)*c_mp2,'--r','LineWidth',2)
plot(c/c_mp2,chiM(2,:)*c_mp2,'-r','LineWidth',2) %normalize with cmp.
```

Figure 3: Problem 3: Normalized Velocity and Speed Distribution Function Code for Xenon


Maxwellian Distribution Functions for Xenon Ion
Maxwellian Speed Distribution T i = 0.5 eV
Maxwellian Speed Distribution T i = 5 eV
Maxwellian Velocity Distribution T i = 0.5 eV
Maxwellian Velocity Distribution T i = 5 eV
Probability
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
-10
-8
-6
-4
-2
0
2
4
6
8
10
c1/(2kT1/m)1/2 or C/(2kT1/m)1/2

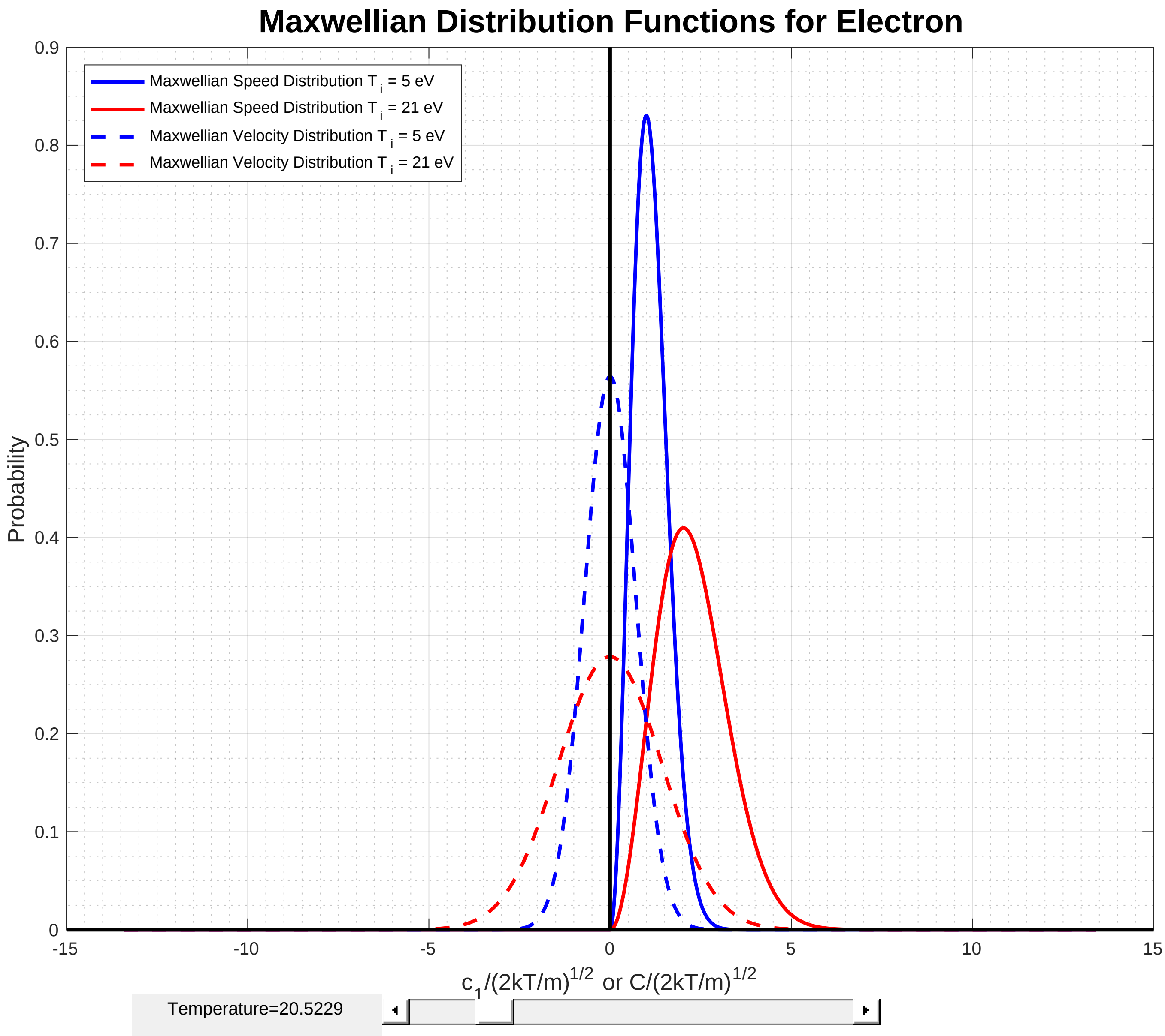


Xenon: T1 = 0.5 eV
Maxwellian Speed Distribution T i = 0.5 eV
Maxwellian Velocity Distribution T i = 0.5 eV
Probability
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-10
-8
-6
-4
-2
0
2
4
6
8
10
c1/(2kT1/m)1/2 or C/(2kT1/m)1/2



Xenon: T2 = 5 eV
Maxwellian Speed Distribution T i = 5 eV
Maxwellian Velocity Distribution T i = 5 eV
Probability
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-3
-2
-1
0
1
2
3
c1/(2kT2/m)1/2 or C/(2kT2/m)1/2

Problem 3: Playground

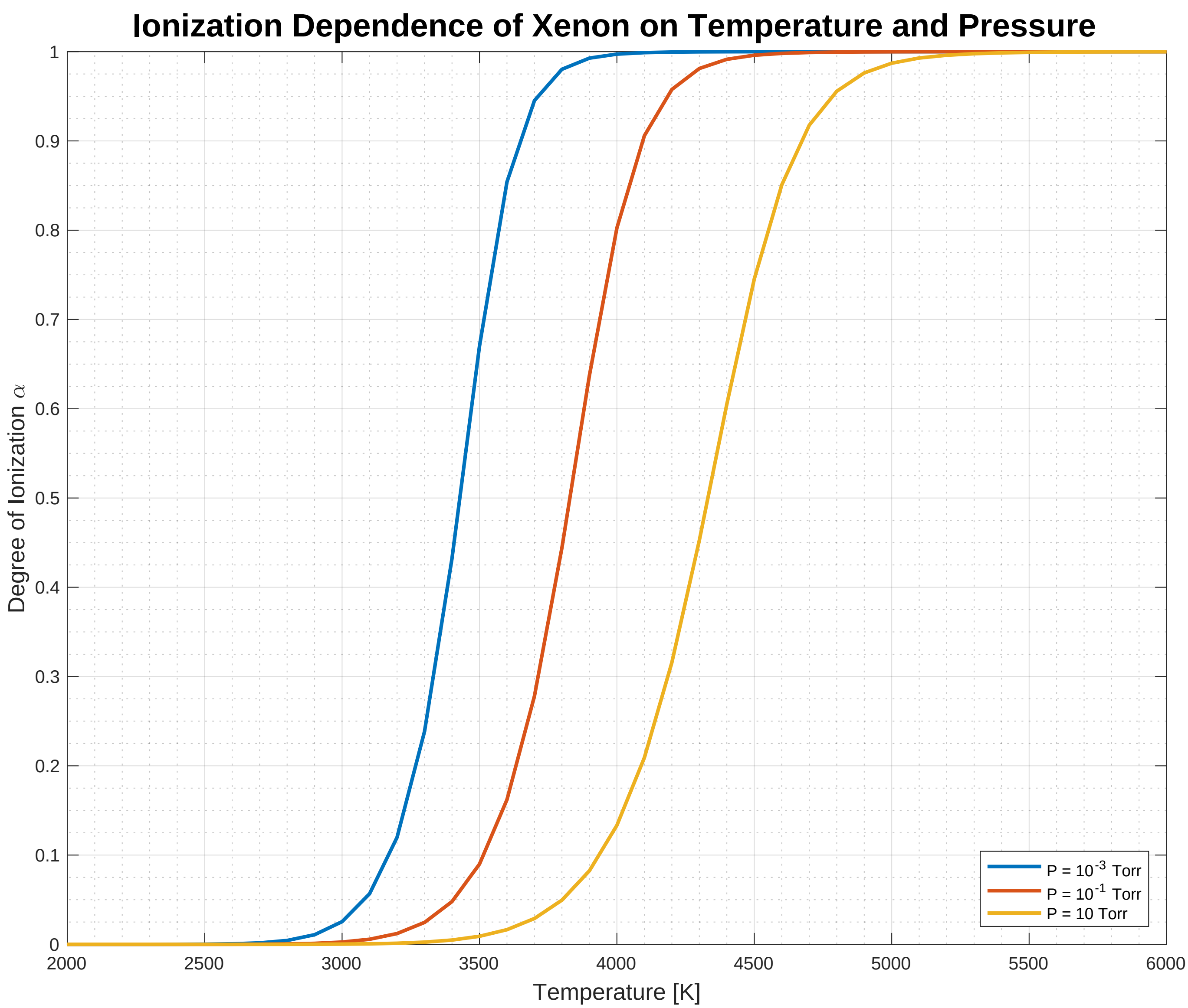


Maxwellian Distribution Functions for Electron
Maxwellian Speed Distribution T i = 5 eV
Maxwellian Speed Distribution T i = 21 eV
Maxwellian Velocity Distribution T i = 5 eV
Maxwellian Velocity Distribution T i = 21 eV
Probability
c1/(2kT/m)1/2 or C/(2kT/m)1/2
Temperature=20.5229

Problem 4:

```matlab
function hw4p4
    T = [2000:100:6000];
    P = [10^-3, 10^-1, 10];
    for i=1:length(T)
        AlphaPT(1,i)=SAHA(P(1),T(i));
        AlphaPT(2,i)=SAHA(P(2),T(i));
        AlphaPT(3,i)=SAHA(P(3),T(i));
    end
    plot(T,AlphaPT,'LineWidth',2)
    %Ploting code removed for space

    %Atomic Xenon Partition Function (AXPF)
    function f_A = AXPF(T)
        th = T/1; %Dimensionless Temperature T/To
        f_A = -1.08e-21*th^5 + 1.86e-16*th^4 -6.49e-12*th^3 + 8.97e-8*th^2 -5.42e-4*th +
            2.02;
    end

    %Ionic Xenon Partition Function (IXPF)
    function f_p = IXPF(T)
        th = T/1; %Dimensionless Temperature T/To
        f_p = 5.8e-17*th^4 -3.71e-12*th^3 + 8.0e-8*th^2 -6.37e-4*th + 6.97;
    end

    %SAHA Equation
    function alpha = SAHA(P,T)
        m = 2.18e-25; %Mass of Xenon [kg]
        k = 1.38e-23; % Boltzmann's constant [J/K]
        h = 6.62607e-34;%Planck's Constant [m^2 kg/s]
        epsilon_i = 12.13; %Ionization Energy wrt. Atomic Ground State [eV]
        epsilon = 1.60218e-19*epsilon_i; %Ionization Energy [J]

        P = P*133.322; %torr to Pa [kg/ m s^2]
        syms alpha real
        term1 = ((2*((2*pi*m)^(3/2))*((k*T)^(5/2)))/(P*h^3));
        term2 = (IXPF(T)/AXPF(T));
        term3 = exp(-epsilon/(k*T));
        eqn1 = ((alpha^2)/(1-alpha^2)) == term1*term2*term3;
        sol = vpa(solve(eqn1, alpha));
        alpha = sol(2);
    end
end
```

Figure 4: Problem 4: Ionization for Xenon with Dependence on Pressure and Temperature



Ionization Dependence of Xenon on Temperature and Pressure
Degree of Ionization α
Temperature [K]
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
2000
2500
3000
3500
4000
4500
5000
5500
6000
P = 10^{-3} Torr
P = 10^{-1} Torr
P = 10 Torr