Bilkent University – Institute of Material Science and Nanotechnology

MSN 514 – Computational Methods for Material Science and Complex Systems

Homework 4 (due 07/03/2021)

Name: Nihal Ahmed
Student ID: 22001044
Department: Material Science & Nanotechnology

Simulate many hard balls moving in a two-dimensional box. Wait for the system to equilibrate and then start moving one of the walls outwards to simulate adiabatic expansion. Note that, the balls that hit the moving wall will slow down. Plot the temperature, pressure and entropy versus volume. Compare the results with classical thermodynamics.

1.0 Introduction

In classical thermodynamics adiabatic process is a process which takes place without transferring heat or mass with the surrounding and system. For the case of adiabatic process Q=0. As there is no heat transfer in the system hence the system's entropy tends to remain constant for adiabatic process. This is the classical explanation of the adiabatic process. According to statistical thermodynamics if the adiabatic process is in thermal equilibrium then such a process is isentropic. Because if we change one parameter of a system slowly which in our case is the volume then all occupation probabilities remain unchanged during the process.

Statistical thermodynamics can be used to interpret the adiabatic process. According to statistical thermodynamics the entropy of a system can be defined using the following expression

$$S = \sum_{i=1}^{N} -p_i \ln(p_i)$$

Here, p_i is the probability of finding the system in it's i th state. İn this report we try to analyze the change in pressure, entropy and temperature with respect to volume for an adiabatic process. We determine the parameters based on statistical thermodynamics and then compare the results with the classical approach.

2.0 Numerical Calculation

2.1 Statistical Thermodynamics

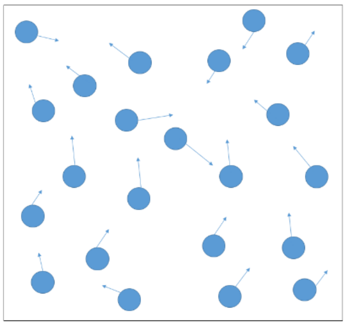


Figure 1. Gas molecules inside closed volume

2.1.1 Collusion of Balls

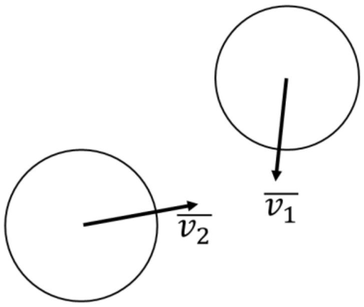


Figure 2. Collusion between two balls

The collusion between two balls is defined using the following equation

$$\overline{v_2'} = \overline{v_2} - [(\overline{v_2} - \overline{v_1}).(\overline{r_2} - \overline{r_1})]\frac{\overline{r_1} - \overline{r_2}}{(\overline{r_1} - \overline{r_2})^2}$$

2.1.2 Collusion of balls with walls

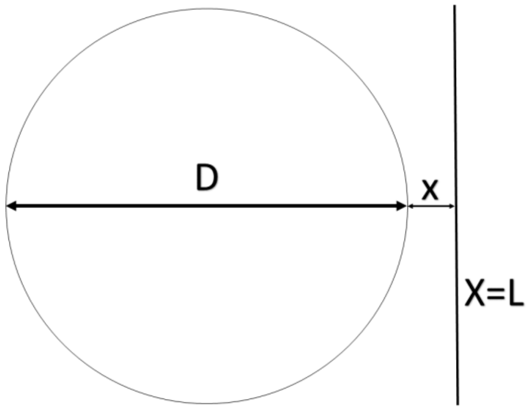


Figure 3. Collusion of wall with balls

The collusion of the ball and wall is defined as

$$if\ x > L - \frac{D}{2}\ then\ v_x = -v_x + 2v_w$$

Here we have added v_w term as the wall is slowly moving away otherwise the system will act as a isothermal system instead of adiabatic.

We know that,

Shanon Entropy, $S = \sum_{i=1}^{N} -p_i \ln(p_i)$

Now in order know all the information of a classical system one must have information about the position and velocity of the system. Now we calculate the entropy for the adiabatic system with from phase space

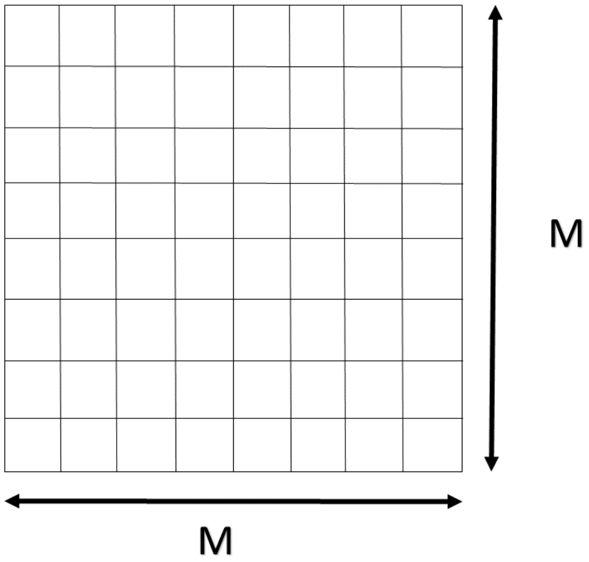


Figure 4. Microstates of the system

We can define the entropy of the system by following equation

$$S = (iv_y - 1)M^3 + (iv_x - 1)M^2 + (ix - 1)M + iy$$

Here, x and y are position component and v_x and v_y are velocity components. Here we have 3 as highest power of M as the system is four dimensional

We calculate the temperature from velocity distribution

$$T = \frac{1}{N}\sum\frac{v^2}{2}$$

Here, N=number of total particles

2.2 Classical Thermodynamics

In classical thermodynamics real gas is defined by the following equation

$$(P + a)(V - b) = nRT$$

And for isothermal system

$$PV^{\gamma} = VT^{\frac{1}{\gamma-1}}\text{=}constant$$

For our system $C_v = \frac{2}{2} = 1$ considering R=1 and T=1

And $C_p = C_v + R = 1 + 1 = 2$

Hence $\gamma = \frac{C_p}{C_v} = 2$

Classically the pressure and temperature are calculated using the following equation

$$P\infty a(V - b)^{-2}$$

And $\mathrm{T}\infty(V - b)^{-1}$

3.0 Result & Discussion

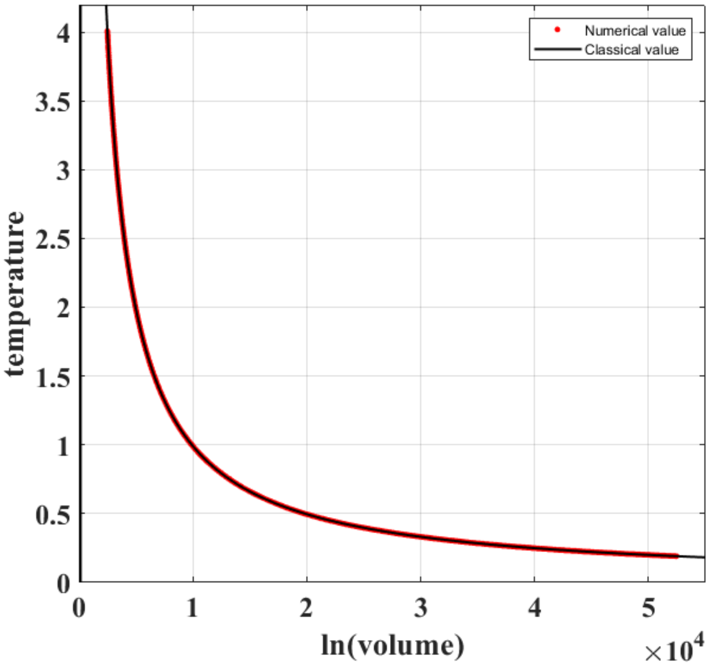


Figure 5. Temperature distribution with respect to volume change

The graph indicates that there is very less fluctuation between the statistical value and classical value. The graph shows that as we increase the volume the temperature decreases which is the case for adiabatic expansion. As we increase the volume the gas gets cooled.

Figure 6 depicts that as we increase the volume then the pressure decreases and this decrease is in the order of -1. There are certain discrepancy between the statistical value and the classical value but this is due to the numerical computational error. The nature of both curves are similar hence they can be considered accurate.

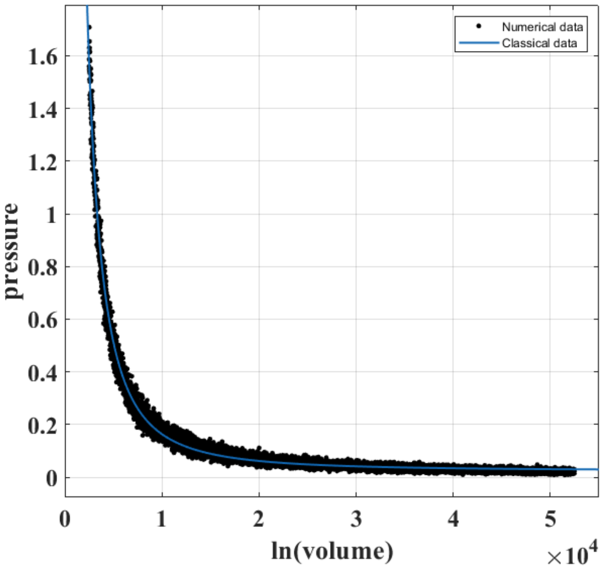


Figure 6. pressure distribution with respect to volume change

Figure 7 indicates the change in entropy with respect to increase in volume. As we slowly increase the volume hence it becomes an isentropic system meaning that the entropy will remain constant. Although there is some increase in the entropy but it is negligible and this is basically because of the numerical computation limitations.

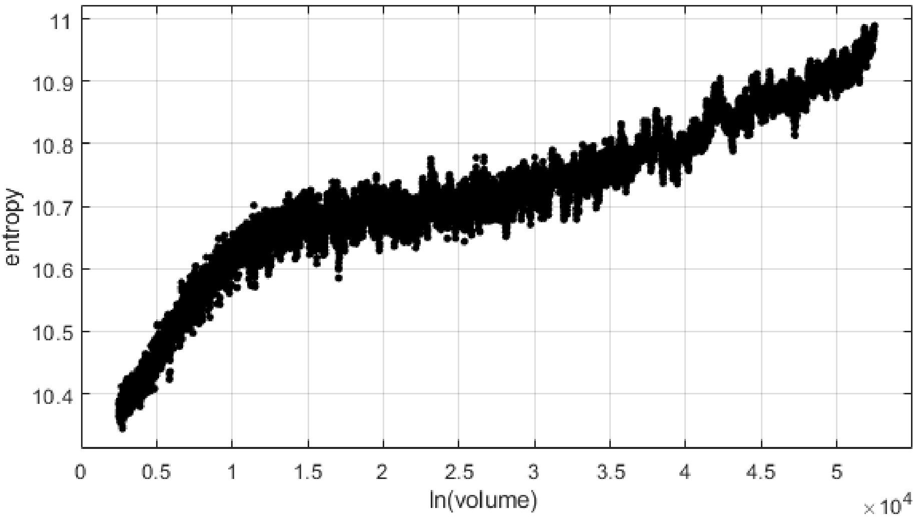


Figure 7. entropy distribution with respect to volume change

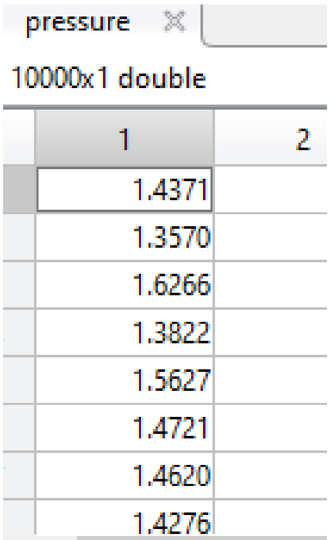
4.0 Conclusion

In this report the pressure ,entropy and temperature with respect to volume has been determined for an adiabatic expansion. The results indicate that the pressure and temperature drops for the adiabatic expansion and the entropy remains almost constant. Both classical and statistical approach give similar result. Hence both approach can be taken to solve such systems.

Appendix

```
clear
L = [50 50]; % dimensions of the box
N = 1000; % number of particles
D = 1/4; % diameter of particles
r = rand(N,2)*(L(1,1)-D/2); % define initial positions
v = randn(N,2); % define initial velocities
dt = 0.1; % time step

T = 10000; % number of steps
Q = 100; % ensemble over time

M = 10;
d_vx = 1; min_vx = -5;
d_vy = 1; min_vy = -5;
d_rx = 2*L(1,1)/(M - 1); min_rx = 0;
d_ry = 2*L(1,2)/(M - 1); min_ry = 0;
d_ps = d_vx*d_vy*d_rx*d_ry;
n_ps(Q,M^4) = 0;
entropy(T,1) = 0; % entropy calculated from phase space

vL=[0.01 0];
```

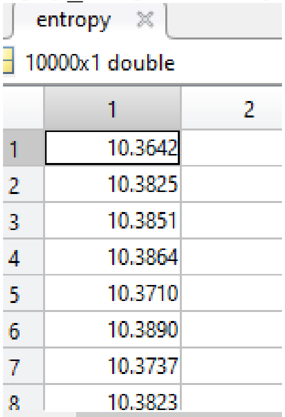
```
for t=1:T

    d_rx = L(1,1)/9;
    d_ry = L(1,2)/9;
    d_vrxy = d_vx*d_vy*d_rx*d_ry;

    pressure_q(1:Q,1) = 0;
    temperature_q(1:Q,1) = 0;

    for q=1:Q
          % This part defines the collisions between balls
    [sx,nx] = sort(r(:,1)); % sort the x coordinate
    for kn = 1 : N-1
        n = nx(kn); % scan particles from left to right
        for km = kn+1:N
            m = nx(km);
            if (sx(km) - sx(kn) > D) % stop if x's differ more than D
                break
            else
                rp = r(n,:)-r(m,:); % vector between two balls
                nrp = norm(rp);
                if (nrp < D) % make sure that particles are close enough
                    rv = v(n,:)-v(m,:); % velocity difference vector
                    if (rv*rp'<0) % make sure that particles are moving
towards each other
                        v(n,:) = v(n,:) - (rv*rp')*rp/nrp^2; % bounce
                        v(m,:) = v(m,:) + (rv*rp')*rp/nrp^2; % bounce
                    end
                end
            end
        end
    end

    %this part calculates the collusions of particles with wall and
    %pressure
     nv = (v(:,1) > 0).*(r(:,1) > L(1,1)-D/2); v(nv==1,1) = -
v(nv==1,1)+2*vL(1,1); pressure_q(q,1)=pressure_q(q,1)-
2*sum(v(nv==1,1))/dt/L(1,2)-2*sum(nv==1)*vL(1,1)/dt/L(1,2);
     nv = (v(:,2) > 0).*(r(:,2) > L(1,2)-D/2); v(nv==1,2) = -
v(nv==1,2)+2*vL(1,2); pressure_q(q,1)=pressure_q(q,1)-
2*sum(v(nv==1,2))/dt/L(1,2)-2*sum(nv==1)*vL(1,2)/dt/L(1,1);
     nv = (v(:,1) < 0).*(r(:,1) < D/2); v(nv==1,1) = -v(nv==1,1);
pressure_q(q,1)=pressure_q(q,1)+2*sum(v(nv==1,1))/dt/L(1,2);
     nv = (v(:,2) < 0).*(r(:,2) < D/2); v(nv==1,2) = -v(nv==1,2);
pressure_q(q,1)=pressure_q(q,1)+2*sum(v(nv==1,2))/dt/L(1,1);
```


pressure

10000x1 double

1	2
1.4371	
1.3570	
1.6266	
1.3822	
1.5627	
1.4721	
1.4620	
1.4276	

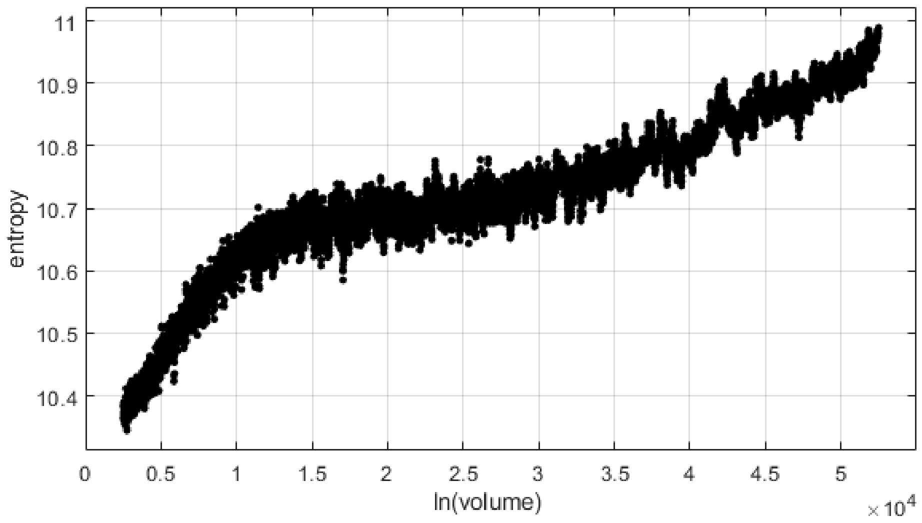
```
     % Calculate entropy from velocity distribution
    c_vrxy = M^3*floor((v(:,1) - min_vx)/d_vx + 1/2) + M^2*floor((v(:,2) -
min_vy)/d_vy + 1/2) + ...
             M*floor((r(:,1) - min_rx)/d_rx) + floor((r(:,2) -
min_ry)/d_ry) + 1;
    [n_vrxy(q,1:M^4)] = hist(c_vrxy,(1:M^4));

    %calculate temperature
    temperature_q(q,1)= 2*sum(v(:,1).^2+v(:,2).^2)/N;
    r = r+v*dt;
    end

    volume(t,1)= L(1,1)*L(1,2);
    pressure(t,1)=sum(pressure_q)/Q;
    temperature(t,1)=sum(temperature_q)/Q;

    p_vrxy= sum(n_vrxy)/sum(sum(n_vrxy))/d_vrxy;
    entropy(t,1) =sum(-
p_vrxy(p_vrxy>0).*log(p_vrxy(p_vrxy>0)))*d_vrxy;%normalizing the entropy
```


entropy

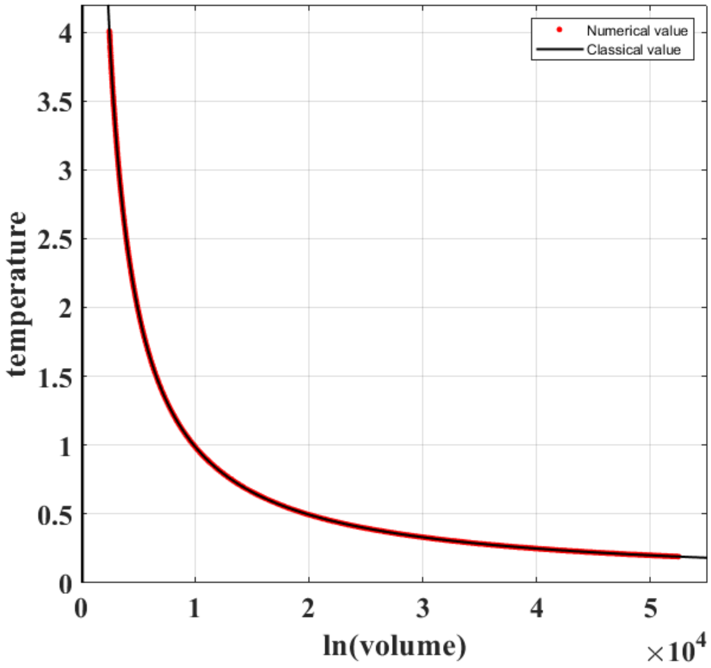
10000x1 double

	1	2
1	10.3642	
2	10.3825	
3	10.3851	
4	10.3864	
5	10.3710	
6	10.3890	
7	10.3737	
8	10.3823	

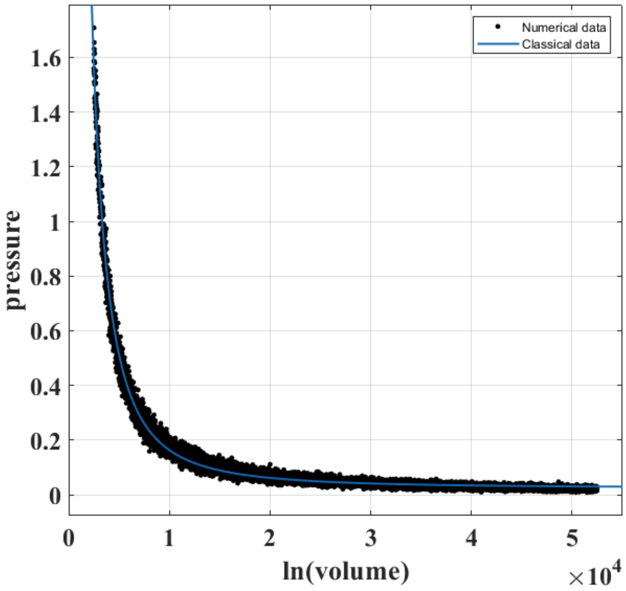
```
    plot(log(volume(1:t,1)),entropy(1:t,1),'r','Linewidth',2);
```



```
plot(log(volume(1:t,1)),temperature(1:t,1),'r','Linewidth',2);
```



```
plot(log(volume(1:t,1)),pressure(1:t,1),'r','Linewidth',2);
```



```
    L = L+vL*Q*dt;

    f=polyfit(log(volume(1:t,1)),entropy(1:t,1),1);
    [t f]
end
```