CHE-622

Grand Canonical Ensemble – Monte Carlo Code

Instructor:

Dr. Pankaj Apte

Submitted By:

Gaurav Dixit (Y8205)

Utsav Kumar (Y8127546)

Vikas Mishra (Y8127564)

Date of Submission: 25.04.2012

MAIN PROGRAM

```
function [TOT_ENERGY, RHO_AVG, N] = GC_MC()
%This function calculates the average energy and density for Grand
%Canonical Ensemble

% Parameters for the Ensemble are:-
% <SIGMA>
% <VOLUME>
% <R CUT>
% <ABSOLUTE TEMPERATURE - TEMP>

% Other Parameters
% <L - CUBE LENGTH>
% <Z - ACTIVITY>
% <EPSILON - POTENTIAL DEPTH>
% <K - BOLTZMAN CONSTANT>
% <EPSILON/K = 119.8K>
% <TEMPERATUR = 86.5K>
% <TEMP = K*T/EPSILON = 0.722>

%PARAMETER VALUES

SIGMA=0.3405e-9;
R_CUT=2.5.*SIGMA;
Vol=307.937*SIGMA^(3);
L=Vol^(1/3);
V=307.937;
z=0.8314;
TEMP=0.722;

NMIN=1;
NMAX=1000;

%ARRAYS (RX, RY and RZ)to store the location of the particles, Whereas
%LOCATE is used to store update the deletion and insertion of the particle
%by moving the deleted particle at the end of the array

LOCATE=zeros(1,NMAX);
RX=zeros(1,NMAX);
RY=zeros(1,NMAX);
RZ=zeros(1,NMAX);

N=0;                          %Number of particles
NCYCLE=100000;                %Number of cycle of runs
DELTAR=0.025.*SIGMA;          %Maximum displacement allowed for each particle

TOT_ENERGY=0;                 %Stores the TOTAL ENERGY of the system


for i=1:1:NCYCLE
    r=rand;                             %random number generator between (0-1)

    % FOR INSERTION OF ATOM - CREATION
```

```
        if (r <=0.33)           %Enters the loop if r<=0.33
            NTRIAL=N+1;         %Increases the value of N by a dummy variable
            if (NTRIAL < NMAX )
                %Randomly assigning a position to the atom

                RX_NEW=(rand-0.5).*L./2;
                RY_NEW=(rand-0.5).*L./2;
                RZ_NEW=(rand-0.5).*L./2;

%Calculating the change in POTENTIAL because of the introduction of new
particle
% Expression has to be multiplied with POTENTIAL DEPT i.e.EPSILON
                [DELTV]=DELTA_POT(R_CUT, SIGMA, RX_NEW, RY_NEW, RZ_NEW, RX,
                                                                    RY, RZ, N);

                %Condition for the METROPOLIS
                DELTCB=1+((NTRIAL).*exp((1./TEMP).*DELTV)./(z.*V));
                ACCEPT=1/DELTCB;

                %Checking whether the move is ACCEPTED or not if YES then
                %value is updated like POTENTIAL, POSITION and N
                if (rand <=ACCEPT);
                    [RX, RY, RZ, LOCATE]=ADD_POSN(RX_NEW, RY_NEW, RZ_NEW,
                                            NTRIAL, LOCATE, RX, RY, RZ);

%DELTV multiplied with EPSILON (i.e 119.8(EPSILON/K) *K(BOLTZMAN CONSTANT))
                    POT=DELTV.*119.8.*1.38.*10^(-23);
                    TOT_ENERGY=TOT_ENERGY+POT;
                    N=NTRIAL;
                end
            end

        % FOR DELETION OF PARTICLE _ DESTRUCTION
        elseif (r <=0.66 && r>0.33)

            NTRIAL = N-1;       %DUMMY value of updated N

            if (NTRIAL > NMIN)  %IF Number of particles is > 1
                I=randi(N);     %RANDOMLY choosing an atom

                %Calculating the change in potential on DELETION
                %DELTV value has to multiplied with EPSILON
                [DELTV]=DELTA_POT(R_CUT, SIGMA, RX(I), RY(I), RZ(I), RX, RY,
                                                                    RZ, N);

                %CONDITION for METROPOLIS, here EPSILON value is present in
                %ABSOLUTE TEMPERATUR i.e TEMP
                DELTCB=1+(z.*V).*exp((1./TEMP).*DELTV)./(NTRIAL);
                ACCEPT=1/DELTCB;

                %%Checking whether the move is ACCEPTED or not if YES then
                %value is updated like POTENTIAL, POSITION and N
                if (rand <=ACCEPT);
                    [RX, RY, RZ, LOCATE]=DEL_POSN(I, RX, RY, RZ, LOCATE,
                                                                    NTRIAL);
```

```
%DELTV multiplied with EPSILON (i.e 119.8(EPSILON/K) *K(BOLTZMAN CONSTANT))
                    POT=DELTV.*119.8.*1.38.*10^(-23);
                    TOT_ENERGY=TOT_ENERGY+POT;
                    N=NTRIAL;
                end

            end

        %FOR DISPLACEMENT OF PARTICLE - DISPLACEMENT
        elseif (r <=1 && r >0.66)

            if N>1
            I=randi(N);                         %RANDOMLY selecting a particle

            %Giving perticle shift in X,Y and Z direction
            RX_DISP=RX(I)+DELTAR;
            RY_DISP=RY(I)+DELTAR;
            RZ_DISP=RZ(I)+DELTAR;

            %PERIODIC BOUNDARY CONDITION applied
            if (RX_DISP>L/2 || RX_DISP<-L/2)
                RX_DISP=RX_DISP-sign(RX_DISP).*L;
            end
            if (RY_DISP>L/2 || RY_DISP<-L/2)
                RY_DISP=RY_DISP-sign(RY_DISP).*L;
            end
            if (RZ_DISP>L/2 || RZ_DISP<-L/2)
                RZ_DISP=RZ_DISP-sign(RZ_DISP).*L;
            end

            %Calculating the change in potential on DELETION
            %DELTV value has to multiplied with EPSILON
            [DELTV]=DELTA_POT_DISP(R_CUT, SIGMA, RX_DISP, RY_DISP, RZ_DISP,
                                                    RX, RY, RZ, N, I);

            %CONDITION for METROPOLIS, here EPSILON value is present in
            %ABSOLUTE TEMPERATURE i.e TEMP
            DELTCB=exp((-1./TEMP).*(DELTV));

            %Checking whether the move is ACCEPTED or not if YES then
            %value is updated like POTENTIAL, POSITION and N
            ACCEPT=min(1,DELTCB);
                if (rand <=ACCEPT);
                    [RX, RY, RZ]=UPDATE_POSN(RX_DISP, RY_DISP, RZ_DISP, I,
                                                        RX, RY, RZ);

%DELTV multiplied with EPSILON (i.e 119.8(EPSILON/K) *K(BOLTZMAN CONSTANT))
                    POT=DELTV.*119.8.*1.38.*10^(-23);
                    TOT_ENERGY=TOT_ENERGY+POT;
                end

            end
        end
end
```

```
%To get the TOTAL ENERGY in PER MOLE form so multiplied with AVAGADRO
%NUMBER and DIVIDED by N i.e. TOTAL NUMBER OF ATOMS IN SYSTEM

TOT_ENERGY=TOT_ENERGY.*6.*10^(23)./N;

%AVERAGE DENSITY
RHO_AVG = N./V;
end
```

VARIABLES

SIGMA = 0.3405 nm

VOLUME = 307.9 $SIGMA^3$

R CUT = 2.5 SIGMA

ABSOLUTE TEMPERATURE – TEMP

L - CUBE LENGTH = $VOLUME^{1/3}$

Z – ACTIVITY = ρ (initially taken as this and adjusted to get the values as close as possible to the original ones)

EPSILON/K = 119.8K

TEMPERATURE = 86.5K

TEMP = K*T/EPSILON = 0.722

CODE FOR CALCULATING CHANGE IN POTENTIAL

FOR DELETION AND INSERTION OF PARTICLE

```
function [DELTV] = DELTA_POT(R_CUT, SIGMA, RXI, RYI, RZI, RX, RY, RZ, N)
%To Calculate the CHANGE IN POTENTIAL when a particle is INSERTED or
%DELETED from the box. For both the cases only the DELTV sign changes
RCUTSQ=R_CUT*R_CUT;
SIGSQ=SIGMA*SIGMA;
DELTV=0;

for J=1:1:N %Here N is total number of atoms present
    RXIJ=RXI-RX(J);
    RYIJ=RYI-RY(J);
    RZIJ=RZI-RZ(J);

    RIJSQ=RXIJ.*RXIJ + RYIJ.*RYIJ + RZIJ.*RZIJ;

    if (RIJSQ <= RCUTSQ)
        SR2=SIGSQ./RIJSQ;
        SR6=SR2.*SR2.*SR2;
        VIJ=SR6.*(SR6-1);
        DELTV=DELTV+VIJ;
    end
end

end
```

FOR DISPLACEMENT OF PARTICLE

```
function [DELTV] = DELTA_POT_DISP(R_CUT, SIGMA, RXI, RYI, RZI, RX, RY, RZ, N,
I)
%To Calculate the CHANGE IN POTENTIAL when a particle is moved from its
%ORIGINAL POSITION to a NEW POSITION

R_CUT=2.5.*SIGMA;
RCUTSQ=R_CUT*R_CUT;
SIGSQ=SIGMA.*SIGMA;
DELTV=0;%CHANGE IN POTENTIAL

for J=1:1:N %Here N is total number of atoms present
    if (J~=I)%To avoid calculating the particles old position
        RXIJ=RXI-RX(J);
        RYIJ=RYI-RY(J);
        RZIJ=RZI-RZ(J);

        RXIJ_OLD=RX(I)-RX(J);
        RYIJ_OLD=RY(I)-RY(J);
        RZIJ_OLD=RZ(I)-RZ(J);

        RIJSQ=RXIJ.*RXIJ + RYIJ.*RYIJ + RZIJ.*RZIJ;
        RIJSQ_OLD=RXIJ_OLD.*RXIJ_OLD + RYIJ_OLD.*RYIJ_OLD +
RZIJ_OLD.*RZIJ_OLD;
```

```
        %Only those atoms address are passed which are at a distance of LESS
THAN R_CUT and the
        %DELTV(change in potential) is returned back but it is not multiplied
with
        %EPSILON which will be done in MAIN PROGRAM
        if (RIJSQ <= RCUTSQ)
            SR2=SIGSQ./RIJSQ;
            SR2_OLD=SIGSQ./RIJSQ_OLD;
            SR6=SR2.*SR2.*SR2;
            SR6_OLD=SR2_OLD.*SR2_OLD.*SR2_OLD;
            VIJ=4.*((SR6.*(SR6-1))-(SR6_OLD.*(SR6_OLD-1)));
            DELTV=DELTV+VIJ;
        end
    end
end

end
```

FOR UPDATING THE ADDRESS MATRIX OF ATOM

ADDING POSITION DURING INSERTION

```
function [RX, RY, RZ, LOCATE] = ADD_POSN(RX_NEW, RY_NEW, RZ_NEW, NTRIAL,
LOCATE, RX, RY, RZ)
%Function to INSERT an atom
%If LOCATE already contains a deleted particle which is send to the end of
%array is updated if the position is already empty then new value are
%inserted
if LOCATE(NTRIAL)~=0
   ATOM = LOCATE(NTRIAL);
   RX(ATOM)=RX_NEW;
   RY(ATOM)=RY_NEW;
   RZ(ATOM)=RZ_NEW;
else
    LOCATE(NTRIAL)=NTRIAL;
    RX(NTRIAL)=RX_NEW;
    RY(NTRIAL)=RY_NEW;
    RZ(NTRIAL)=RZ_NEW;
end
end
```

DELETING POSITION DURING DESTRUCTION

```
function [RX, RY, RZ, LOCATE] = DEL_POSN(I, RX, RY, RZ, LOCATE, NTRIAL)
%ATOM location are DELETED by moving it to the end of the array i.e. after
%Nth position so when we read the address till N we avoid deleted particle
%address

ATOM=LOCATE(I);
RX(ATOM)=Rx;
RY(ATOM)=Ry;
RZ(ATOM)=Rz;

for i=I:1:NTRIAL
    RX(i)=RX(i+1);
    RY(i)=RY(i+1);
    RZ(i)=RZ(i+1);
end

RX(NTRIAL+1)=Rx;
RY(NTRIAL+1)=Ry;
RZ(NTRIAL+1)=Rz;

LOCATE(NTRIAL+1)=I;


end
```

UPDATING POSITION DURING DISPLACEMENT

```
function [RX, RY, RZ] = UPDATE_POSN(RX_DISP, RY_DISP, RZ_DISP, I, RX, RY, RZ)
%Updating the position of the DISPLACED PARTICLE if the move is accepted
RX(I)=RX_DISP;
RY(I)=RY_DISP;
RZ(I)=RZ_DISP;
end
```

OTHER FUNCTIONS USED

LONG RANGE CORRECTION FACTOR

```
function [VLRCO] = LRC(SIGMA, N, R_CUT, V)
%To Calculate the LONG RANGE CORRECTION factor

PI=3.14;
SIGSQ=SIGMA.*SIGMA;
SIGCUB = SIGSQ.*SIGMA;
SR3 = (SIGMA./R_CUT).^(3);
SR9 = SR3.^(3);
VLRCO=(8./9).*PI*SIGCUB*(SR9-3.*SR3).*N./V;


end
```

TOTAL ENERGY CALCULATION OF SYSTEM

```
function [TOT_ENERGY] = TOTAL_ENERGY(R_CUT, SIGMA, RX, RY, RZ, N, V)

%Program To CALCULATE THE TOTAL ENERGY of the system

RCUTSQ=R_CUT*R_CUT;
SIGSQ=SIGMA*SIGMA;
DELTV=0;

for I=1:1:N %Here n is total number of atoms present
    for J=I+1:1:N
        RXIJ=RX(I)-RX(J);
        RYIJ=RY(I)-RY(J);
        RZIJ=RZ(I)-RZ(J);

        RIJSQ=RXIJ.*RXIJ + RYIJ.*RYIJ + RZIJ.*RZIJ;

        if (RIJSQ <= RCUTSQ)
            SR2=SIGSQ./RIJSQ;
            SR6=SR2.*SR2.*SR2;
            VIJ=4.*SR6.*(SR6-1);
            DELTV=DELTV+VIJ;
        end
    end
end

TOT_ENERGY=(DELTV.*8.314.*119.8.*N);
end
```

RESULTS

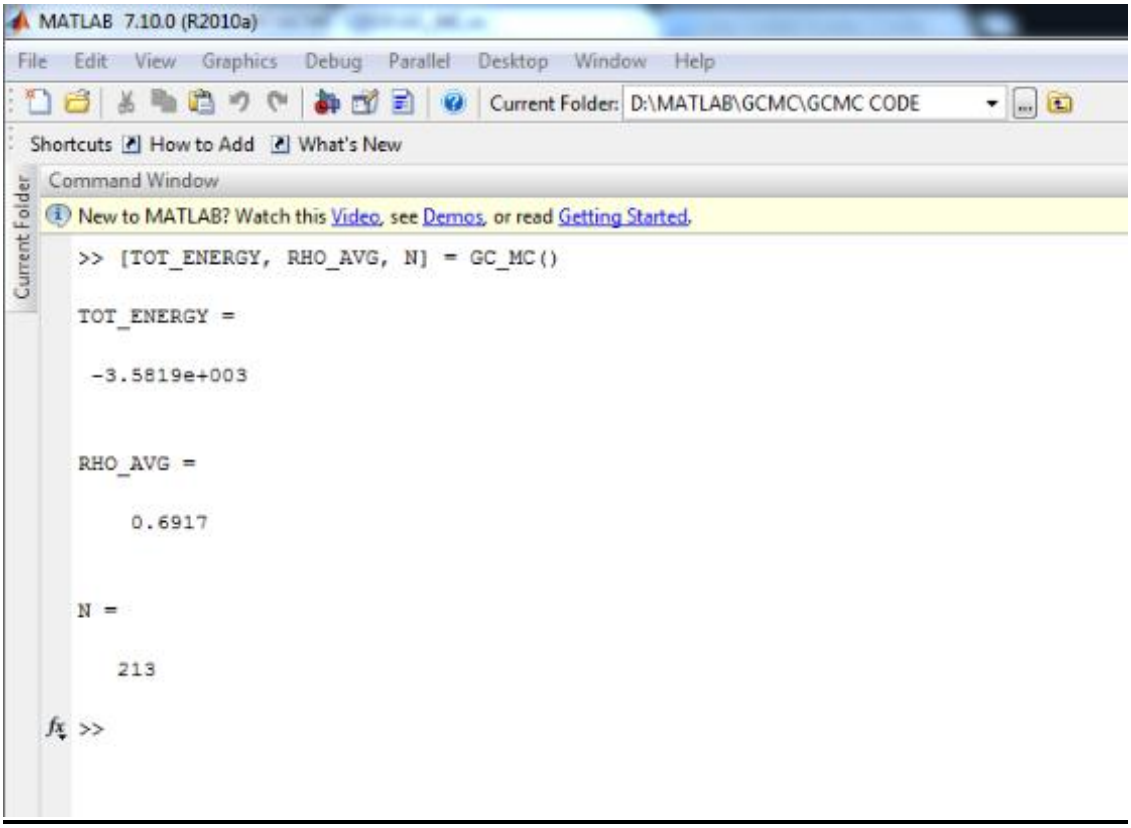
VARIABLE	SIMULATION VALUES	ORIGINAL VALUES
<U>	-3581.9 J/mole	-4897.4 J/mole
<ρ>	0.6917 Moles/m^3	0.8314 Moles/m^3
<N>	213	256

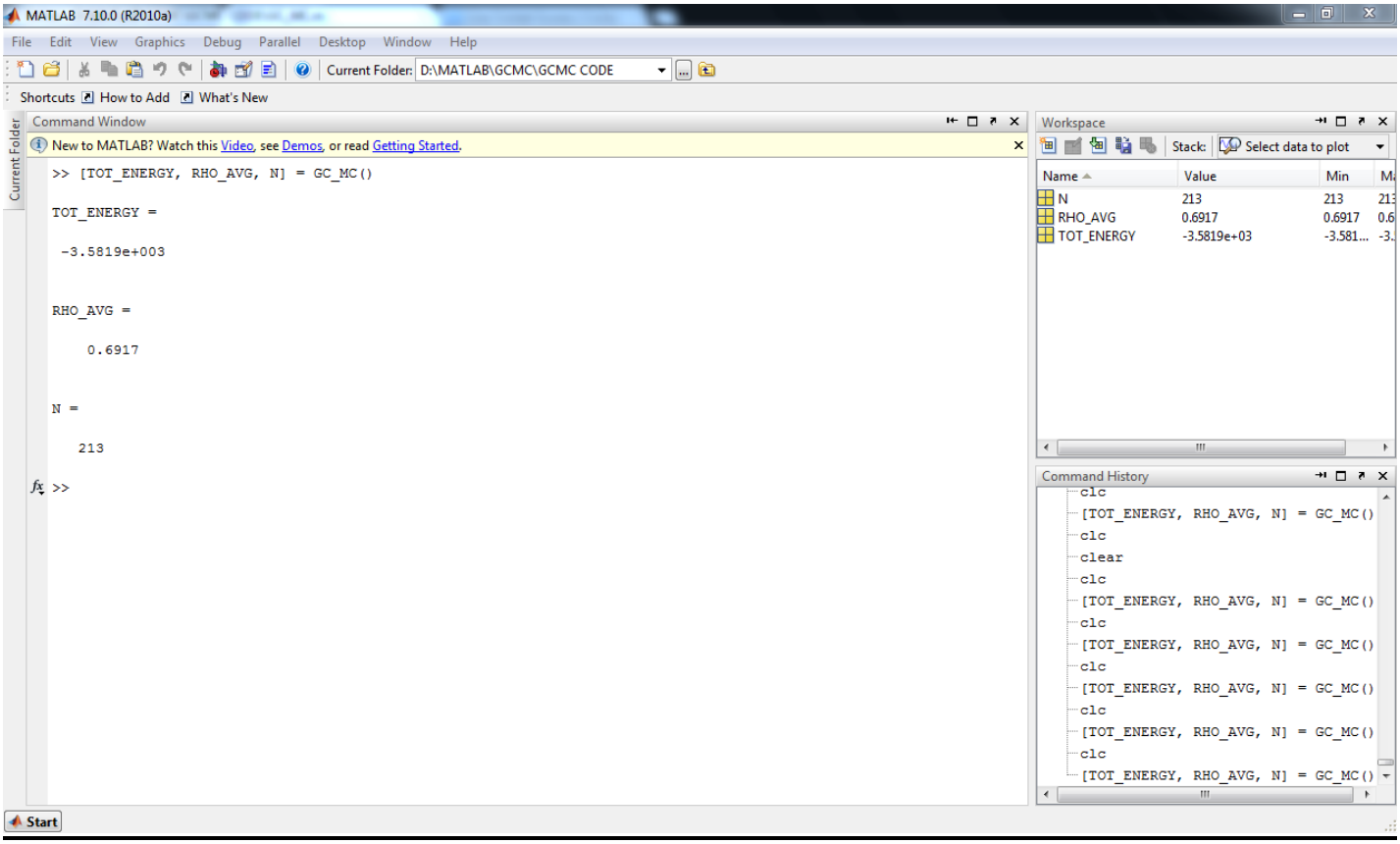
REMARKS

- The value of <U> was compared with the TOTAL ENERGY that was calculated separately for the configuration and it was found close to the <U> value that was calculated by updating the change in energy term at the end of each iteration.
- DELTAR (Displacement given to particle) value was adjusted so as to get a success ratio around 40-50%.
- Activity coefficient was initially chosen to be equal to ρ but later down it was corrected to get simulation values close to the prescribed value.
- Periodic Boundary Condition, cut-off radius and Nearest Neighbor conditions were also used.
- The adjusted value of Z (Activity) was 0.9.

REFERENCE

- A comparison of constant energy, constant temperature and constant pressure ensembles in molecular dynamics simulations of atomic liquids; D. Brown; J. H. R. Clarke; Chemistry Department, U.M.I.S.T., Manchester, England
- Computer Simulations of liquids by M.P.Allen and D.J. Tildesley, Clarendon Press, Oxford 1987
- Understanding Molecular Simulations from Algorithms to Applications, D.Frenkel and B.Smit, Academic Press, 2002


MATLAB 7.10.0 (R2010a)
File Edit View Graphics Debug Parallel Desktop Window Help
Current Folder: D:\MATLAB\GCMC\GCMC CODE
Shortcuts How to Add What's New
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> [TOT_ENERGY, RHO_AVG, N] = GC_MC()
TOT_ENERGY =
-3.5819e+003
RHO_AVG =
0.6917
N =
213
>>


MATLAB 7.10.0 (R2010a)
File Edit View Graphics Debug Parallel Desktop Window Help
Current Folder: D:\MATLAB\GCMC\GCMC CODE
Shortcuts How to Add What's New
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> [TOT_ENERGY, RHO_AVG, N] = GC_MC()
TOT_ENERGY =
-3.5819e+003
RHO_AVG =
0.6917
N =
213
>>
Workspace
Stack: Select data to plot
Name Value Min
N 213 213
RHO_AVG 0.6917 0.6917
TOT_ENERGY -3.5819e+03 -3.581...
Command History
clc
[TOT_ENERGY, RHO_AVG, N] = GC_MC()
clc
clear
clc
[TOT_ENERGY, RHO_AVG, N] = GC_MC()
clc
[TOT_ENERGY, RHO_AVG, N] = GC_MC()
clc
[TOT_ENERGY, RHO_AVG, N] = GC_MC()
clc
[TOT_ENERGY, RHO_AVG, N] = GC_MC()
clc
[TOT_ENERGY, RHO_AVG, N] = GC_MC()
Start