



Gibbs Ensemble

Ch 8

Method to study phase equilibria in a single simulation.

vapor-liquid and liquid-liquid equilibria.

not very efficient for very dense phases: like in the grand-canonical ensemble, the method does rely on a reasonable number of successful particle insertions to achieve compositional equilibrium.

condition for coexistence of two or more phases:

coexisting phases must have equal

- pressure ($P_I = P_{II}$)
- temperature ($T_I = T_{II}$)
- chemical potentials ($\mu_I = \mu_{II}$)

The simulation must find the values of the equilibrium P, T , and the density of both phases. → coexistence line

In the Gibbs ensemble there are 2 kind of particles (I, II) in two boxes.

The total number of particles and the total volume of the two boxes remain constant

The total system is at NVT conditions

Partition Function

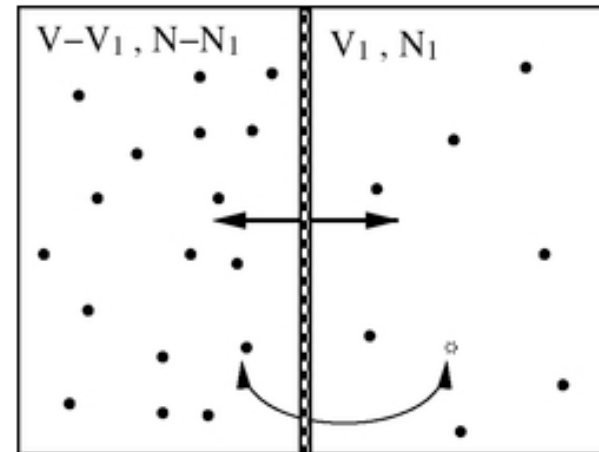
In Ch 5. we wrote an expression the partition function for a system of N particles distributed over two volumes V_1 and $V_2 = V - V_1$, where the particles interact with each other in volume 1 but behave like an ideal gas in volume 2

To derive the partition function of the grand-canonical ensemble (section 5.6), we assumed that the particles in volume V_2 behaved as ideal gas molecules.

$$Q(N, V_1, V_2, T) = \sum_{N_1=1}^N \frac{(V - V_1)^{N-N_1} V_1^{N_1}}{(N - N_1)! \Lambda^{3N} N_1!} \int_0^1 ds_2^{N-N_1} \int_0^1 ds_1^{N_1} \exp[-\beta U(s_1)]$$

Now we consider the case where:

- the particles in both volumes are subject to the same intermolecular interactions
- the volumes V_1 and V_2 can change in such a way that the total volume $V = V_1 + V_2$ remains constant



$$Q_G(N, V, T) = \sum_{N_1=1}^N \frac{1}{V(N - N_1)! \Lambda^{3N} N_1!} \int_0^V dV_1 (V - V_1)^{N - N_1} V_1^{N_1} \times \\ \int_0^1 ds_2^{N - N_1} \exp[-\beta U(s_2)] \int_0^1 ds_1^{N_1} \exp[-\beta U(s_1)]$$

the probability distribution is:

$$N(N_1, V_1, s_1^{N_1}, s_2^{N-N_1}) = \frac{(V - V_1)^{N-N_1} V_1^{N_1}}{(N - N_1)! \Lambda^{3N} N_1!} \exp[-\beta(U(s_2) + U(s_1))]$$

Monte Carlo scheme: Trial moves are:

1. Displacement of a randomly selected particle.
2. Change of the volume in such a way that the total volume remains constant.
3. Transfer of a randomly selected particle from one box to the other

Acceptance rules

detailed balance: $K(o \rightarrow n) = K(n \rightarrow o)$

with $K(o \rightarrow n) = N(o) \times \alpha(o \rightarrow n) \times \text{acc}(o \rightarrow n)$.

Particle Displacement

the state n is obtained from state o via the displacement of a randomly selected particle in box I .

$$\frac{N(n)}{N(o)} = \frac{\exp(-\beta U_n)}{\exp(-\beta U_o)}$$

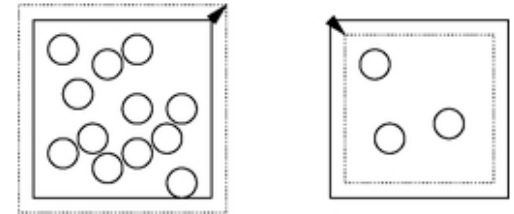
$$\text{acc}(o \rightarrow n) = \text{Min}\{I, Nn/No\}$$

Volume Change

For a change of the volume of box I by an amount dV , the ratio of the statistical weights are:

$$\frac{Nn}{No} = \frac{(V - V_{1n})^{N-N_1} V_{1n}^{N_1}}{(V - V_{1o})^{N-N_1} V_{1o}^{N_1}} \exp[-\beta(U_n - U_o)]$$

$$\text{acc}(o \rightarrow n) = \text{Min}\{1, Nn/No\}$$



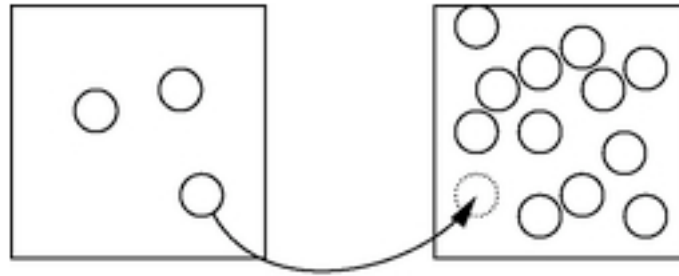
The volume change can be made as a random walk in $\ln[V/(V - V_1)]$ instead of in V_1

the acceptance rule for the volume has to be modified.

$$\text{acc}(o \rightarrow n) = \min\left\{1, \frac{(V - V_{1n})^{N-N_1+1} V_{1n}^{N_1+1}}{(V - V_{1o})^{N-N_1+1} V_{1o}^{N_1+1}} \exp[-\beta(U_n - U_o)]\right\}$$

Particle Exchange

The new configuration is generated from the old one (N_1 particles in box 1) by removing a particle from box 1 and inserting this particle in box 2.



$$acc(o \rightarrow n) = \frac{N_1(V - V_1)}{(N - N_1 + 1)V_1} \exp[-\beta(U_n - U_o)]$$

Implementation

Npart attempts to displace a (randomly selected) particle in one of the (randomly chosen) boxes

Nvol attempts to change the volume of the subsystems

Nswap attempts to exchange particles between the boxes (initial box and particle selected randomly).

NOT IN ORDER: randomly select which move is attempted.

```
PROGRAM mc_Gibbs
```

Gibbs ensemble simulation

```
do icycl=1,ncycl
```

perform `ncycl` MC cycles

```
  ran=ranf()*(npart+nvol+nswap)
```

```
  if (ran.le.npart) then
```

```
    call mcmove
```

attempt to displace a particle

```
  else if (ran.le.(npart+nvol))
```

```
    call mcvol
```

attempt to change the volume

```
  else
```

```
    call mcswap
```

attempt to swap a particle

```
  endif
```

```
  call sample
```

sample averages

```
enddo
```

```
end
```

- On average, we perform per cycle `npart` attempts to displace
- `npart` attempts to displace particles, `nvol` attempts to change the volume, and `nswap` attempts to swap particles between the two boxes.
- Subroutine MCMOVE attempts to displace a randomly selected particle

PROGRAM mc_Gibbs	Gibbs ensemble simulation
do icycl=1,ncycl	perform ncycl MC cycles
ran=ranf()*(npart+nvol+nswap)	
if (ran.le.npart) then	
call mcmove	attempt to displace a particle
else if (ran.le.(npart+nvol))	
call mcvol	attempt to change the volume
else	
call mcswap	attempt to swap a particle
endif	
call sample	sample averages
enddo	
end	

- The subroutine MCVOL attempts to change the volume of the two boxes
- the subroutine MCSWAP attempts to swap a particle between the two boxes
- subroutine SAMPLE samples the ensemble averages.

SUBROUTINE mcvol	attempt to change
	the volume
call toterg(box1,en1o)	energy old conf. box 1
call toterg(box2,en2o)	and 2 (box1: box length)
vol=box1**3	old volume box 1 and 2
vo2=v-vol	
lnvn=log(vol/vol2)+	random walk in $\ln V_1/V_2$
+ (ranf()-0.5)*vmax	
v1n=v*exp(lnvn)/(1+exp(lnvn))	new volume box 1 and 2
v2n=v-v1n	
box1n=v1n**(1/3)	new box length box 1
box2n=v2n**(1/3)	new box length box 2
do i=1,npart	
if (ibox(i).eq.1) then	determine which box
fact=box1n/box1o	
else	
fact=box2n/box2o	
endif	
x(i)=x(i)*fact	rescale positions
enddo	
call toterg(box1n,en1n)	total energy box 1

- The term $\text{ibox}(i) = 1$ indicates that particle i is in box 1;
- $\text{npart} = \text{npbox}(1) + \text{npbox}(2)$ where $\text{npbox}(i)$ gives the number of particles in box i .
- In this algorithm we perform a random walk in $\ln V$

do i=1,npart	REMOVED
if (ibox(i).eq.) then	determine which box
fact=box1o/box1n	

fact=box2n/box2o	
endif	
x(i)=x(i)*fact	rescale positions
enddo	
call toterg(box1n,en1n)	total energy box 1
call toterg(box2n,en2n)	total energy box 2
arg1=-beta*((en1n-en1o)+	
+ (npbox(1)+1)*log(v1n/v1o)/beta)	appropriate weight function
arg2=-beta*((en2n-en2o)+	acceptance rule (8.3.3)
+ (npbox(2)+1)*log(v2n/v2o)/beta)	
if (ranf().gt.exp(arg1+arg2)) then	
do i=1,npart	REJECTED
if (ibox(i).eq.) then	determine which box
fact=box1o/box1n	
else	
fact=box2o/box2n	
endif	
x(i)=x(i)*fact	restore old configuration
enddo	
endif	
return	
end	

The subroutine toterg calculates the total energy of one of the two boxes.

SUBROUTINE mcswap	attempts to swap a particle between the two boxes
if (ranf().lt.0.5) then	which box to add or remove
in=1	
out=2	
else	
in=2	
out=1	
endif	
xn=ranf()*box(in)	new particle at a random position
call ener(xn,enn,in)	energy new particle in box in
w(in)=w(in)+vol(in)*	update chemical potential (8.3.5)
+ exp(-beta*enn)/(npbox(in)+1)	
if (npbox(out).eq.0) return	if box empty return
ido=0	find a particle to be removed
do while (ido.ne.out)	

The subroutine ener calculates the energy of a particle at the given position and box.

At the end of the simulation, the chemical potential can be calculated from $w(\text{box})$ using $\mu_{\text{box}} = -\ln \langle w_{\text{box}} \rangle / \beta$.

x(o)=xn	add new particle to box in
ibox(o)=in	
nbox(out)=npbox(out)-1	


```

    in=1
    out=2
else
    in=2
    out=1
endif
xn=ranf()*box(in)
call ener(xn,enn,in)
w(in)=w(in)+vol(in)*
+ exp(-beta*enn)/(npbox(in)+1)
if (npbox(out).eq.0) return
ido=0
do while (ido.ne.out)
    o=int(npart*ranf())+1
    ido=ibox(o)
enddo
call ener(x(o),eno,out)
arg=exp(-beta*(enn-eno +
+ log(vol(out)*(npbox(in)+1)/
+ (vol(in)*npbox(out))))/beta))
if (ranf().lt.arg) then
    x(o)=xn
    ibox(o)=in
    nbox(out)=npbox(out)-1
    nbox(in)=npbox(in)+1
endif
return
end

```

new particle at a random position
 energy new particle in box in
 update chemical potential (8.3.5)

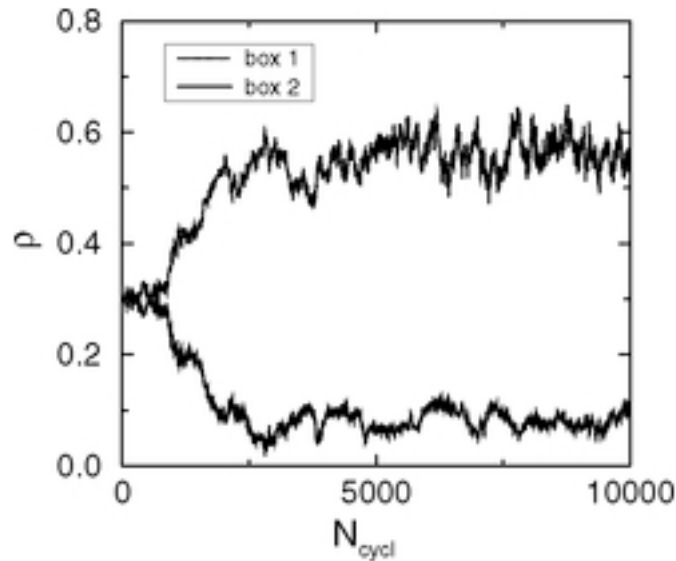
if box empty return
 find a particle to be removed

finds a particle in box "out"

energy particle o in box out

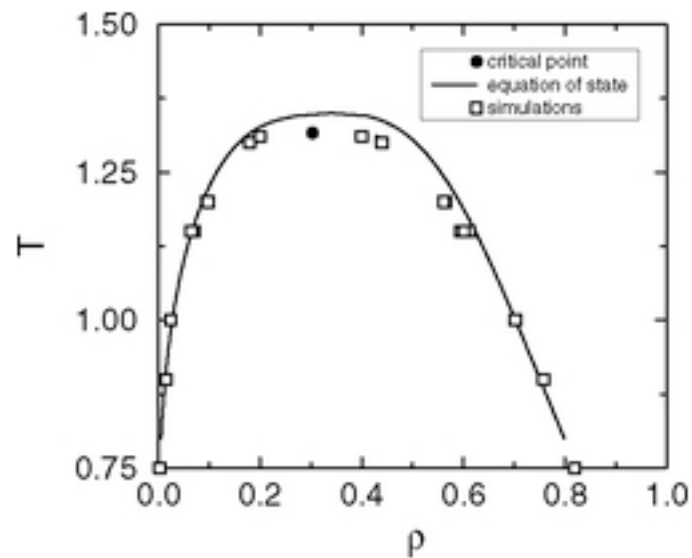
acceptance rule (8.3.4)

add new particle to box in



- the density of the fluid in the two boxes as a function of the number of Monte Carlo cycles
- The simulation was started with equal density in both boxes.
- During the first 1000 cycles, the system has not yet “decided” which box would evolve to a liquid density and which box to a gas density.
- After 5000 cycles, the system already has reached equilibrium and the coexistence properties can be determined.

Phase diagram of the Lennard-Jones



The Gibbs ensemble simulations give as the estimate for the critical point $T_c = 1.316 \pm 0.006$ $\rho_c = 0.304 \pm 0.006$.