Classical Monte Carlo simulation of Many body Systems

✓ Detailed balance condition and Metropolis algorithm

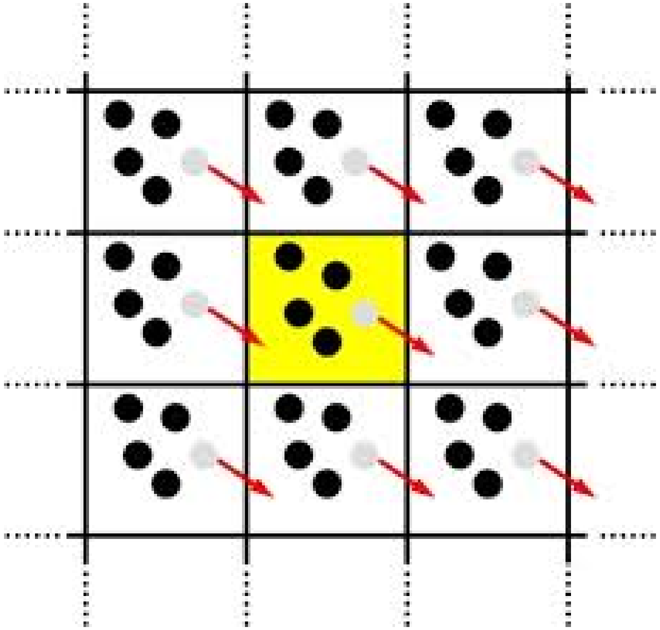
✓ Calculation of properties

Classical Monte Carlo simulation of Many Body systems

- N atoms/molecules in a box
- With periodic boundary conditions
- A molecular model& force field is assumed
- Several configurations are generated at random

  using a Metropolis algorithm
- Properties are calculated *within the framework*

  of Statistical Physics



Where the Monte Carlo method (MC) comes in

Some concepts from statistical mechanics

If a system in equilibrium can be in N states,
the probability of the system having energy E_n is:

$$\frac{1}{Q}\exp(-E_n/k_BT)$$

D. A. McQuarrie,
“Statistical Mechanics”,
University Science Books, 2000

Where Q is the partition function:

$$Q = \sum_{n=1}^{N} \exp(-E_n/k_BT)$$

The expected value of a observable A is:

$$< A >= \frac{1}{Q}\sum_{|i>} \exp(-E_i/k_BT) < i|A|i >$$

✓ The classical expression for the partition function Q

of a systems of N identical atoms is:

$$Q = \frac{1}{h^{3N} N!} \int d\vec{p}^{\,N} d\vec{r}^{\,N} \exp(-H(\vec{r}^{\,N}, \vec{p}^{\,N})/k_B T)$$

Where $\vec{r}^N$, $\vec{p}^N$ stand for the coodinates and momenta of all N particles and H($\vec{r}^N$, $\vec{p}^N$) is the hamiltonian:

$$H(\vec{r}^{\,N}, \vec{p}^{\,N}) = \sum_{i=1}^{N} \frac{\vec{p}_i^{\,2}}{2m} + U(\vec{r}^{\,N})$$

The classical expression for <*A*> is:

$$< A >= \frac{\int d\vec{p}^{\,N} d\vec{r}^{\,N} A(\vec{p}^{\,N}, \vec{r}^{\,N}) \exp(-\beta H(\vec{r}^{\,N}, \vec{p}^{\,N})}{\int d\vec{p}^{\,N} d\vec{r}^{\,N} \exp(-\beta H(\vec{p}^{\,N}, \vec{r}^{\,N}))}$$

$A(\vec{p}^{\,N}) \rightarrow$ analytical computation (easy)

$A(\vec{r}^{\,N})$ → numerical computation (very difficult and costly !!)

✓ We must then focus on the calculation of averages of the type:

$$\boxed{< A >= \frac{\int d\vec{r}^{\,N} A(\vec{r}^{\,N}) \exp(-\beta U(\vec{r}^{\,N}))}{\int d\vec{r}^{\,N} \exp(-\beta U(\vec{r}^{\,N}))}}$$

$$\beta = \frac{1}{k_B T}$$

$$\exp(-\beta U(\vec{r}^{\,N})) \equiv \text{Boltzmann factor}$$

- We denote the configurational part of the partition function by Z:

$$Z \equiv \int d\vec{r}^{\,N} \exp(-\beta U(\vec{r}^{\,N}))$$

- The probability density of finding the system in a configuration around $\vec{r}^N$ *is*

$$\boxed{\mathcal{N}(\vec{r}^{\,N}) = \frac{\exp(-\beta U(\vec{r}^{\,N}))}{Z}}$$

The Metropolis method

- Let’s go back to the averages we are interested in:

$$< A >= \frac{\int d\vec{r}^{\,N} A(\vec{r}^{\,N}) \exp(-\beta U(\vec{r}^{\,N}))}{\int d\vec{r}^{\,N} \exp(-\beta U(\vec{r}^{\,N}))}$$

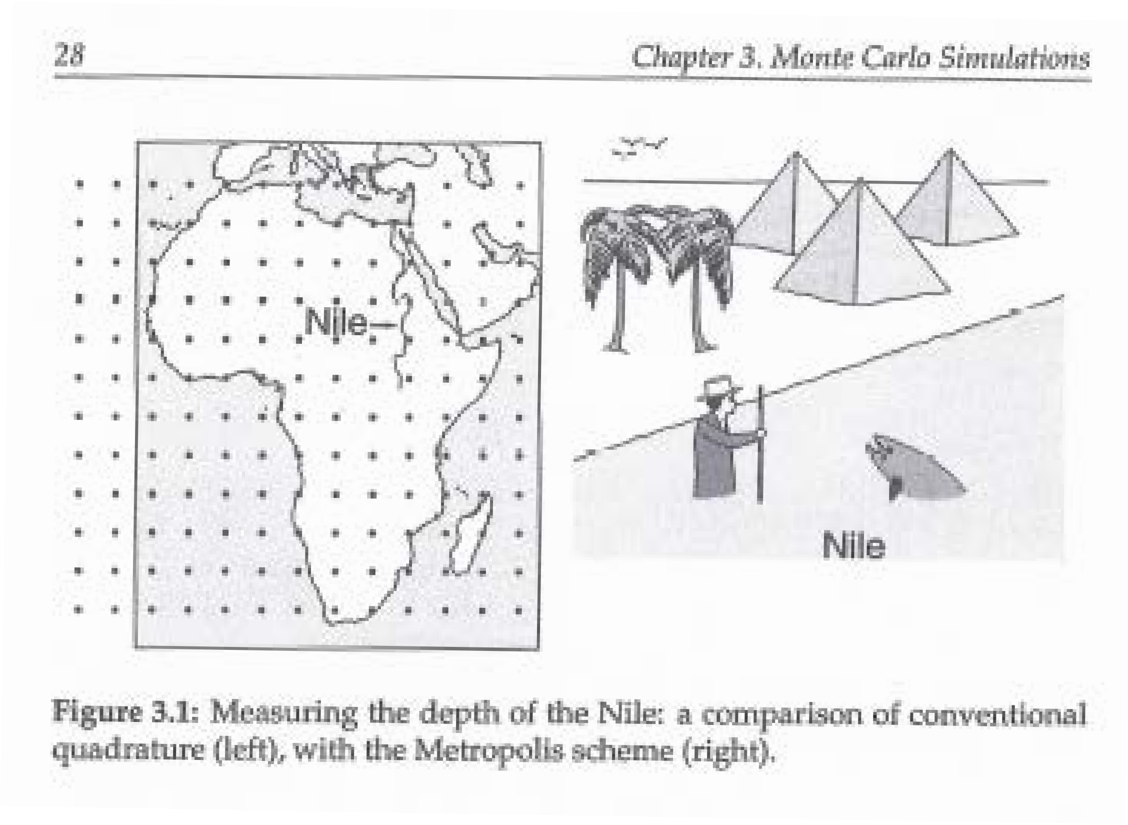
- This is in fact the ratio of two integrals. In a pioneering work, Metropolis *et al.* showed that it is possible to devise an efficient Monte Carlo squeme to sample such a ratio:

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. N. Teller and E. Teller, “Equation of state calculations by fast computing machines”,
J. Chem. Phys. 21, 1087-1092 (1953)

✓ The probability density of finding the system in a configuration around $\vec{r}^N$ *is*

$$\mathcal{N}(\vec{r}^{\,N}) \propto \exp(-\beta U(\vec{r}^{\,N}))$$

Look at the picture to illustrate the idea of the Metropolis method:



28 Chapter 3. Monte Carlo Simulations

Figure 3.1: Measuring the depth of the Nile: a comparison of conventional quadrature (left), with the Metropolis scheme (right).

From Frenkel&Smit book

Let ‘s consider how to generate points in configuration space with a relative probability proportional to the Boltzmann factor:

- We prepare the system in a configuration $\vec{r}^N$, which we denote by o (old), that has a nonvanishing Boltzmann factor $\exp[-\beta U(o)]$

- Next, we generate a new trial configuration $\vec{r}'^N$, which we denote by n (new), by adding a small random displacement Δ

  The Boltzman factor of this trial configuration is $\exp[-\beta U(n)]$

- We must decide whether we accept or reject the trial configuration

- Detailed balance condition:

In equilibrium, the average number of accepted trial moves that result leaving state o must be exactly equal to the number of accepted trial moves from all other states n to state o

$$\boxed{\mathcal{N}(o)\pi(o \to n) = \mathcal{N}(n)\pi(n \to o)}$$

$\mathcal{N}(o)$: probability of being in configuration o

$\pi(o \to n) \equiv$ Transition probability from configuration o to n

It can be written as:

$$\pi(o \to n) = \alpha(o \to n) \times \mathrm{acc}(o \to n)$$

$\alpha(o \to n) \equiv$ Transition matrix that determines the probability of performing a trial move from o to n

$\mathrm{acc}(o \to n) \equiv$ Probability of accepting a trial move from o to n

If we ssume that α is symmetric (in general, it is not necesary):

$$\alpha(o \to n) = \alpha(n \to o)$$

The detailed balance condition can be written as:

$$\mathcal{N}(o)\ \mathrm{acc}(o \to n) = \mathcal{N}(n)\ \mathrm{acc}(n \to o)$$

From this equation if follows:

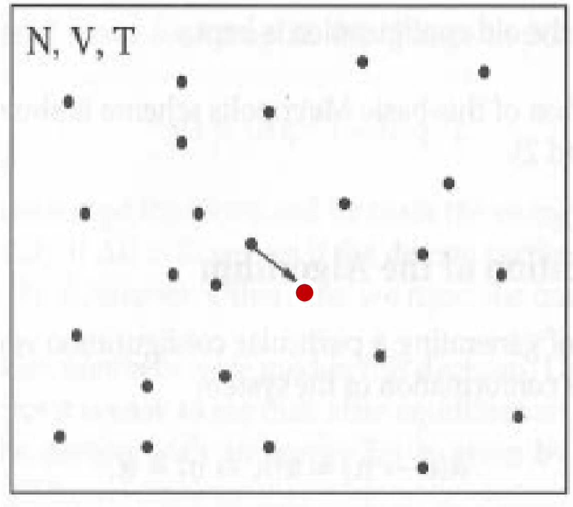
$$\frac{\mathrm{acc}(o \to n)}{\mathrm{acc}(n \to o)} = \frac{\mathcal{N}(n)}{\mathcal{N}(o)} = \exp\{-\beta[U(n) - U(o)]\}$$

The Metropolis *et al*. choice:

$$\boxed{\begin{array}{ll} \mathrm{acc}(o \to n) = \dfrac{\mathcal{N}(n)}{\mathcal{N}(o)} & \text{if} \quad \mathcal{N}(n) < \mathcal{N}(o) \\ \mathrm{acc}(o \to n) = 1 & \text{if} \quad \mathcal{N}(n) \geq \mathcal{N}(o) \end{array}}$$

A basic Monte Carlo Algorithm

1. Select a particle at random, and calculate its energy $U(\vec{r}^N)$.



2. Give the particle a random displacement, $r'=r+\Delta$, and calculate its new energy $U(\vec{r}'^N)$.

3. Accept the move from $\vec{r}^N$ to $\vec{r}'^N$ with probability

$$\mathrm{acc}(\mathrm{o} \to \mathrm{n}) = \min\left(1, \exp(-\beta[U(\vec{r}\,'^N) - U(\vec{r}^{\,N})]\right)$$

This means:

$U(\vec{r}\,'^N) \leq U(\vec{r}^{\,N}) \Rightarrow$ We accept the trial

$U(\vec{r}\,'^N) > U(\vec{r}^{\,N}) \Rightarrow$ We generate a random number from a uniform distribution [0,1]

- If smaller than $exp(-\beta[U(\vec{r}'^N) - U(\vec{r}^N)]) \Rightarrow$ We accept the trial
- If larger than $exp(-\beta[U(\vec{r}'^N) - U(\vec{r}^N)]) \Rightarrow$ We reject the trial

Trial moves

How to generate a trial displacement:

$$\boxed{\begin{aligned} x_i' &\rightarrow x_i + \Delta(Randf - 0.5) \\ y_i' &\rightarrow y_i + \Delta(Randf - 0.5) \\ z_i' &\rightarrow z_i + \Delta(Randf - 0.5) \end{aligned}}$$

Randf: a random number uniformly distributed from [0,1]

A criterium to choose the value of Δ:

to obtain an acceptance of 50 %

Basic Monte Carlo Algorithm

```
  PROGRAM mc                          basic Metropolis algorithm

  Do icycl=1,ncycl                    perform ncycl  MC cycles
  call mcmove                         displace a particle
  if (mod(icycl,nsamp).eq.0)
+   call sample                       sample averages
  enddo
  end
```

Attempt to displace a particle

```
  SUBROUTINE mcmove                      Attempts to displace a particle

  o=int (ranf () *npart)+1               select a particle at random
  call ener(x(o),eno)                    energy old configuration
  xn = x(o)+ (ranf ()-0.5)*delx          give particle random displacement
  call ener(xn,enn)                      energy new configuration
  if (ranf( ).lt.exp(-beta               acceptance rule
+   *(enn-eno)) x(o)=xn                  acceptance replace x(o) by xn
  return
  end
```

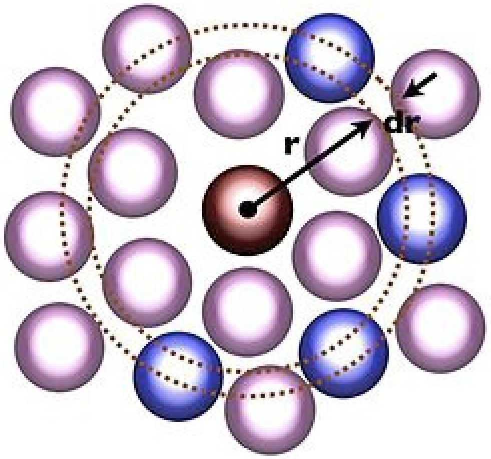
The ranf( ) is a random number uniform in [0,1]

Simulating a Many Body system

- Structure: The radial distribution function (RDF)

- The **radial distribution function (RDF)** describes how density varies as a function of distance from a reference particle . It can be defined as

$$g(r) = \frac{1}{\rho}\frac{n(r)}{4\pi r^2 dr} = \frac{n(r)}{n_{id}(r)}$$



calculation of g(r)

$\rho = \frac{N}{V}$ average number density of particles

$n(r)$ Mean number of particles within a distance of r and r+dr away from a particle.

$n_{id}(r)$ Mean number of particles ….. for an ideal gas

- The RDF is usually determined by calculating the distance between all particle pairs and binning them into a histogram.

Algorithm for the calculation of the RDF

```
                                                   radial distribution function
 subroutine gr (switch)
                                                   switch = 0 initialization,
                                                    = 1 sample, and =2 results
 if (switch. eq.0) then                             initialization
   ngr = 0
   delg=box/(2*nhis)                                 bin size
   do i = 1,nhis                                    nhis = total number of bins
       g(i) = 0
   enddo
else if (switch.eq.1) then                         sample
 ngr = ngr + 1
 do i=1,npart-1
     do j=i+1,npart                               loop over all pairs
         xr = x(i)-x(j)                !!
         xr = xr-box*nint(xr/box)      !!          periodic boundary conditions
         r = sqrt(xr**2)               !!
         if (r.lt.box/2)  then                     only within half the box length
            ig = int(r/delg)
            g(ig) = g(ig) +2                       contribution for particle i and j
         endif
      enddo
  endo
```

Algorithm for the calculation of the RDF *cont.*

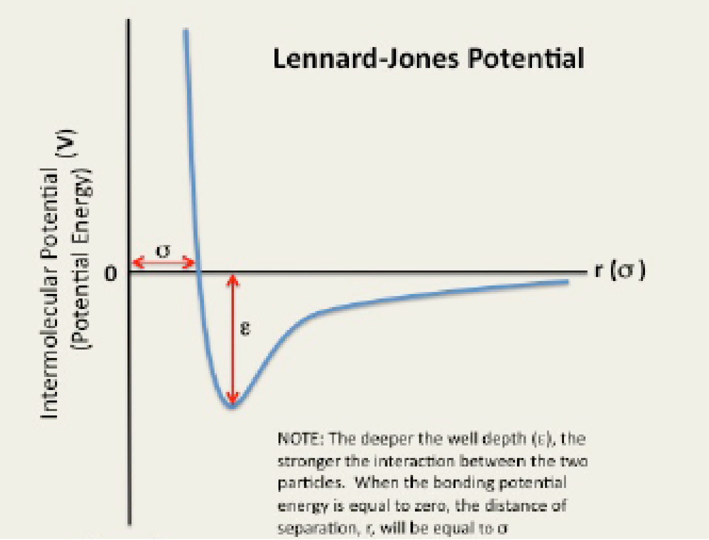
```
      else if (switch.eq.2) then                          determine g(r)
      do i=1,nhis
         r=delg*(i + 0.5)                                   distance r
         vb = ((i+1)**3-i**3)*delg**3               volume between bin i+1 and i
         nid = (4/3)*pi*vb*rho                      number of ideal gas part . in vb
        g(i) =g(i) /(ngr*npart*nid)                    normalize  g(r)
       enddo
       endif
    endif
     return
     end
```

Comment : The factor pi=3.14159 ….

From Frenkel&Smit book

Lennard-Jones interaction potential

$$U_{LJ}(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$



Lennard-Jones Potential

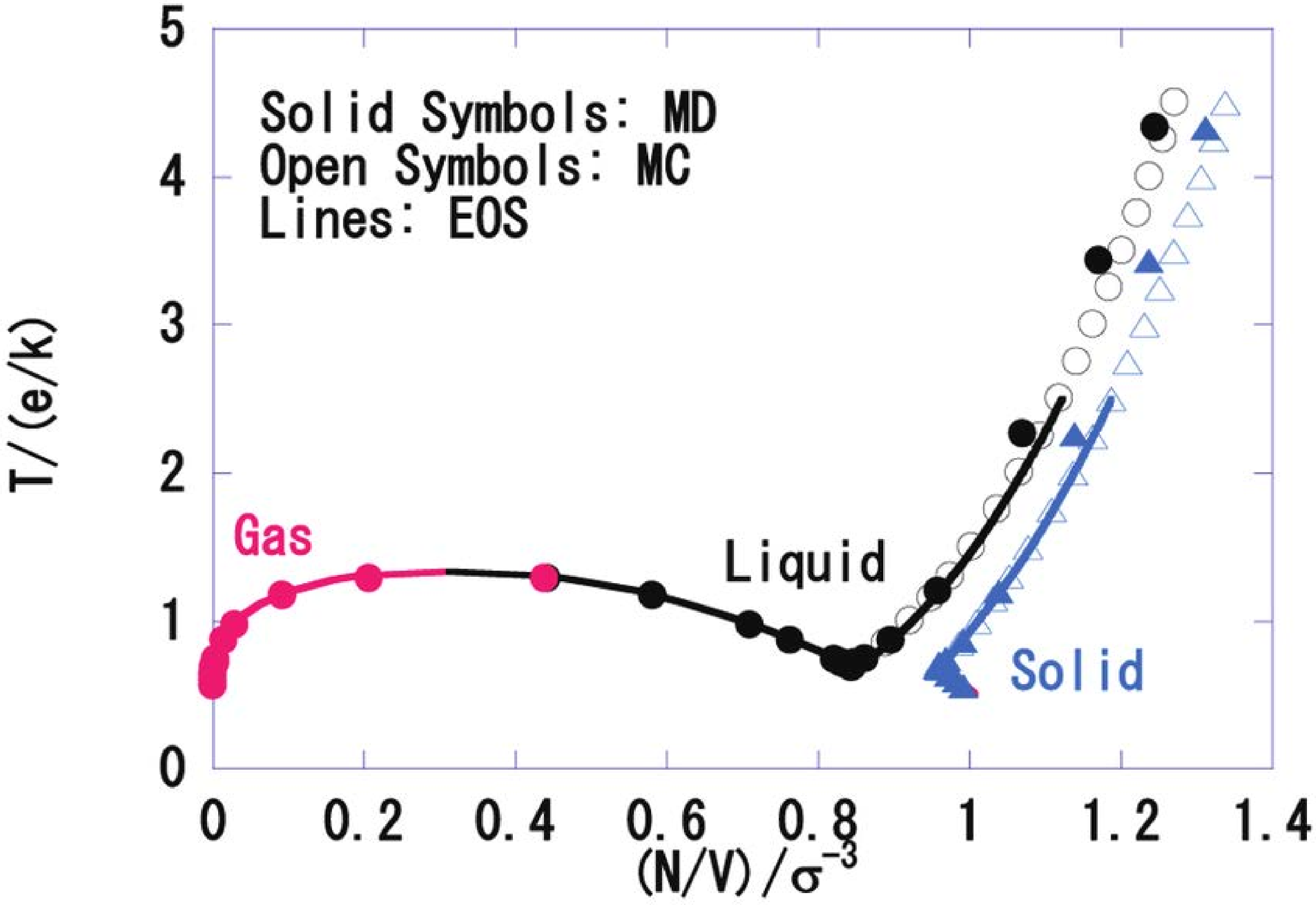
NOTE: The deeper the well depth (ε), the stronger the interaction between the two particles. When the bonding potential energy is equal to zero, the distance of separation, r, will be equal to σ

In reduced units:

$$U_{LJ}^{*}(r^{*}) = 4\left[\left(\frac{1}{r^{*}}\right)^{12} - \left(\frac{1}{r^{*}}\right)^{6}\right]$$

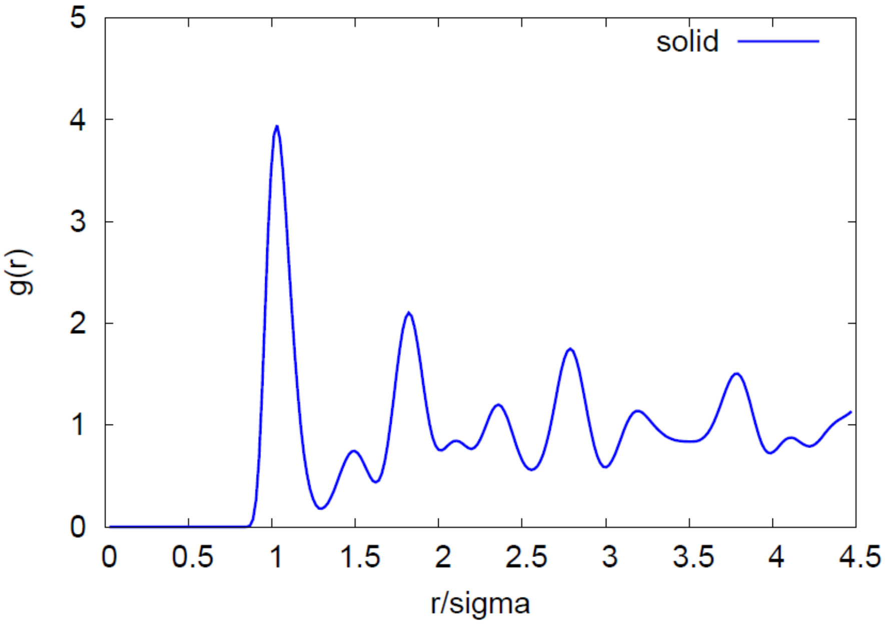
$$U_{LJ}^{*} = \frac{U_{LJ}}{\varepsilon} \qquad r^{*} = \frac{r}{\sigma}$$

Phase diagram of a L-J system: Comparison between theoretical predictions and computer simulation results

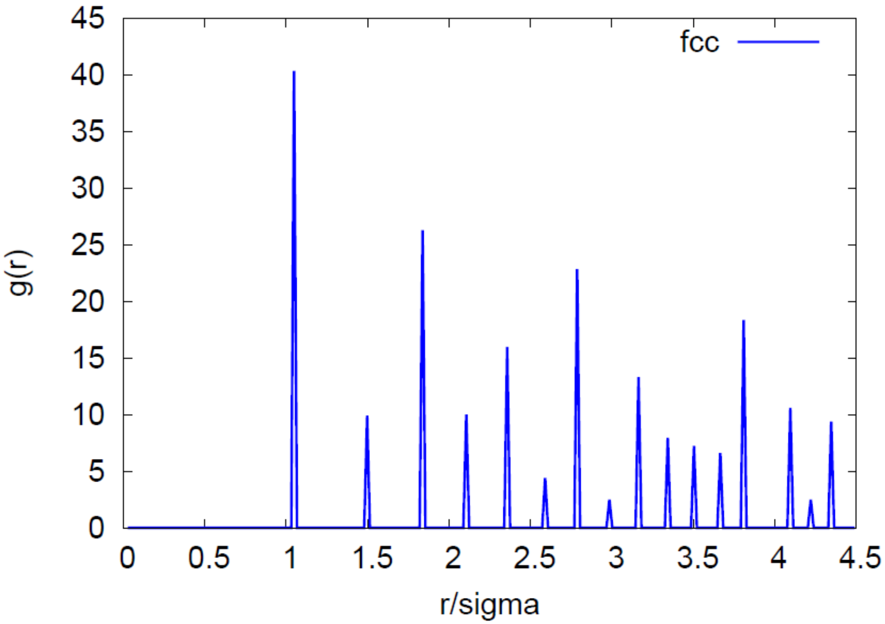


from K. Yosuke, Y. Yuri, J. Comput. Chem. Jpn. 12, 2 (2013)

The radial distribution function: comparison between a solid and a fcc lattice

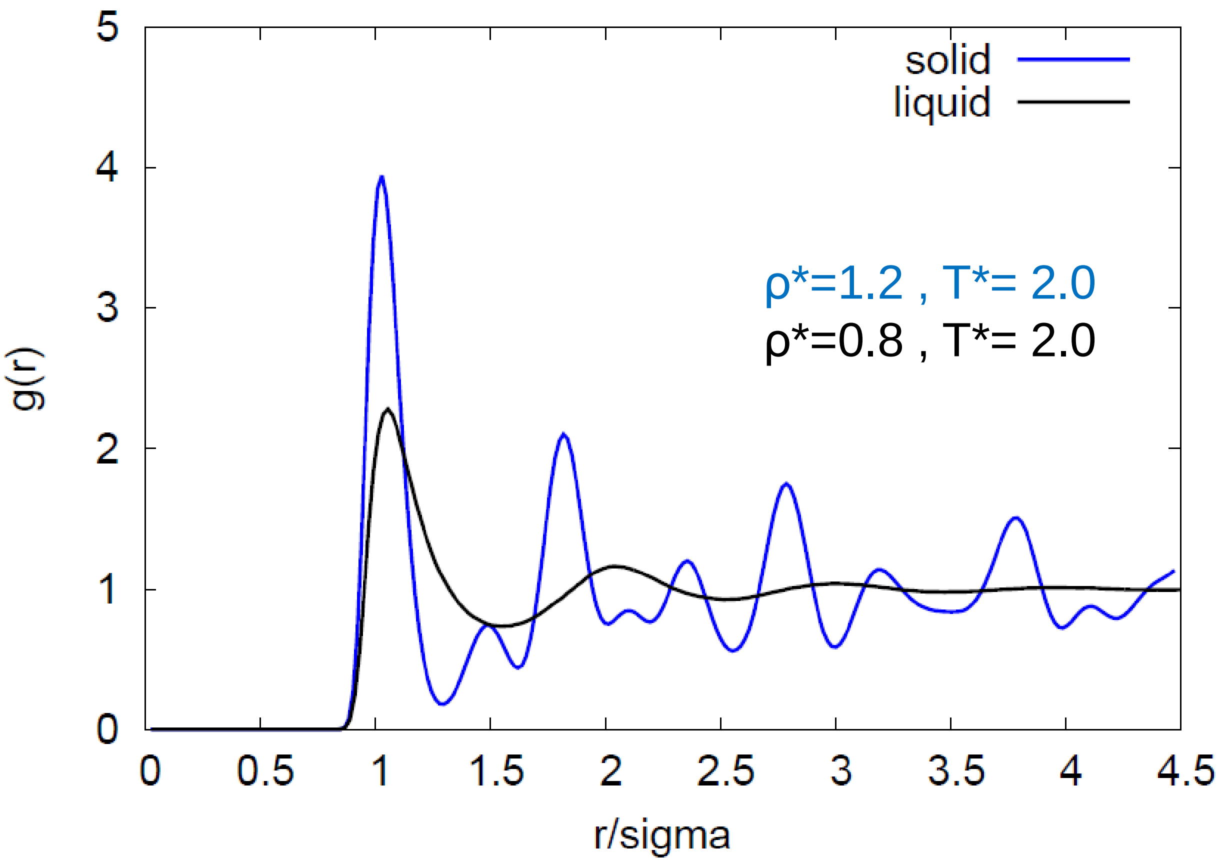


ρ*=1.2 , T*= 2.0



ρ*=1.2

The radial distribution function: comparison between a solid and a liquid



The solid exhibits a long range order !!

Simulating a Many Body system

- Thermodynamic properties: Energy

- **Energy**

$$E = \frac{3}{2} N k_B T + < U_N >$$

- **Potential Energy (*interaction energy*)**

$$< U_N > = < U_N(\vec{r}_1, \vec{r}_2, \cdots, \vec{r}_N) >$$

We assume pairwise interactions and we truncate the potential at a distance r_c :

$$< U_N > = < \sum_{i=1}^{N} \sum_{j>i, r_{ij}<r_c}^{N} U(r_{ij}) >$$