9. Importance sampling

Metropolis

Rouse algorithm

Norman-Filinov algorithm for grand canonical ensemble

Finite size problem, periodic boundary conditions

Importance Sampling

- Suppose we select $\underset{\approx}{X}_l$ (points in phase space according to some probability $P(\underset{\approx}{X}_l)$.

To obtain the thermal average of some property which depends on the configuration $\underset{\approx}{X}$ of the chain. [$\underset{\approx}{X}_l$ is a point in phase space

$$\langle A(\underset{\approx}{X})\rangle = \frac{\sum_{l=1}^{M} \exp^{-E(\underset{\approx}{X}_l)/k_BT} \, A(\underset{\approx}{X}_l)/P(\underset{\approx}{X}_l)}{\sum_{l=1}^{M} \exp^{-E(\underset{\approx}{X}_l)/k_BT}/P(\underset{\approx}{X}_l)}$$

selected according to some probability $P(\underset{\approx}{X}_l)$]

The best choice is one in which

$$P(\underset{\approx}{X}_l) \propto \exp^{-E(\underset{\approx}{X}_l)/k_BT}$$

then we can replace the weighted average above by a simple arithmetic average:

$$\langle A(\underset{\approx}{X})\rangle = \frac{1}{M}\sum_{l=1}^{M} A(\underset{\approx}{X}_l)$$

How to do it?

Metropolis suggested:

Construct a Markov process where each state $\underset{\approx}{X}_{l+1}$ is constructed from a previous state $\underset{\approx}{X}_l$ via a suitable transition probability $W(\underset{\approx}{X}_l \rightarrow \underset{\approx}{X}_{l+1})$

It is possible to choose the transition probability W such that in the limit $M \rightarrow \infty$ the distribution function $P(\underset{\approx}{X}_l)$

of the states generated by this Markov process tends toward the equilibrium distribution:

$$\left[P(\underset{\approx}{X}_\ell) \right]_{\text{equil}} = \frac{1}{Z} \exp - E(\underset{\approx}{X}_\ell)/k_B T$$

as it should.

To achieve this, impose the principle of detailed balance

$$\left[P(\underset{\approx}{X}_\ell) \right]_{\text{equil}} W(\underset{\approx}{X}_\ell \rightarrow \underset{\approx}{X}_{\ell'}) = \left[P(\underset{\approx}{X}_{\ell'}) \right]_{\text{equil}} W(\underset{\approx}{X}_{\ell'} \rightarrow \underset{\approx}{X}_\ell)$$

That is, the ratio of transition probabilities for a "move" $\underset{\approx}{X}_\ell \rightarrow \underset{\approx}{X}_{\ell'}$ and the inverse move $\underset{\approx}{X}_{\ell'} \rightarrow \underset{\approx}{X}_\ell$ depend only on the energy change

$$\Delta E = U(\underset{\approx}{X}_{\ell'}) - U(\underset{\approx}{X}_\ell)$$

$$\frac{W(\underset{\approx}{X}_\ell \rightarrow \underset{\approx}{X}_{\ell'})}{W(\underset{\approx}{X}_{\ell'} \rightarrow \underset{\approx}{X}_\ell)} = \exp(-\Delta E/k_B T)$$

This does not specify the choice of $W(\underset{\approx}{X}_\ell \rightarrow \underset{\approx}{X}_{\ell'})$ uniquely.

A frequently used choice that satisfies this is:

$$W(\underset{\approx}{X}_\ell \rightarrow \underset{\approx}{X}_{\ell'}) = \begin{cases} \exp(-\Delta E/kT) & \text{if } \Delta E > 0 \\ 1 & \text{otherwise} \end{cases}$$

Metropolis proved that such a sequence of states (Markov chain)

$$\underset{\approx}{X}_\ell \rightarrow \underset{\approx}{X}_{\ell'} \rightarrow \underset{\approx}{X}_{\ell''} \rightarrow \cdots$$

actually has the property that its probability distribution $P(\underset{\approx}{X}_\ell)$ converges toward the equilibrium one, $P_{equil}(\underset{\approx}{X}_\ell) = \frac{1}{Z} \exp\left(\frac{-E(\underset{\approx}{X}_\ell)}{kT}\right)$

How to generate the Markov chain of M events:

In practice, what does the "move"

$$\underset{\approx}{X}_\ell \rightarrow \underset{\approx}{X}_{\ell'}$$

mean? Enormous freedom of choice of this move, subject only to 2 restrictions:

(a) $W_{\Delta E=0}(\underset{\approx}{X}_\ell \rightarrow \underset{\approx}{X}_{\ell'}) = W_{\Delta E=0}(\underset{\approx}{X}_{\ell'} \rightarrow \underset{\approx}{X}_\ell)$

(b) $W(\underset{\approx}{X}_\ell \rightarrow \underset{\approx}{X}_{\ell'})$ in the presence of the energy change ΔE should yield values significantly different from 0 and 1 sufficiently often.

Therefore, one mostly performs moves where only one (or a few) degrees of freedom is (are) changed. Otherwise condition (b) cannot be satisfied. We want the average acceptance rate for the trial moves to be not too small. Ideally the average acceptance rate is 0.5

Examples:

1. "Slithering snake" (reptation) algorithm: start from a SAW configuration of a long chain: choose a random end, remove one bond from this end of the chain and add a bond in a random direction at the other end, accept this move only if it does not violate the SAW constraints

2. Rouse algorithm for relaxation of a chain in a heat bath:

Randomly choose a group of neighboring bonds along the chain, randomly flip them to new positions on the lattice; accept this move only if it does not violate the SAW constraints or the "entanglement restriction" (i.e., during the random motion of the links of a chain the chain must not intersect itself.) Also, the end bonds may rotate to new positions

3. Norman-Filinov algorithm for grand-canonical Ensemble: (constant μ, V, T) equal probability for 3 moves:

$$2[\text{create}/\text{annihilate}] + \text{displace}$$

Transition $\rightarrow$ P_{acc} Probability

$$P_{acc} = \begin{cases} \min\left[1,\ \underline{\exp(-\Delta E/k_BT)}\right] & \dfrac{\Delta E}{k_BT} \leq 180 \\ 0 & \geq 180 \end{cases}$$

min means choose the lower one of these two

CREATE: $\Delta E = \Delta U_{ji}(\underset{\approx}{r}^N) + k_BT \ln\left(\dfrac{N+1}{V\rho^0}\right) - \mu$

ANNIH: $\Delta E = \Delta U_{ji}(\underset{\approx}{r}^N) - k_BT \ln\left(\dfrac{N}{V\rho^0}\right) + \mu$

DISPL: $\Delta E = \Delta U_{ji}(\underset{\approx}{r}^N)$

where $\Delta U_{ji}(\underset{\approx}{r}^N) = \underset{\text{NEW}}{U_j(\underset{\approx}{r}^N)} - \underset{\text{OLD}}{U_i(\underset{\approx}{r}^N)}$

N = number of particles (varying)

μ = fixed chemical potential

T = fixed temperature V = fixed volume

ρ^0 = "standard" density of 1 particle per Å^3

$U_i(\underset{\approx}{r}^N)$ = potential energy of ith configuration with N particles

GRAND CANONICAL MONTE CARLO

Constant μ, V, T

Norman-Filinov

equal probability for 3 moves:

2[create/ annihilate] + displace

$$P_{acc} = \begin{cases} \text{min. } (1, \exp(-\Delta E/k_B T)) & \dfrac{\Delta E}{k_B T} \leq 180 \\ 0 & > 180 \end{cases}$$

create $\quad \Delta E = \Delta U_{ji}(r^N) + kT \ln \left(\dfrac{N+1}{V\rho^0}\right) - \mu$

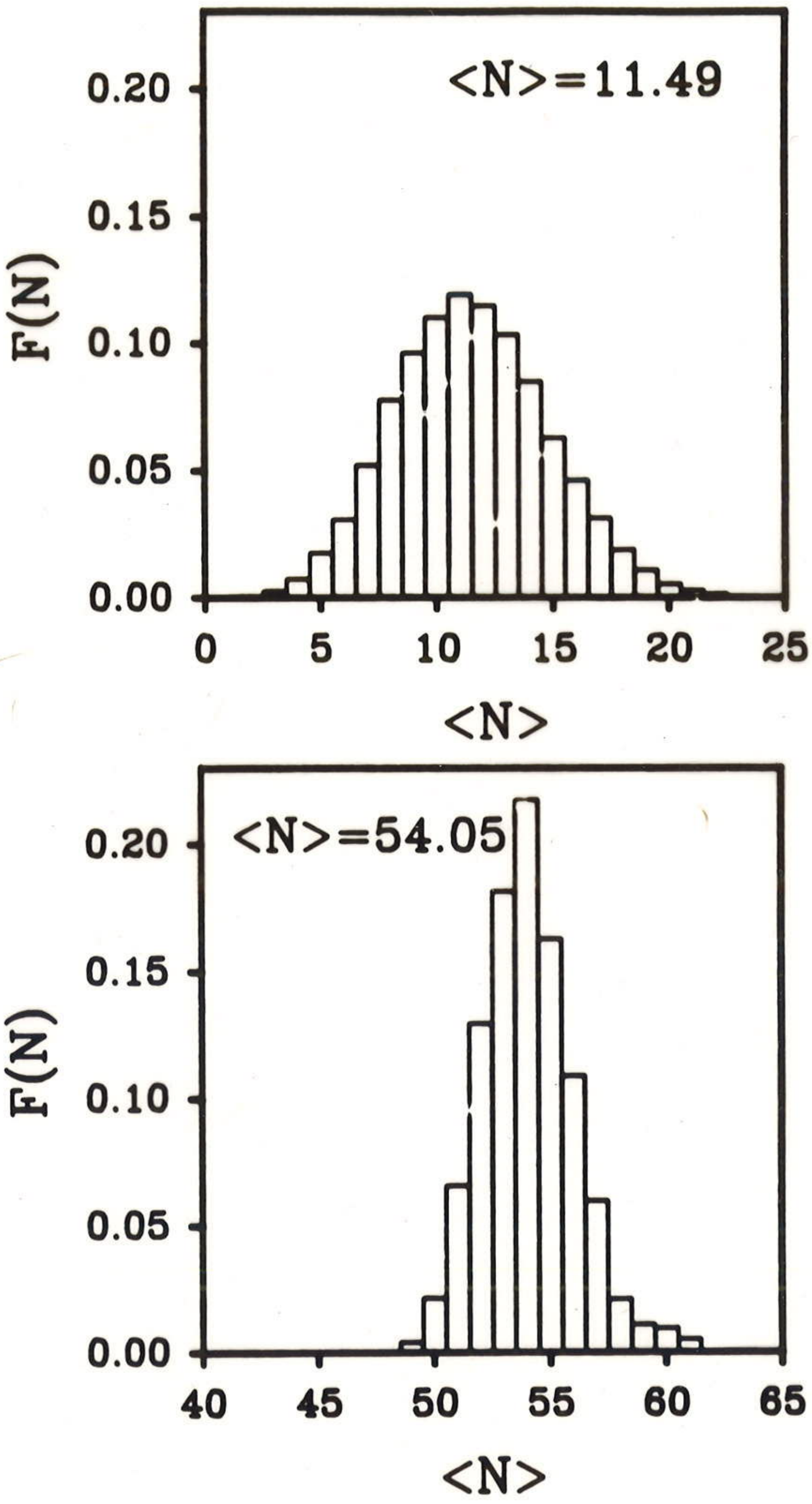
annih. $\quad \Delta E = \Delta U_{ji}(r^N) - kT \ln \left(\dfrac{N}{V\rho^0}\right) + \mu$

displ. $\quad \Delta E = \Delta U_{ji}(r^N)$

$$\Delta U_{ji}(r^N) = \underset{\text{NEW}}{U_j(r^N)} - \underbrace{\underset{\text{OLD}}{U_i(r^N)}}_{\text{ASSUME PAIRWISE ADDITIVE!}}$$

$$\langle \mu \rangle_{GCMC} = \mu_{\text{bulk gas}}$$

$$\mu_{\text{bulk gas}} = RT \ln(\rho/\rho 0) + RT(Z-1) + \int_0^\rho d\rho \, \frac{P-RT\rho}{\rho^2}$$


<N>=11.49
F(N)
0.20
0.15
0.10
0.05
0.00
0
5
10
15
20
25
<N>
<N>=54.05
F(N)
0.20
0.15
0.10
0.05
0.00
40
45
50
55
60
65
<N>

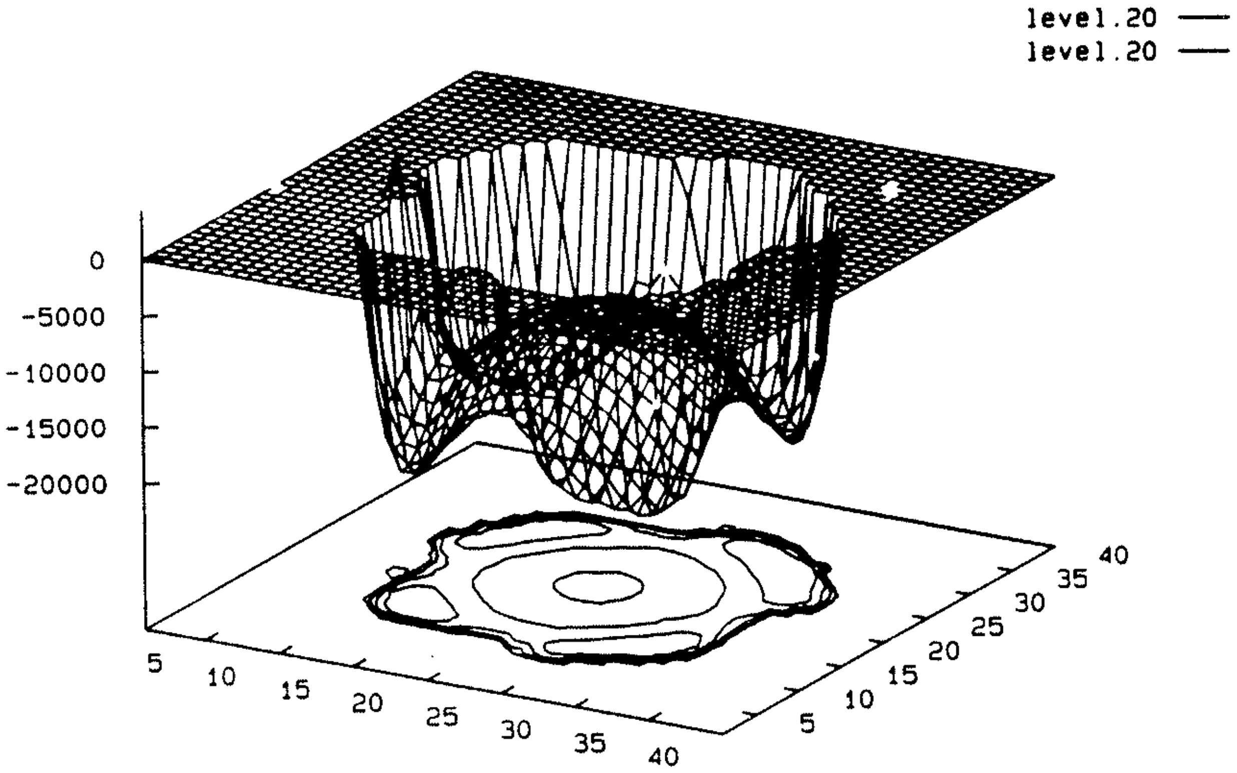
$$\langle \mu \rangle_{GCMC} = \mu = \mu_{bulk\,gas}$$

$$\mu_{bulk\,gas} = RT \ln\left(\frac{\rho}{\rho^0}\right) + RT(Z-1) + \int_0^{\rho} d\rho \frac{(P - RT\rho)}{\rho^2}$$

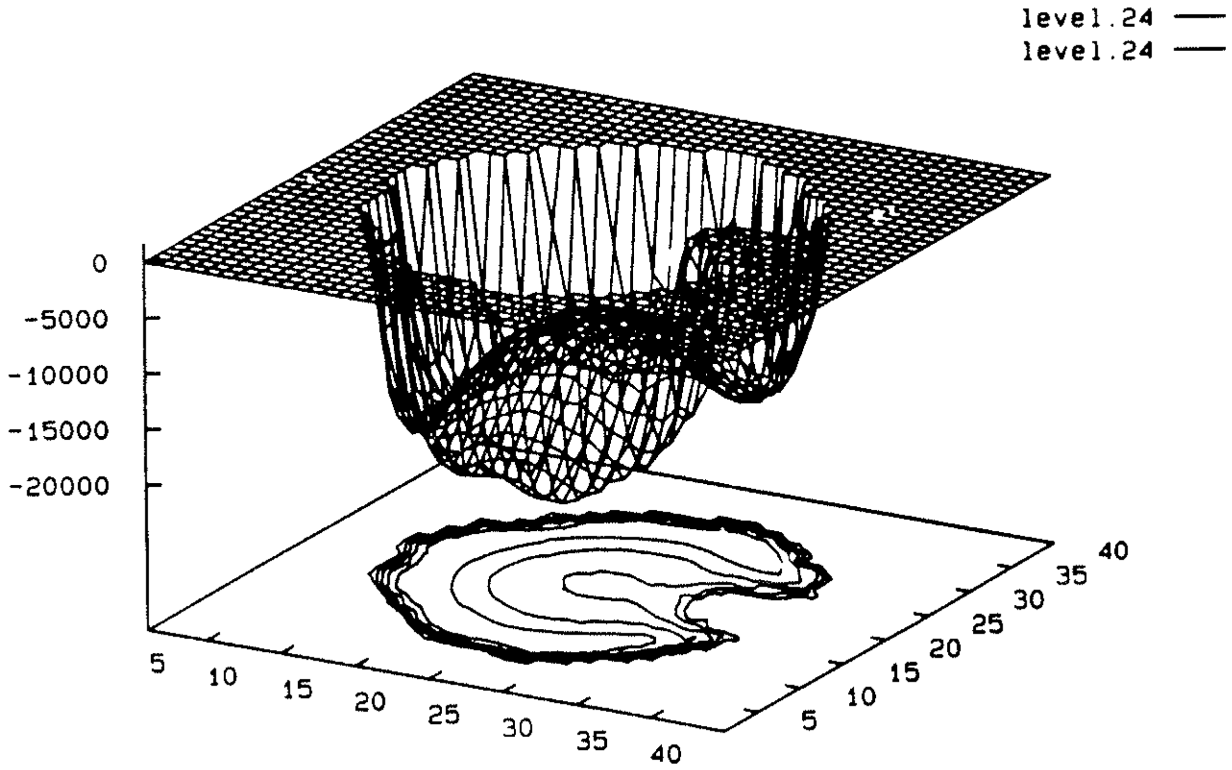
where $Z =$ compressibility factor $\frac{PV}{RT}$

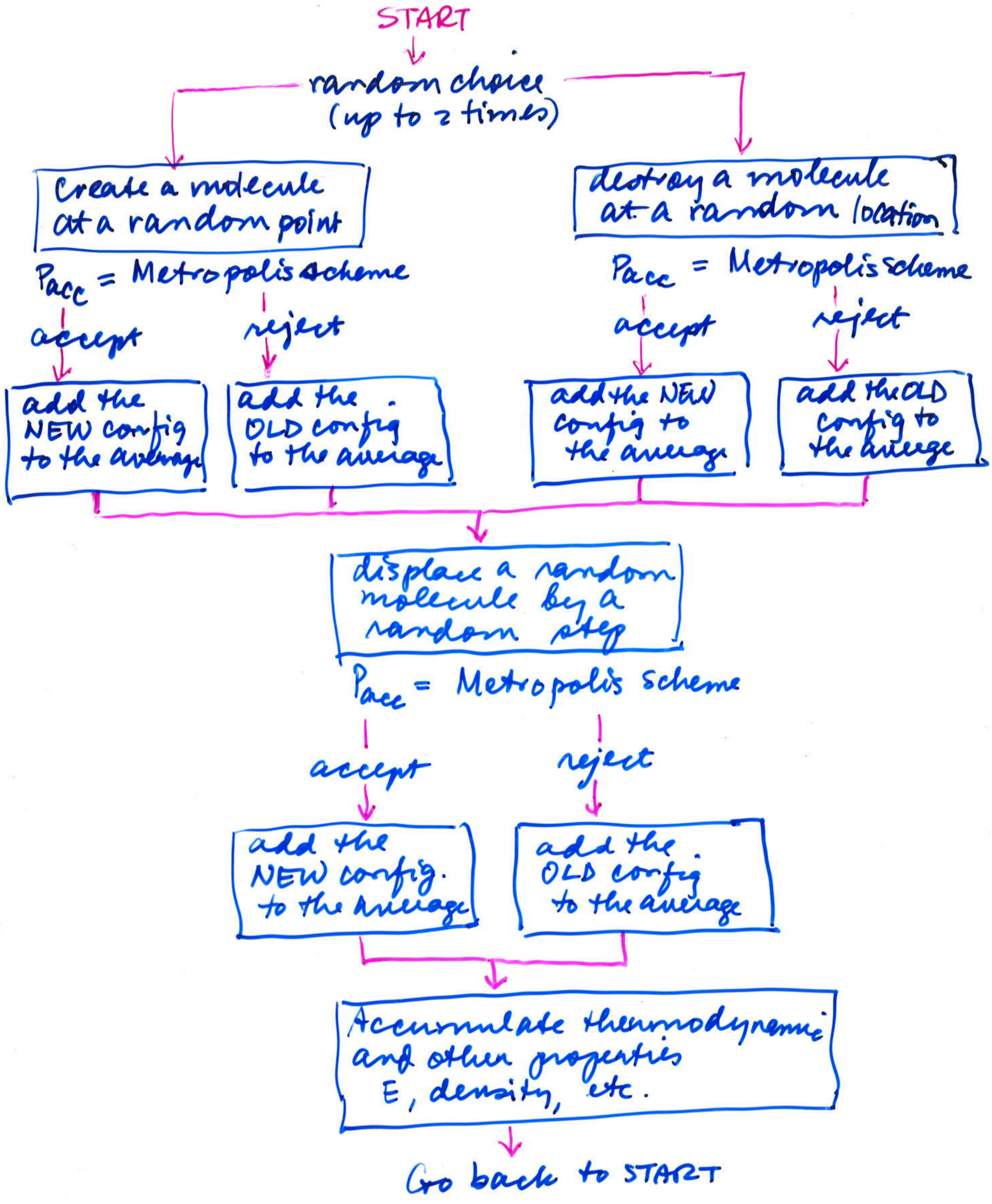
$\rho^0 =$ same "standard" density as used in GCMC

Potential Energy Surface



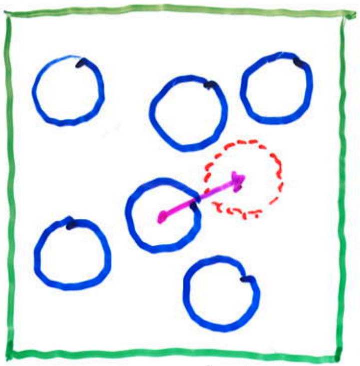
Potential Energy Surface




START
random choice
(up to 2 times)
Create a molecule at a random point
destroy a molecule at a random location
P_{acc} = Metropolis scheme
P_{acc} = Metropolis scheme
accept
reject
accept
reject
add the NEW config to the average
add the OLD config to the average
add the NEW config to the average
add the OLD config to the average
displace a random molecule by a random step
P_{acc} = Metropolis scheme
accept
reject
add the NEW config. to the average
add the OLD config to the average
Accumulate thermodynamic and other properties E, density, etc.
Go back to START

A Grand Canonical Monte Carlo cycle.

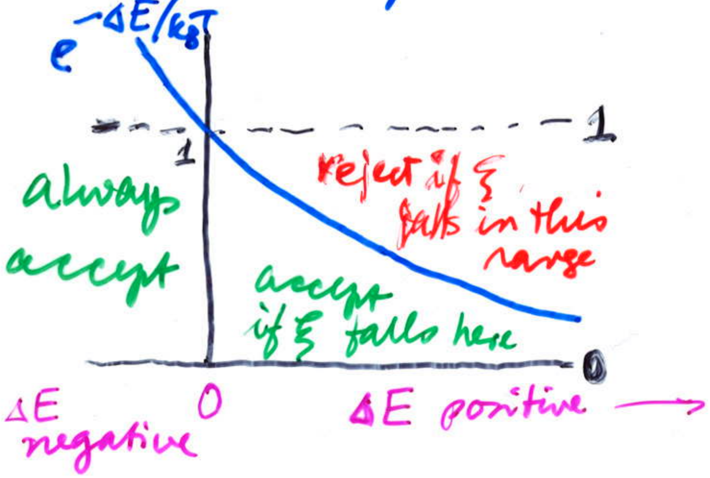
Do about $10^6 - 10^7$ cycles



Configuration j is generated from configuration i by displacing one atom

Generate a random number ξ between 0 and 1.0

Compare ξ with P_{acc}. If ξ is less than P_{acc} the move is accepted.



$$\Delta E = \underset{\text{NEW}}{U_j} - \underset{\text{OLD}}{U_i}$$

if $\Delta E/k_BT$ is too large (say greater than 75 or 180) then immediately reject the move (no need to evaluate $\exp^{-\Delta E/k_BT}$, also AVOIDs underflow in calculation of the exponential).

Finite size problem:

How does one extract estimates for observables of an infinite system from Monte Carlo averages on finite systems?

PERIODIC BOUNDARY CONDITIONS

SIMULATION BOX

The cubic box is replicated throughout space to form an infinite lattice of boxes. In the course of the simulation, as a molecule moves in the original box, its periodic image in each of the neighboring boxes moves in exactly the same way. Thus, As a molecule leaves the original box, one of its images will enter through the opposite face. As particle 1 moves through a boundary its images 1_A, 1_B, ... move across their corresponding boundaries. The number density in the original box (and hence in the entire system) is conserved. It is not necessary to store the coordinates of all the images, only the coordinates of the molecules in the original box.

Periodic boundary conditions have little effect on the EQUILIBRUM thermodynamic properties and structures of fluids away from phase transitions and where the interactions are short-ranged.

1.5.4 Computer code for periodic boundaries

How do we handle periodic boundaries and the minimum image convention, in a simulation program? Let us assume that, initially, the N molecules in the simulation lie within a cubic box of side L, with the origin at its centre, i.e. all coordinates lie in the range $(-\frac{1}{2}L, \frac{1}{2}L)$. As the simulation proceeds, these molecules move about the infinite periodic system. When a molecule leaves the box by crossing one of the boundaries, it is usual to switch attention to the image molecule entering the box, by simply adding L to, or subtracting L from, the appropriate coordinate. One simple way to do this uses a FORTRAN IF statement to test the positions immediately after the molecules have been moved (whether by MC or MD):

```
IF ( RX(I) .GT.  BOXL2 ) RX(I) = RX(I) - BOXL
IF ( RX(I) .LT. -BOXL2 ) RX(I) = RX(I) + BOXL
```

Here, BOXL is a variable containing the box length L, and BOXL2 is just half the box length. Similar statements are applied to the y and z coordinates. An alternative to the IF statement is to use FORTRAN arithmetic functions to calculate the correct number of box lengths to be added or subtracted:

```
RX(I) = RX(I) - BOXL * ANINT ( RX(I) / BOXL )
```

The function ANINT(X) returns the nearest integer to X, converting the result back to type REAL; thus ANINT (-0.49) has the value 0.0, whereas ANINT (-0.51) is -1.0. By using these methods, we always have available the coordinates of the N molecules that currently lie in the 'central' box. It is not strictly necessary to do this; we could, instead, use uncorrected coordinates, and follow the motion of the N molecules that were in the central box at the start of the simulation. If it is decided to do this, however, care must be taken that the minimum image convention is correctly applied, so as to work out the vector between the two closest images of a pair of molecules, no matter how many 'boxes' apart they may be.

The minimum image convention may be coded in the same way as the periodic boundary adjustments. Of the two methods mentioned above, the arithmetic formula is usually preferable, being simpler; the use of IF statements inside the inner loop, particularly on pipeline machines, is to be avoided . Immediately after calculating a pair separation vector, the following statements should be applied:

```
RXIJ = RXIJ - BOXL * ANINT ( RXIJ / BOXL )
RYIJ = RYIJ - BOXL * ANINT ( RYIJ / BOXL )
RZIJ = RZIJ - BOXL * ANINT ( RZIJ / BOXL )
```

The above code is guaranteed to yield the minimum image vector, no matter how many 'box lengths' apart the original images may be.

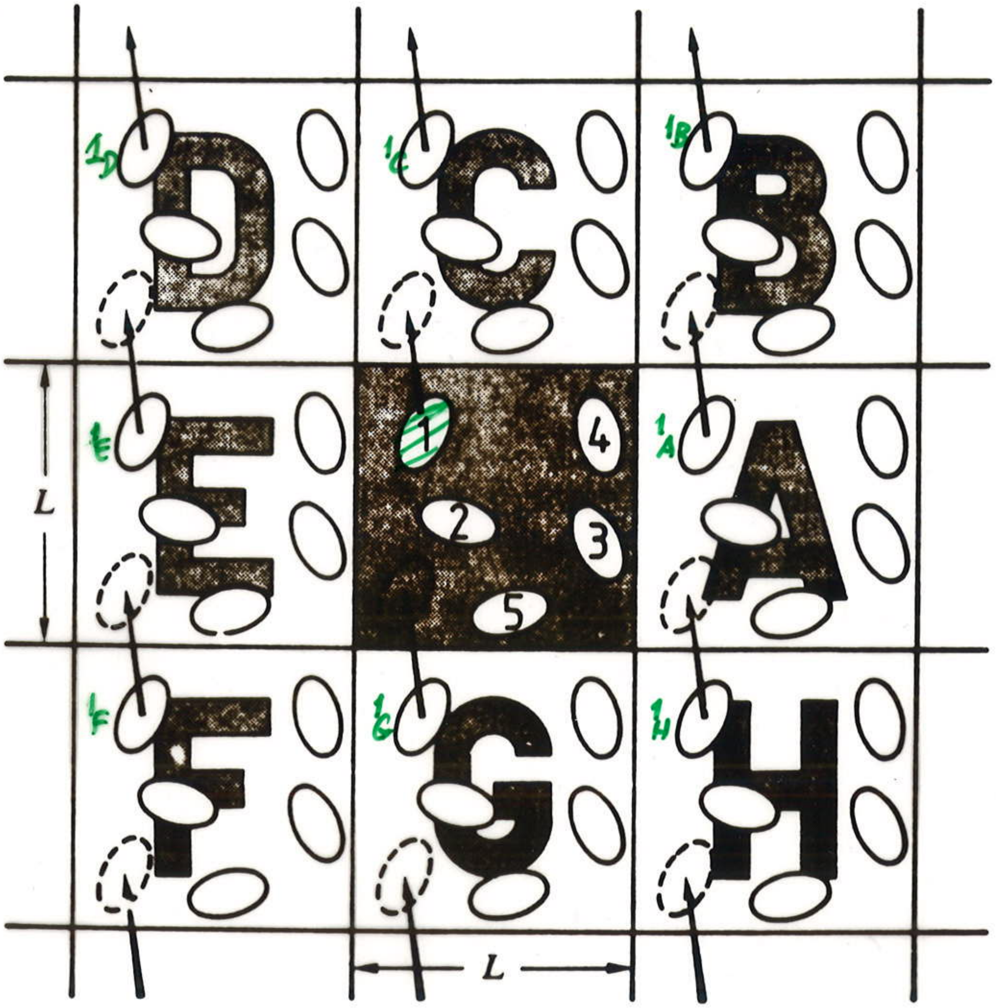
The calculation of minimum image distances is simplified by the use of reduced units: the length of the box is taken to define the fundamental unit of length in the simulation. Some workers define $L = 1$, others prefer to take $L = 2$. By setting $L = 1$, with particle coordinates nominally in the range $(-\frac{1}{2}, +\frac{1}{2})$, the minimum image correction above becomes

```
RXIJ = RXIJ - ANINT ( RXIJ )
RYIJ = RYIJ - ANINT ( RYIJ )
RZIJ = RZIJ - ANINT ( RZIJ )
```

which is simpler, and faster, than the code for a general box length.

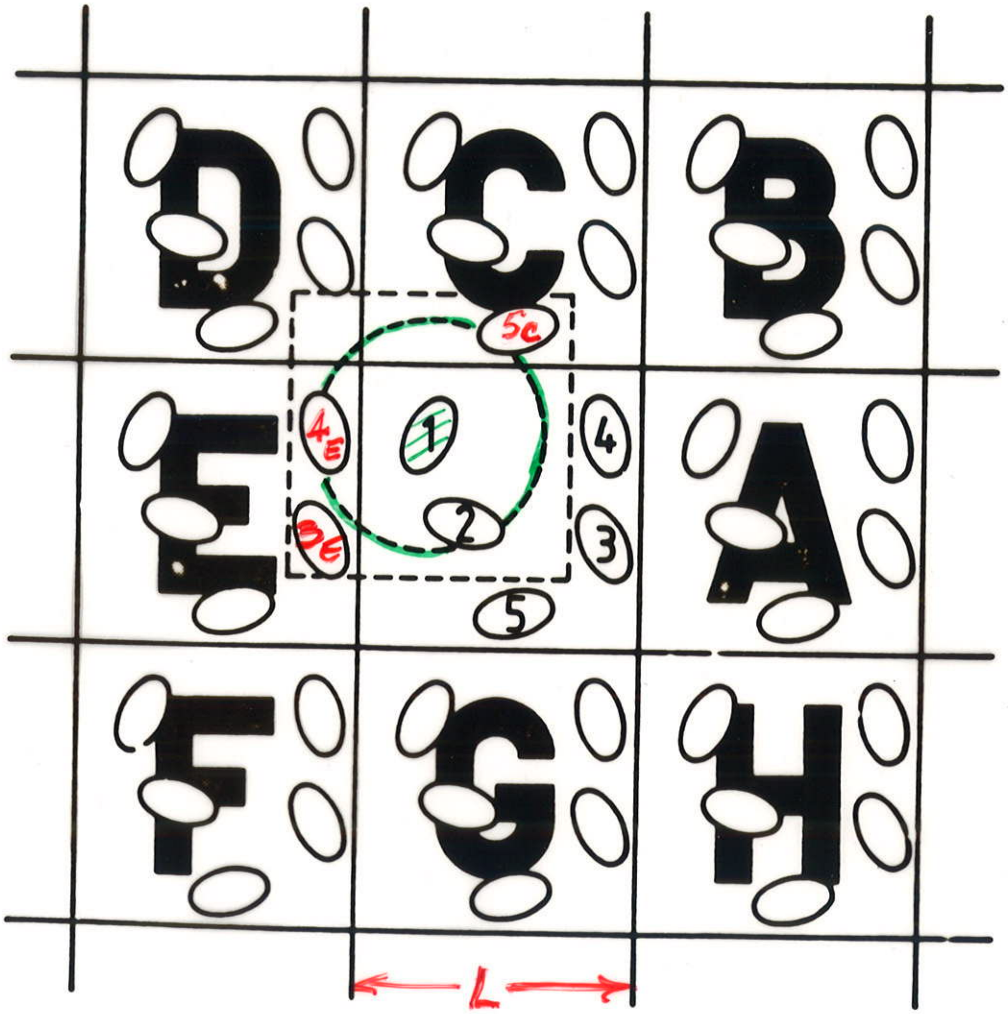
Now we turn to the implementation of a spherical cutoff, i.e. we wish to set the pair potential (and all forces) to zero if the pair separation lies outside some distance r_c. It is easy to compute the square of the particle separation r_{ij} and, rather than waste time taking the square root of this quantity, it is fastest to compare this with the square of r_c, which might be computed earlier and stored in a FORTRAN variable RCUTSQ. After computing the minimum image intermolecular vector, the following statements would be employed:

```
RIJSQ = RXIJ ** 2 + RYIJ ** 2 + RZIJ ** 2

IF ( RIJSQ .LT. RCUTSQ ) THEN

   ... compute i-j interaction ...
   ... accumulate energy and forces ...

ENDIF
```



PERIODIC BOUNDARY CONDITIONS

for a 2-dimensional system



"MINIMUM IMAGE CONVENTION"

Choose cutoff distance $\frac{1}{2}L$

Includes interactions between
1 and $\left\{ \begin{matrix} 2 \quad 4_E \text{ and } 5_C \\ 3_E \end{matrix} \right\}$ which are within $\frac{1}{2}L$ of particle 1

Simulation of a fluid (a Lennard-Jones fluid)

Problem: Consider N particles in a box of volume V. Choose the volume such that the simulation can be carried out for a particular density ρ. The particles interact with each other via a Lennard-Jones pair potential.

$$U(r_{ij}) = 4\epsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right]$$

parameters are σ and ϵ, r_{ij} is the distance between particle i and j.

Choose a particle. Generate a random number and displace the particle a distance δ derived from the random number. Calculate the change in the energy. Write an algorithm and program it. Compute the average internal energy, specific heat and pressure.

Spherical cut-off for pair potentials

cut-and-shifted potential (to have no discontinuity in the potential)

$$U^{S}(r_{ij}) = \begin{cases} U(r_{ij}) - U(r_c) & \ldots r_{ij} \le r_c \\ 0 & \ldots r_{ij} > r_c \end{cases}$$

$U(r_c)$ = value at the cut-off distance

shifted force potential (to have no discontinuity in the forces):

$$U^{SF}(r_{ij}) = \begin{cases} U(r_{ij}) - U(r_c) - \left(\frac{dU(r_{ij})}{dr_{ij}}\right)_{r_{ij}=r_c} \cdot (r_{ij} - r_c) & \ldots r_{ij} \le r_c \\ 0 & \ldots r_{ij} > r_c \end{cases}$$