METROPOLIS ALGORITHM FOR THE GRAND CANONICAL ENSEMBLE

\[ \Xi(\mu, V, T) = \sum_{N=0}^{\infty} \frac{e^{\beta \mu N}}{N!} \int d^3q e^{-\beta U(q_1, \ldots, q_N)} \]

We should update both $N$ and $\{q_1, \ldots, q_N\}$

(a) At given iteration the system consists in $N$ particles located in $\{q_1, \ldots, q_N\}$

We update their positions using the usual canonical Metropolis update.

(b) We must change the number $N$, i.e. $N \rightarrow N \pm 1$

We use the Metropolis algorithm.

As usual we must discuss the proposal for the insertion move ($N \rightarrow N+1$) and the deletion move ($N \rightarrow N-1$). [We propose them with the same prob]

We have $N$ particles in $\{q_1, \ldots, q_N\}$

2) Insertion

We choose a random point in the box: (cubic $L^3$).

$\vec{r} = (L*\text{RAN}(\cdot), L*\text{RAN}(\cdot), L*\text{RAN}(\cdot))$

We choose a label in $\{1, \ldots, N+1\}$ (label = $i$)

The new configuration is

$\vec{r}_1, \ldots, \vec{r}_{i-1}, \vec{r}_{i+1}, \ldots, \vec{r}_N, \vec{r}_i$

$[\vec{r}_i; \text{has been moved at the end, and } \vec{r}_i \text{ replaces } \vec{r}_i]$
Note that if $i = N+1$ the new configuration is simply
\[ \tilde{r}_1, \ldots, \tilde{r}_N, \tilde{r} \] [$\tilde{r}$ is appended at the end]

**DELETION**

We choose a random label $i$ in $\{0, \ldots, N\}$ and delete particle at $\tilde{r}_i$. The new conf. is
\[ \tilde{r}_1, \ldots, \tilde{r}_{i-1}, \tilde{r}_i, \tilde{r}_{i+1}, \ldots, \tilde{r}_N \]

Note that we have moved the lost particle to the "hole".

The insertion/deletion move have been chosen so that it is possible to go back and forth until a non vanishing probability.

**INSERTION**

\begin{align*}
\text{N part:} & \quad r_1, \ldots, r_N \quad \text{choose } \circ \quad \downarrow \text{insert} \\
\text{N+1 part:} & \quad r_1, \ldots, r_{i-1}, r_i, r_{i+1}, \ldots, r_N, r_i \quad \downarrow \text{delete} \quad \text{choose the same } i \\
\text{N-part:} & \quad r_1, \ldots, r_{i-1}, r_i, r_{i+1}, \ldots, r_N
\end{align*}

SAME CONF AS THE ONE WE STARTED FROM
We wish now to compute $P^{(0)}$.

$$P^{(0)} \left[ N \rightarrow \bar{N}, \{ r_1, \ldots, r_N \} \rightarrow N+1, \{ r_1, \ldots, r_N, r_{N+1} \} \right]$$

Up arrow: we choose a point in the box.

We choose a label among $(N+1)$ possibilities.

Down arrow: (P^{(0)} is a prob. DENSITY)

$$P^{(0)} \left[ N+1, \{ r_1, \ldots, r_N, r_{N+1} \} \rightarrow N, \{ r_1, \ldots, r_N \} \right]$$

$$= \frac{1}{N+1} \cdot \frac{1}{2} \left( \text{we choose a particle among} \right) \cdot \frac{1}{2}$$

$$R(N \rightarrow N+1) = \frac{P^{(0)}(N \rightarrow N+1) \pi(N+1)}{P^{(0)}(N, N+1) \pi(N)}$$

$$= \frac{1}{2^{(N+1)}} \cdot \frac{\lambda^{N} N!}{\lambda^{2N} (N+1)!} \cdot \frac{1}{\lambda^2} \cdot e^{-\beta U_{N+1}}$$

$$= \frac{e^{\beta \mu} \lambda^{N+1}}{\lambda^{3(N+1)}} \cdot e^{-\beta U_{N+1}}$$

$$= \frac{e^{\beta \mu} \lambda^{N+1}}{\lambda^{3(N+1)}} \cdot e^{-\beta (U_{N+1} - U_N)}$$

$$A(N, N+1) = \min \left( 1, \frac{e^{\beta \mu} \lambda^{N+1}}{\lambda^{3(N+1)}} \cdot e^{-\beta (U_{N+1} - U_N)} \right)$$
\[ \Theta(N+1, N) = \min \left( 1, \frac{1}{R(N+1, N)} \right) \]
\[ = \min \left( 1, \frac{1}{R(N,N+1)} \right) \]
\[ = \min \left( 1, \frac{x^3(N+1)}{V e^{\beta N}} e^\beta (U_{N+1} - U_N) \right) \]
\[ \Theta(N, N-1) = \min \left( 1, \frac{x^3 N}{V e^{\beta N}} e^\beta (U_N - U_{N-1}) \right) \]

**In practice:** If we have \( N \) particles in \( \{n_1, \ldots, n_N\} \)

1. Choose which move to perform.
   
   \( X = \text{RAND()} \)
   
   - If \( X < 0.5 \) delete one particle.
   - If \( X \geq 0.5 \) add one particle.

2. If the move is to delete a particle, choose one particle to be deleted.
   
   Compute \( U_N - U_{N-1} \) and accept the move with probability
   
   \[ \min \left( 1, \frac{x^3 N}{V e^{\beta N}} e^\beta (U_N - U_{N-1}) \right) \]

3. Alternatively, add one particle.
   
   Compute \( U_{N+1} - U_N \) and accept the move with probability
   
   \[ \min \left( 1, \frac{V e^{\beta N}}{x^3(N+1)} e^{-\beta (U_{N+1} - U_N)} \right) \]
COMMENT: The reorganization of the labels is only needed in the proof, but not in the practical implementation.

The reorganization is usually defined in the following way.

We choose a point $\tilde{\mathbf{r}} = (LRAN(\mathcal{X}), LRAN(\mathcal{Y}), LRAN(\mathcal{Z}))$
and the new configuration is

$\tilde{r}_1, \ldots, \tilde{r}_N, \tilde{r}$ (end)

Delhino is unchanged. If we use the same acceptances as before, the new algorithm is still correct: it does not satisfy detailed balance, but it satisfies stationarity.