

# 1 Metropolis sampling

In this section, a brief walkthrough of the Metropolis-Hastings algorithm [2] will be given. First, the process of obtaining acceptance ratios that satisfy the Principle of Detailed Balance will be given. Then, the Metropolis-Hastings algorithm will be stated.

## 1.1 Deriving Metropolis acceptance ratios

The Principle of Detailed Balance states that in an equilibrium system, the number of transitions from a configuration  $x$  to a configuration  $x'$ , must be equal to the number of transitions from  $x'$  to  $x$ . This principle can be expressed as:

$$p(x)P(x \rightarrow x') = p(x')P(x' \rightarrow x) \quad (1)$$

where  $p(x)$  is the steady state probability distribution, and  $P(x \rightarrow x')$  is the probability of transitioning to  $x'$  conditional to original state  $x$ . Solving for the ratio of transition probabilities:

$$\frac{P(x \rightarrow x')}{P(x' \rightarrow x)} = \frac{p(x')}{p(x)} \quad (2)$$

The transition probability can be rewritten as a factor of proposal and acceptance probabilities:

$$P(x \rightarrow x') = g(x \rightarrow x')A(x \rightarrow x') \quad (3)$$

where  $g(x \rightarrow x')$  is the probability of proposing an update that takes the state of the system from  $x$  to  $x'$ , and  $A(x \rightarrow x')$  is the probability of accepting such update. Substituting the factored transition probabilities into Eq. (2):

$$\frac{g(x \rightarrow x')A(x \rightarrow x')}{g(x' \rightarrow x)A(x' \rightarrow x)} = \frac{p(x')}{p(x)} \quad (4)$$

The ratio of acceptance probabilities becomes:

$$\frac{A(x \rightarrow x')}{A(x' \rightarrow x)} = \frac{p(x')g(x' \rightarrow x)}{p(x)g(x \rightarrow x')} \equiv R \quad (5)$$

## 1.2 The Metropolis-Hastings algorithm

$$\pi \approx 3.14 \dots \quad (6)$$

Essentially, Eq. (6) is an approximation. In the incredible paper by Prokockokokok [1], the authors play some Pokemon. In the Metropolis paper [2]. On the Brian Keng's website [3]

## 2 Lattice Worm Algorithm (WA) Updates

In this section, a set of ergodic lattice worm updates is introduced. First, an explanation of how each update changes the system is given.. Then, a walkthrough of the decisions that comprise each update will be given, including the probabilities for each desicion's outcome. Finally, the Metropolis conditions are derived or the direct sampling explained, depending on the update.

### 2.1 Insert/Delete worm

The insert worm update creates a particle on a flat region of a worldline, then destroys it after a certain time, also inside the same flat region. Formally, the particle is created by acting on the state at that imaginary time with the bosonic creation operator (worm tail) and anihilated by acting with the bosonic anihilation operator (worm head). An antiworm can instead be inserted by first inserting the worm head and then the tail. In other words, an antiworm will first anihilate a particle and create one at a later time inside the flat region. The update proceeds as follows:

0. Do with probability  $p_{iw}$  **What is this actually? There's already an attempt**

and acceptance prob.

1. Randomly choose an integer  $i \in [0, L - 1]$ , where  $L$  is the number of sites on the lattice. The  $i^{th}$  site will be chosen with probability  $p_i = 1/L$ .
2. Randomly choose an integer  $f \in [0, F - 1]$ , where  $F$  is the number of flat regions on site  $i$ . The  $f^{th}$  flat region will be chosen with probability  $p_f = 1/F$ .
3. Count the number of particles  $n_{flat}$  on the flat region and check if inserting an antiworm is possible:
  - (a) If  $n_{flat} = 0$  : Only a worm can be inserted with probability  $p_{type} = 1$
  - (b) Else: A worm or antiworm can be inserted with probability  $p_{type} = 1/2$
4. Randomly choose a real number  $\Delta\tau_{worm} = rand() * \Delta\tau_{flat}$ , where  $\Delta\tau_{flat}$  is the length of the flat region and  $rand()$  is a random number from the uniform distribution in the interval  $[0, 1)$ . The probability of the worm being of length  $\Delta\tau_{worm}$  is  $p_{len} = 1/\Delta\tau_{flat}$ .
5. Randomly select a real number  $\tau = \tau_{min} + rand() * (\Delta\tau_{flat} - \Delta\tau_{worm})$ , where  $\tau_{min}$  is the lower bound of the flat region. The probability of inserting the worm (antiworm) tail (head) at  $\tau$  is  $p_{\tau} = 1/(\Delta\tau_{flat} - \Delta\tau_{worm})$ .
6. Calculate the ratio of weights of configurations pre and post worm insertion:

$$\frac{W_+}{W_-} = \eta^2 e^{-\Delta\tau_{worm} \Delta V} \quad (7)$$

where  $W_+$  and  $W_-$  are the weights of the configuration with a worm and no worm, respectively,  $\eta$  is the worm fugacity and  $\Delta V$  is the change in potential energy pre and post worm insertion.

- Derive  $\Delta V$  here for insert worm and antiworm
- Give a similar walkthrough of the delete worm update
- Derive the Metropolis condition from Detailed Balance

$$\pi \approx 3.14159... \tag{8}$$

## 2.2 Insert/Delete ground state (gs) worm

These updates will be very similar to the insert/delete worm update. The main difference is that these are particular to the case of zero temperature. Remember that at zero temperature, imaginary time is not subject to periodic boundary conditions, as is the case at finite temperature. Instead, open boundary conditions are imposed in the imaginary time direction, while keeping the space direction periodic.

The open boundary conditions will allow now for the insertion of worms that will have one of its ends go past either the  $\tau = 0$  or the  $\tau = \beta$  boundaries. These worms that go past the imaginary time boundaries will be called ground state (gs) worms. In practice, inserting a gs-worm will look like inserting only one worm end to the worldline configuration at a time, and analogously for deletion.

### 2.2.1 Insert gs-worm

At  $T = 0$ , a ground state (gs) worm or antiworm can be inserted. One of the ends of this worm will lie either past the  $\tau = 0$  or the  $\tau = \beta$  boundaries. The other end, will lie in the flat region preceding this boundary (i.e, the first or the last flat region). In practice, this will look like inserting a single worm end to the worldline configuration at these flat regions. This worm end can only be inserted if there are no worm ends or one worm end present. The update proceeds as follows:

0. Do with probability  $p_{gsiw}$
1. Choose site with probability  $p = 1/L$
2. Choose first or last flat:
  - (a) if  $n_{flats} = 1$ : treat the flat as first or last (depending on the implementation) with  $p = 1$
  - (b) else:  $p = 1/2$
3. Choose gs-worm or gs-antiworm: if  $n_i = 0$ : can only insert worm with prob.  $p = 1$  else: choose either with  $p = 1/2$
4. Choose the insertion time of the worm end  $\tau = \tau_{min} + rand()( \tau_{max} - \tau_{min} )$  with probability  $p = 1/(\tau_{max} - \tau_{min})$

### 2.2.2 Delete gs-worm

The Delete gs-worm update can only be done if there's at least one worm end on the worldline configuration. A random worm end is chosen and then deleted. The update proceeds as follows:

0. Do with probability  $p_{gsdw}$
1. Choose the worm end to be deleted with probability  $p = 1/2$

### 2.2.3 Acceptance probabilities of gs-worm insert/delete

From detailed balance:

$$w_- p_{gsiw}^{att} p_{gsiw}^{acc} = W_+ p_{gsdw}^{att} p_{gsdw}^{acc} \quad (9)$$

Solving for the ratio of acceptance probabilities:

$$\frac{p_{gsiw}^{acc}}{p_{gsdw}^{acc}} = \frac{p_{gsdw}^{att} W_+}{p_{gsiw}^{att} W_-} \quad (10)$$

Inserting the attempt probabilities:

$$\frac{p_{gsiw}^{att}}{p_{gsdw}^{att}} = \frac{p_{gsdw}^{att}}{p_{gsiw}^{att}} \frac{L(\tau_{max} - \tau_{min})}{p_{flat}} \frac{W_+}{W_-} \quad (11)$$

where  $p_{flat}$  is the probability of choosing either the first or last flat region of the worldline. If there are no kinks in the worldline,  $p_{flat} = 1$ , and if there are kinks,  $p_{flat}=1/2$ .

The ratio of weights for configurations after a gs-worm insertion and after it is deleted is:

$$\frac{W_+}{W_-} = \eta e^{-\Delta\tau\Delta V} \quad (12)$$

## 2.3 Timeshift

### 2.3.1 Forward

The timeshift update consists of moving either a worm head or tail either forward or backwards in imaginary time. The update proceeds as follow:

0. Do with probability  $p_{fw}$  :
1. Choose which worm end will move, head or tail, with probability  $p = 1/2$
2. Randomly choose a real number  $\tau_{new} = \tau + rand() * (\tau_{max} - \tau)$ , where  $\tau_{max}$  is the upper bound of the flat region delimited by the moving end and the next kink or worm end and  $\tau$  is the original time of the moving worm end. The probability that the worm end will move to this time is  $p = 1/(\tau_{max} - \tau)$
3. Calculate the ratio of weights of configurations pre and post timeshifting forward:

$$\frac{W_+}{W_-} = e^{-\Delta\tau_{worm}\Delta V} = 1 \quad (13)$$

where  $W_+$  and  $W_-$  are the weights of the configuration post and pre moving forward, respectively, and  $\Delta V$  is the change in potential energy.

Since the number of particles on each site remains constant before and after the update, the change in potential energy is  $dV = 0$  and thus the exponential becomes unity.

### 2.3.2 Backward

0. Do with probability  $p_{bw}$  :
1. Choose which worm end will move, head or tail, with probability  $p = 1/2$
2. Randomly choose a real number  $\tau_{new} = \tau_{min} + rand() * (\tau - \tau_{min})$ , where  $\tau_{min}$  is the lower bound of the flat region delimited by the moving end and the preceding kink or worm end and  $\tau$  is the original time of the moving worm end. The probability that the worm end will move to this time is  $p = 1/(\tau - \tau_{min})$
3. Calculate the ratio of weights of configurations pre and post timeshifting backward:

$$\frac{W_-}{W_+} = e^{\Delta\tau_{worm}\Delta V} = 1 \quad (14)$$

### 2.3.3 Detailed balance

$$p_{fw}^{att} p_{fw}^{acc} W_- = p_{fw}^{att} p_{fw}^{acc} W_+ \quad (15)$$

The attempt probabilities for the forward and backward timeshift can be read off from the description of the update. Substituting them in and solving for  $p_{fw}^{acc}/p_{bw}^{acc}$ :

$$\frac{p_{fw}^{acc}}{p_{bw}^{acc}} = \frac{\tau_{max} - \tau}{\tau - \tau_{min}} = R \quad (16)$$

## 2.4 Spaceshift before

## 2.5 Spaceshift after

## References

- [1] N. V. Prokof'ev, B. V. Svistunov, and I. S. Tupitsyn. "Exact, complete, and universal continuous-time worldline monte carlo approach to the statistics of discrete quantum systems." *Journal of Experimental and Theoretical Physics* **87**, 310 (1998).  
<https://doi.org/10.1134%2F1.558661>
- [2] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. "Equation of state calculations by fast computing machines." *The Journal of Chemical Physics* **21**, 1087 (1953).  
<https://doi.org/10.1063/1.1699114>
- [3] B. Keng. "Markov chain monte carlo methods, rejection sampling and the metropolis-hastings algorithm."  
<http://bjlkeng.github.io/posts/markov-chain-monte-carlo-mcmc-and-the-metropolis-hastings-algorithm/>