

Monte Carlo Simulations of Dynamical processes

Kinetic Monte Carlo method: Consider the Monte Carlo (MC) simulation of a complex dynamical phenomenon that involves N independent processes. As an example, consider the problem of modeling the time-dependent evolution of a 2-dimensional Ising model with N spins, where spins are flipped in time due to the influence of thermal fluctuations.

For simplicity, consider that the time-evolution of any spin j is described by the first order differential equation $\frac{ds_j(t)}{dt} = -k_j^{(1)} s_j(t)$, with $s_j(t) = s_j(0) \exp(-k_j^{(1)} t)$. We assume that the rate constants $k_j^{(1)}$ can take different values $k_j^{(1)} = (k_j^{(1)}(0), k_j^{(1)}(1), k_j^{(1)}(2), k_j^{(1)}(3), k_j^{(1)}(4))$, depending on the number of nearest neighbors (1—4) of j that have the same spin as j .

Accurate MC simulations should sample event j with probability $p_j = \frac{k_j^{(1)}}{k}$, where $k = \sum_{j=1}^N k_j^{(1)}$ is the total rate. This can be accomplished according to the so-called 'kinetic MC' method as follows:

- (1) Set the simulation time to zero, $t = 0$.
- (2) Pick a random number r_1 between 0 and k .
- (3) Select the process labeled by index α that satisfies the condition,

$$\sum_{j=1}^{\alpha-1} k_j^{(1)} \leq r_1 < \sum_{j=1}^{\alpha} k_j^{(1)}. \quad (1)$$

- (4) Carry out event α .
- (5) Assign the time $t_\alpha = -\frac{1}{k} \ln r_2$ to the selected event, where r_2 is a random number between 0 and 1, and advance the simulation time $t = t + t_\alpha$.
- (6) Update the values of $k_j^{(1)}$ that have changed due to event α .

(7) Recompute k .

(8) Goto (2).

Selecting the event α , according to Eq. (1), can be pictured as throwing a dart to a ruler of length k , subdivided by segments of length $k_1^{(1)}, k_2^{(1)}, \dots, k_N^{(1)}$ as depicted in Fig. 1. The process α corresponds to the segment targeted by the dart ($\alpha = 7$, in Fig.1).

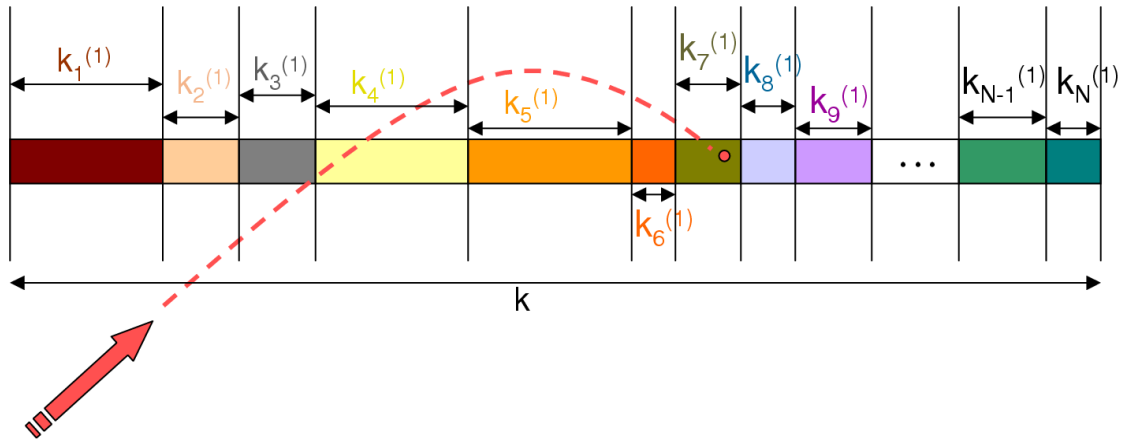


Figure 1: Dart and ruler scheme described in the text.

Unfortunately, finding out which segment was hit by the dart requires $O(N)$ operations when the search is based on a naïve linear search procedure, since it is necessary to evaluate Eq. (1) with $\alpha = 1$, then with $\alpha = 2, \dots$, etc., until finding the value of α that satisfies Eq. (1). Such a procedure is prohibitively expensive for most realistic applications. It is, therefore, necessary to implement an algorithm based on a binary tree where α can be selected in $O(\log_2 N)$ steps, as described later in this section [J.L. Blue; I. Beichl, *Phys. Rev. E* (1995) **51**, R867-R868].

Binary tree method: In order to explain the binary tree method, consider a system with only 8 possible events, where rate constants $k_1^{(1)}, \dots, k_8^{(1)}$ define the roots of the tree depicted in Fig. 2. The next layer of nodes in the tree, $k_1^{(2)}, \dots, k_4^{(2)}$, is defined by computing the partial sums of pairs of rate constants, as follows: $k_1^{(2)} = k_1^{(1)} + k_2^{(1)}, k_2^{(2)} = k_3^{(1)} + k_4^{(1)}, \dots, k_4^{(2)} = k_7^{(1)} + k_8^{(1)}$. The third layer

of nodes, $k_1^{(3)}, \dots, k_2^{(3)}$ is defined analogously, as follows: $k_1^{(3)} = k_1^{(2)} + k_2^{(2)}$, $k_2^{(3)} = k_3^{(2)} + k_4^{(2)}$. Finally, the top of the tree is $k_1^{(4)} = k_1^{(3)} + k_2^{(3)} = k$.

In order to select the dynamical process α , we start from the top of the tree, where $k_1^{(4)} = k_j^{(n)}$, with $j = 1$ and $n = 4$, and we proceed as follows:

(0) Generate a random number r_1 .

(1) If $r_1 \leq k_{2j-1}^{(n-1)} = k_1^{(3)}$, make $j = 2j - 1$. Otherwise, subtract $k_{2j-1}^{(n-1)}$ from r_1 and make $j = 2j$.

(2) If $n = 1$, then $\alpha = j$. Otherwise, make $n = n-1$ and go to (1).

Updating the binary tree, after carrying out the selected event, can also be done recursively from $n=1$ to the top of the tree, by propagating the effects of the new rate constants.

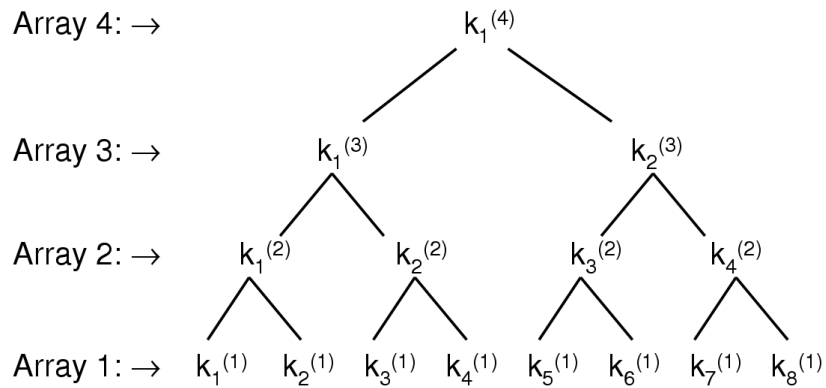


Figure 2: Binary tree explained in the text.