Monte Carlo Methods
Lecture 16: Rejection-Free Monte Carlo Methods

Dieter W. Heermann

Heidelberg University

January 2, 2020

Table of Contents

Rejection-Free Monte Carlo I

So far, we have been using the rejection Monte Carlo algorithms. To remind us, the algorithms proceeds from state x to possible state x' as outlined in Algorithm 1.

Algorithm 1 Accept/Reject Monte Carlo Algorithm

```
1: Choose initial state $x$
2: for n-of-samples do
3:     Select a new state $x'$
4:     With probability $p$ accept, i.e. set $x = x'$
5:     With probability $(1 - p)$, $x'$ is rejected
6: end for
```

- The probability will depend on some change induced by the state change
- Construct algorithm that does not involve accept/reject, i.e. always accept
- Thus, the methods will be (synchonously or asynchronously) event-driven [1] (see Algorithm 2)

Rejection-Free Monte Carlo II

Algorithm 2 Event-Driven Algorithm

```
1: for n-of-samples do
2:     Identify all possible events
3:     Identify the event with the smallest time stamp $\Delta t$
4:     Set time $t = t + \Delta t$
5: end for
```

- Methods that rely on rates between states thus the sequence that ultimately will be generated evolves in time (see Fig. 1).
- However, not as in the previous chapters systolically, driven by a constant increment in time, but by leaps of various length in time.
- This also opens up the possibility to make rigorous the notion of time in Monte Carlo methods.

Rejection-Free Monte Carlo III

.

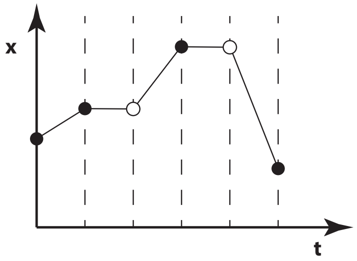

Figure: The lhs panel shows the typical systolic propagation of time for example in the Metropolis Monte Carlo. Sometimes new state proposals are rejected (circles) and the previous state is the new state. The rhs panel depicts the leaps in time that are made to achieve a rejection free algorithm.

Rejection-Free Monte Carlo IV

- Consider a state space Ω and a sequence $\{x_{t_k} \in \Omega\}$ of states from the state space. Often we simply write i or j etc. to label the states.
- Here we assume $t_0 < t_1 < \cdots < t_k < \cdots$.
- So far we have had $\Delta t = t_k - t_{k-1}$ constant, i.e. the system was moved forward in time by a constant stride.
- Furthermore, for two states (x_{k-1}, x_k) we have the Markov property so that the sequence $\{x_{t_k} \in \Omega\}$ is a Markov chain.
- Let us now look at continuous-time Markov chains $\{x_t \in \Omega | t \in \mathbb{R}, t \geq 0\}$. For the chain to be a continuous-time Markov chain the following condition needs to apply

Rejection-Free Monte Carlo V

$$\mathbb{P}(x(t+\tau)=j|x(\tau)=i, x(u)=k, 0\le u\le\tau)=\mathbb{P}(x(t+\tau)=j|x(\tau)=i)\;. \tag{1}$$

Define

$$p_{ij}(t) := \mathbb{P}(x(t+\tau)=j|x(\tau)=i)=\mathbb{P}(x(t)=j|x(0)=i) \tag{2}$$

and for any state i we have (for N possible states)

$$\sum_{j=1}^{N} p_{ij}(t) = 1\;. \tag{3}$$

Let $P(0)=\lim_{t\searrow 0} P(t) = I$ be the initial condition. Then the matrix R defined by

$$\lim_{h\searrow 0}\frac{P(h)-I}{h} = P'(0) = R \tag{4}$$

is the infinitesimal generator of the continuous-time Markov process with rate r_{ij}

$$\sum_{j=1, j\neq i} r_{ij} = -r_{ii} \tag{5}$$

and

Rejection-Free Monte Carlo VI

$$r_{ij} = \lim_{h \searrow 0} \frac{p_{ij}(h)}{h} \geq 0 \text{ and } r_{ii} \leq 0 \ . \tag{6}$$

Define $r_i := -r_{ii} > 0$ to be the rate corresponding to state i. Given R, then for all $t \geq 0$

$$P'(t) = RP(t) \ . \tag{7}$$

and

$$P(t) = Re^{-Rt} \tag{8}$$

as the first-passage-time distribution and further

$$p_{ij} = r_{ij} e^{-r_{ij} t} \ . \tag{9}$$

Since we are talking about first-passage only, only one of the possibilities can happen. Thus, rather than focusing on the transition probabilities (c.f. Fig. 2) as we have in the previous chapters, we can focus on the rates between states opening up to models where there is no Hamiltonian. Even more so, the rates themselves may depend on time. If they do not then the Markov process is stationary.

Rejection-Free Monte Carlo VII

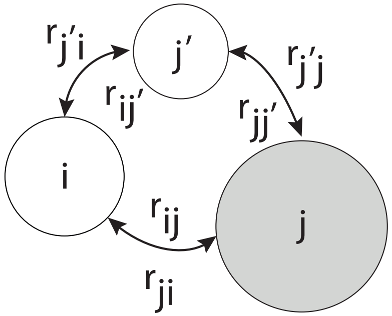

Figure: The figure shows the general situation where the circles denote states in state space that belong to the same state *i*.

Rejection-Free Monte Carlo VIII

- Let n_i denote the population of state i.
- Given that we are dealing with a thermal system then n_i must be proportional to $\exp\{-F(i)/k_B T\}$.
- In equilibrium if we have detailed balance then

$$n_i r_{ij} = n_j r_{ji} \ . \tag{10}$$

- Thus, what is needed for a model is a state space Ω and a set of rates R, i.e. (Ω, R).
- This can for example be a set of chemical reactions with the corresponding rates.
- We envisage that at any given time for a state i not all states j are accessible.
- Thus it is convenient to relabel the currently accessible states with a new label.
- We arrive at a list of N possible events and a list with corresponding rates

$$\{E_n \in \Omega\} \quad \text{with } n = 1, \ldots, N \tag{11}$$

$$\{r_n\} \quad \text{with } n = 1, \ldots, N \tag{12}$$

Rejection-Free Monte Carlo IX

- From a computational point of view, it is immediately apparent that what is needed is a well performing bookkeeping algorithm for the events and the rates as they may change after an event has been chosen
- Let us consider this for the Ising model.
- It was pointed out by Bortz, Kalos and Lebowitz [2] that the probability of accepting new configurations in the Ising model is very low in same cases.
- Consider the case when the temperature is low.
- Then two spins will have likely the same orientation and thus a reversal has very low probability.
- Thus, out of the N attempts only a very low fraction will result in changes. Suppose only attempts are made that are successful.
- For this the rates r_{ij} from state i to j need to be known a priori.

Rejection-Free Monte Carlo X

- In the Ising case we know transition rates among states a priori. For the two-dimensional Ising model

$$\mathcal{H} = -J \sum_{<i,j>} S_i S_j \quad S_i = \pm 1 \tag{13}$$

with its spin-up spin-down symmetry we have the situations as shown in Table 1.

Spin			$\uparrow (+1)$					$\downarrow (-1)$		
Neighbours	4	3	2	1	0	0	1	2	3	4
Class	1	2	3	4	5	6	7	8	9	10

Table: Classes for the kinetic Monte Carlo (n-fold way) algorithm proposed by Bortz et. al. [2]. Corresponding to each class i there is a probability p_i.

- Altogether we have ten possible states, depending on the number of neighbors the central spin is surround by.
- Each of these states we assign a class.
- Assume further that the transition probability between states is given by

$$p = \frac{x}{1+x} \quad \text{with} \quad x = \exp\{-\Delta\mathcal{H}/k_B T\}\ , \tag{14}$$

then all possible transitions $r_{i,}$ are given.

Rejection-Free Monte Carlo XI

- One possibility is to use Eq. 9 to draw time increments for the event to happen.
- This algorithm is known as the first-reaction method [3].
- For this we generate a random number $\rho \in (0,1)$ and compute

$$t_{ij} = -r_{ij}^{-1} \ln(\rho) \ . \tag{15}$$

- Thus, for every state change we know the probability and the first passage times.
- What remains to do is to identify the state change $i \to j$. For this we select the reaction coordinate that comes first in time

$$\Delta t = \min_{ij} t_{ij} \ . \tag{16}$$

- Then this state change is performed and time advances (see Algorithm 3)

$$t = t + \Delta t \ . \tag{17}$$

Rejection-Free Monte Carlo XII

.

Algorithm 3 First Reaction Monte Carlo Algorithm

```
1: Initial time $t = 0$
2: Choose initial state $i$
3: for n-of-samples do
4:     Set up list of transition rates $r_{ij}$ (size $N$)
5:     Generate $N$ random numbers $\rho_j$ from a uniform distribution on $(0, 1]$
6:     $t_{ij} = r_{ij}^{-1} \ln(\rho_j^{-1})$
7:     $\Delta t = \min_{ij} t_{ij}$
8:     Carry out event $i \to j$ that is minimum
9:     Update $t = t + \Delta t$
10:    $i \leftarrow j$
11: end for
```

Rejection-Free Monte Carlo XIII

- Hence, we only perform those state changes that actually occur.
- This is in contrast to the procedure that we have developed so far.
- Note that this algorithm uses $\mathcal{O}(N)$ to build the list of transition rates, $\mathcal{O}(N)$ for the number of random numbers and $\mathcal{O}(N)$ to determine the minimum time.
- The obvious difference to the Metropolis Monte Carlo algorithm is that time does not advance in fixed increments but rather leaps in non-constant strides.
- It must further be pointed out that the transition probabilities change at every step.
- Indeed, one of the key features is that the distribution of rates is coupled to the state space [4] and can change.
- For the Ising case there is no such problem.
- This can be seen when we consider the two-dimensional case shown in Diagram 3.

↑	↑	↓
↑	↓	↑
↑	↑	↑

- This translates into the class scheme from Table 1.

2	3	10
2	4	3
1	2	2

Rejection-Free Monte Carlo XIV

- A spin flip can change the transition probability and with it the class.
- The starting point is a choice of a state the system is started in.
- This determines the possible states that the system can transition into and the corresponding rates r_{ij}.
- The next step is to compute the sum over all the possible rates from i to j, i.e. all possible reaction paths.
- The next step then is to pick one of the possible reaction paths with equal probability followed by advancing the time as shown in Algorithm 4.

Rejection-Free Monte Carlo XV

.

Algorithm 4 Kinetic Monte Carlo Algorithm

```
1:  Initial time $t = 0$
2:  Choose initial state $i$ at random
3:  for n-of-samples do
4:      Set up list of transition rates $r_{ij}$ (size $N$)
5:      Compute $R_{i,j} = \sum_{k=1}^{i} r_{ik}$ for $j = 1, ..., N$
6:      Compute $R_i = R_{i,N}$
7:      Generate $\rho$ from a uniform distribution on $(0, 1]$
8:      Choose $i$ such that $R_{i,j-1} < \rho R_i \leq R_{ij}$
9:      Carry out event $j$
10:     Update $i \to j$
11:     Generate $\rho$ from a uniform distribution on $(0, 1]$
12:     $\Delta t = R_i^{-1} \ln(\rho^{-1})$
13:     $t = t + \Delta t$
14: end for
```

Rejection-Free Monte Carlo XVI

- The beauty of the kinetic Monte Carlo Method is that it easily generalizes to arbitrary states and reactions.
- This is why has been used for many condensed matter systems [5–8] with certain refinements [9–13] and coupled to molecular dynamics [14].
- Further developments are the coarse-grained kinetic Monte Carlo [15**?**] and the first-passage kinetic Monte Carlo algorithm [16].
- Let us return to the initial example of the Ising Model.
- Let n_i be the number of spins in class i (see Table 1), then we need to choose the relative weights $n_i p_i$ according to Algorithm 4 and once a class has been chosen a spin in that class is chosen with probability $1/n_i$.
- Fichthorn and Weinberg [17] showed that under the condition of detailed balance and the effective independence of the events, the Algorithm 4 yields a Poisson process and that static and dynamic properties are consistent with the Hamiltonian dynamics [18].
- However, detailed balance is not necessary!
- As we will see later, the kinetic Monte Carlo method is used for non-equilibrium situation and where detailed balance is not fulfilled but global balance is achieved.
- Note that number of operation, i.e. the complexity is $O(N)$. Makysm [19] showed that using a binning method and recursive search trees, the complexity can be brought down to $O(\log_2 N)$ [10].
- For completeness, even though we are in the chapter on rejection-free Monte Carlo, here is a rejection algorithm for the model pair (Ω, Q)

Rejection-Free Monte Carlo XVII

.

Algorithm 5 Rejection Kinetic Monte Carlo Algorithm

```
1:  for n-of-samples do
2:      Set up list of transition rates $r_n$ (size $N$)
3:      Compute an estimator for the sum of rates $\bar{r}$
4:      while state not selected do
5:          Generate $\rho$ from a uniform distribution on $[0, N)$
6:          Compute $n = (Int)(\rho) + 1$
7:          Select $n$ if $n - \rho < r_n/\bar{r}$
8:      end while
9:      $n$ is new state
10: end for
```

Rejection-Free Monte Carlo XVIII

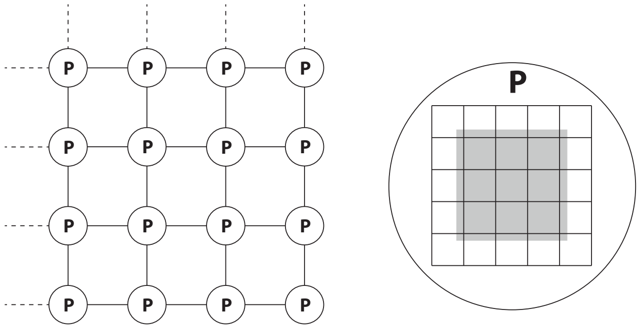

Figure: For the simplest parallel kinetic Monte Carlo algorithm, we assume that the topology for the processors is that of a lattice (for simplicity here a simple square lattice with the processors (P) at the nodes of the lattice) with possible periodic boundary condition (dashed lines). The solid lines represent bi-directional communiation channels (lhs). The rhs panel shows the possibility that a processor has been assigned more than site, say for the 2-D Ising model, L/l lattice sites. The gray shaded area is the part where no communication between the processors is needed for a decision to flip a spin.

Rejection-Free Monte Carlo XIX

- For the Ising model, Lubachevsky [20] has succeeded to parallelize the Monte Carlo algorithm based on the ideas put forward in the more general context by Chandy and J. Misra [21, 22].
- He formulated the algorithm as a distributed discrete-event system.
- Various methods have been designed specifically with lattice models at focus [23–25].
- Also the scaling properties of these type of algorithms have been investigated [26, 27] associating the development of the individual time increments at the individual processors with time increments corresponding to depositions and thus identifying this with surface growth (Kardar-Parisi-Zhang equation [28]).
- The parallelization of the τ-leap has been done by Xu et.al. [29] and for the presence of long-range interactions see [30].
- Also much effort has gone into parallelization of the Gillespie ansatz, for example, Komarov [31].
- The key problem in the parallelization is to avoid event time incompatibilities with communications.
- The solution that Lubachevsky [20] has put forward is the strict synchronization (c.f. Figure 3 and Algorithm 6).

Rejection-Free Monte Carlo XX

- The problem is solved in this algorithm using a global synchronization at the slight expense of efficiency.
- The algorithm presented here is aiming at the above outlined Ising situation.
- We assume two functions $\mathrm{nextState}(i, t_i, \mathrm{neighbours}(i))$ which calls upon the neighbor processors for the corresponding states s_j and $\mathrm{nextTime}(t_i)$ delivers the next time.

Rejection-Free Monte Carlo XXI

.

Algorithm 6 Lubachevsky Parallel Monte Carlo Algorithm

```
1: $s' = s_i$
2: $t' = t_i$
3: for n-of-samples do
4:     if $t_i \leq \min_{j \in \text{neighbours}(i)} t_{ij}$ then
5:         $s' = \text{nextState}(i, t_i, \text{neighbours}(i))$
6:         $t' = \text{nextTime}(t_i)$
7:         Global synchronize
8:         $t = t'$
9:         $s' = s$
10:        Global synchronize
11:    else
12:        Global synchronize
13:        Global synchronize
14:    end if
15: end for
```

Rejection-Free Monte Carlo XXII

- Assume that in the Ising case the lattice is much larger that the number of processor and that there are L/l lattice sites per processors (i.e. $L \times L$ lattice with $l \times l$ blocks).
- There are now interior and boundary sites to be handled by the Algorithm 6. Korniss et. al. [32] have argued that the synchronization steps in the algorithm are not necessary.
- If the same random number generator runs on each of the processors with the same initial seed, they argue that the probability of equal-time nearest-neighbor updates is of measure zero.
- Thus they suggest to treat the interior spins (gray shaded region in Figure 3) like regular spins and use

$$p = \min\{1, \exp(-\Delta H/kT)\} \tag{18}$$

  with $\Delta t = -\ln(\rho)$ (ρ the random number) advancement in time.
- For the boundary spins the criterion in Algorithm 6 is applied.
- To ensure freedom of a deadlock a barrier is used for the boundary spins with a *wait until* the local time t becomes less than or equal to the same quantity for the neighbours.

Rejection-Free Monte Carlo XXIII

- For the kinetic Monte Carlo algorithm for the Ising model Lubachevsky [20] introduced an additional class N_b on top of the 10 classes for the boundary spins. Assume as above that the linear system size is L and that there are $4l$ boundary spins per processor.
- Then $N_b = 4(l-1)$.
- The basic idea is to use the original Monte Carlo, for example Metropolis Monte Carlo, for the boundary spins and for the interior spins the kinetic Monte Carlo.
- Thus the algorithms proceeds as outlined in Algorithm 7.
- For this we augment the 10 classes with the additional class N_b.

Rejection-Free Monte Carlo XXIV

.

Algorithm 7 Lubachevsky Parallel Kinetic Monte Carlo Algorithm

```
1: Initial time $t = 0$
2: for n-of-samples do
3:     Set up list of transition rates $r_i = n_i p_i$ plus $N_b$
4:     Compute $R_k = \sum_{i=1}^{i} r_i$
5:     Generate $\rho$ from a uniform distribution on $(0, 1]$
6:     Choose $i$ such that $R_i < \rho R_i \leq R_i$
7:     Choose a spin with equal probability within the class $i$
8:     if  spin is within the interior then
9:         Flip the spin
10:    else
11:        Wait until the local simulated time $\leq$ neighbour processor
12:        Apply Metropolis Monte Carlo to the spin
13:    end if
14:    Update time
15: end for
```

Rejection-Free Monte Carlo XXV

- A slightly different approach has been taken by Martinez [33] by a synchronous time decomposition of the master equation (synchronous parallel kMC method (spkMC)).
- The basic idea is to create so called null events advancing the internal clock of each processor. This is done without altering the stochastic trajectory of the system.
- Further developments have been done specifically for the reaction-diffusion problems (see [34] and references therein).
- Due to the success of other parallelization algorithms on GPUs, an algorithm was proposed by Jimenez and Ortiz [35], Klingbeil [36] and Agostino et. al. [37].
- Also discrete-event approaches have been developed [38] specifically for the Gillespie ansatz.

Lifting I

- As the name of this section suggests we augment the state space Ω with one or more additional variables.
- Let us first examine this idea for the Ising model in the case of conserved energy.
- Assume that we add the extra variable or degree of freedom to the Hamiltonian [39] (13)

$$\mathcal{H}' = e - \sum_{<i,j>} S_i S_j \quad S_i = \pm 1 \tag{19}$$

  with $\Omega' = \mathbb{N} \times \Omega$.
- The extra variable e allows to lift the system out of the otherwise constraint hyperspace of constant energy.
- Set e to an appropriate value according to the initial energy.
- We can construct a Markov chain by choosing a spin at site ν at random.
- We change the spin direction at site ν to obtain $\Delta\mathcal{H}$ for the energy change in the Ising Hamiltonian.
- If we loose energy, then we transfer the energy to e and accept the change.
- If we would gain energy, then we accept the change under the condition that e has enough energy.

Lifting II

- Let us now look at the more general case. Chen et. al. and others [40–44] constructed a non-reversible Markov chain Monte Carlo Method (Lifted Metropolis-Hastings) as for example also in the (Hamiltonian) Hybrid Monte [45] (see also for the Bouncy Particle Sampler method [46]).
- The effect of this lifting is a reduced mixing time of the Markov chain (at best reduced by the square root of the original time).
- So far we almost always used the detailed balance condition for the transition probability W and the invariant distribution p which we want to obtain from a Markov chain

$$p(x)W(x,x') = p(x')W(x',x) \text{ for all } x,x' \in \Omega \ . \tag{20}$$

- This is not a necessary but sufficient a condition for the transition probability.
- One of possible solutions to Eq. 20 is the Metropolis Hastings transition probability

$$W(x,x') = q(x|x') \min\left\{1, \frac{p(x')q(x|x')}{q(x'|x)p(x)}\right\} \tag{21}$$

with the propositional probability q. The Hybrid Monte Carlo Method [45] has made use of this propositional probability.

Lifting III

- Consider the global balance condition for the transition probability W [47]

$$\int p(x)W(x,x')dx = \int p(x')W(x',x)dx' \tag{22}$$

which we need to really to fulfill and the constraint

$$W(x,x')W(x',x) = 0 \text{ for all } x,x' \in \Omega\,. \tag{23}$$

- W's that fulfill criterion 22 and criterion 23 are said to check a maximal global balance condition [48, 49].
- Following the idea of adding additional degrees of freedom, we augment the system by an auxiliary variable e. Thus for the distribution p this results in

$$p(x,e) = p(x)p(e) \tag{24}$$

and for the above example (19) this would be

$$p(e) \propto \exp\{-\beta e\}\,. \tag{25}$$

and fix the propositional probability as

$$q(x',e|x,e) = \begin{cases} 1, & \text{if } x' = x + e\Delta s \\ 0, & \text{otherwise} \end{cases} \tag{26}$$

Lifting IV

where the statement $x' = x + e\Delta s$ is meant to express that x' and x should not differ too much.

- Thus we updated the state in the direction given by e. This is continued until rejection occurs.
- Then we choose a new e' and continue with (x', e') which lifts the rejection into the lifting space rendering the entire method rejection-free.
- The probability for the choice of e' is based on the condition 22.

Event-Chain Monte Carlo I

- We will extend the rejection-free Monte Carlo simulation methods by considering irreversible Markov chains drawing on idea by Peters [50] and the concept of lifting [41].
- These methods have been successfully developed for the problem of melting in two dimensions [51–54].
- Extensions have been derived for discrete-variable models [55], classical continuous spin models [56, 57] and further generalized to rejection-free global-balance algorithms [58] and the forward event-chain Monte Carlo algorithm [47].
- Here we follow [47] in the exposition of the algorithm.
- The goal is to use the ideas of lifting developed in the previous section to develop a rejection-free Monte Carlo algorithm.
- We use the extra variable e to suggest a new state.
- Rather than using an except/reject on this, we choose a time for the new event to happen and sample all the state in a chain along the way, until we have reached the transition time. We then choose a new e value and continue.
- To sample the time Δs we go about as in (9) and (15).
- For ease presentation we follow the mechanistic language and assume an energy function $E(x)$ and consider e a velocity (see also Hybrid Monte Carlo [45]).

Event-Chain Monte Carlo II

- Thus in Eq. 26 we are looking for displacements in space controlled by the time Δs and the velocity e.
- In Eq (26) we have made a choice for the propositional probability.
- With the notation $[a]^+ = \max\{0, a\}$ and Metropolis choice of transition probability (21) we have

$$W(x, x') = \min\{1, \exp\{-\Delta E(x)e\}\} = \exp\{-[\Delta E(x)e]^+\} \ . \tag{27}$$

- To determine the transition time we add up all the moves until we have reached the event time

$$\Delta E^*(\Delta s) = \int_0^{\Delta s} [\nabla E(x + se)e]^+ ds \tag{28}$$

  and find the time Δs by solving the equation

$$\Delta E^*(\Delta s) = \log(\rho) \tag{29}$$

  where $\rho \in (0, 1]$ is a uniform random number. It rests to choose the transition probability for e. Here Michel and Senecal [47] suggest

$$p(e' \to e) = \delta(e' + e) \tag{30}$$

Event-Chain Monte Carlo III

- In Algorithm 8 the full algorithm is exposed (for parallelization for example for dense hard sphere and polymer systems see [59]).

Event-Chain Monte Carlo IV

Algorithm 8 Event Chain Monte Carlo Algorithm [47]

```
 1: Initial state $x' = x_0$
 2: for n-of-samples do
 3:     Set current event chain length $l_c = l$
 4:     Set random direction $e$
 5:     while True do
 6:         Set initial sample $x = x'$
 7:         Compute $\Delta E^* = -\log(\rho)$, $\rho$ from a uniform distribution on $(0, 1]$
 8:         Compute $\Delta s$
 9:         if $l_c < \Delta s$ then
10:             Compute $x' = x + l_c e$
11:             Set sample $x^k = x'$
12:             Break
13:         else
14:             Compute $x' = x + \Delta s e$
15:             Update chain length $l_c = l_c - \Delta s$
16:             Update direction $-e$
17:         end if
18:     end while
19: end for
```

Bibliography I

[1] David G. Kendall. Random fluctuations in the age-distribution of a population whose development is controlled by the simple "birth-and-death" process. 12(2):278–285, 1950.

[2] A. B. Bortz, M. H. Kalos, and J. L. Lebowitz. A new algorithm for monte carlo simulation of ising spin systems. *Journal of Computational Physics*, 17(1):10–18, 1975.

[3] Daniel T Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4):403–434, 1976.

[4] Tim P. Schulze. Efficient kinetic monte carlo simulation. *J. Comput. Phys. 0021-9991*, 227(4):2455–2462, 2008.

[5] W M Young and E W Elcock. Monte carlo studies of vacancy migration in binary ordered alloys: I. *Proceedings of the Physical Society*, 89(3):735, 1966.

[6] B. Meng and W. H. Weinberg. Dynamical monte carlo studies of molecular beam epitaxial growth models: interfacial scaling and morphology. *Surface Science*, 364(2):151–163, 1996.

[7] Vasily V. Bulatov and Wei Cai. *Computer Simulations of Dislocations (Oxford Series on Materials Modelling*. Oxford Univ Press, 2006.

Bibliography II

[8] Stephan A. Baeurle, Takao Usami, and Andrei A. Gusev. A new multiscale modeling approach for the prediction of mechanical properties of polymer-based nanomaterials. *Polymer*, 47(26):8604–8617, 12 2006.

[9] K. Binder and M.H. Kalos. In K. Binder, editor, *Monte Carlo Methods in Statistical Physics*, volume 7 of *Springer Topics in Current Physics*, page 225. Springer-Verlag Berlin Heidelberg, 1979.

[10] James L. Blue, Isabel Beichl, and Francis Sullivan. Faster monte carlo simulations. *Physical Review E*, 51(2):R867–R868, 02 1995.

[11] Michael A. Gibson and Jehoshua Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *The Journal of Physical Chemistry A*, 104(9):1876–1889, 03 2000.

[12] T. P. Schulze. Kinetic monte carlo simulations with minimal searching. *Physical Review E*, 65(3):036704–, 02 2002.

[13] James M. McCollum, Gregory D. Peterson, Chris D. Cox, Michael L. Simpson, and Nagiza F. Samatova. The sorting direct method for stochastic simulation of biochemical systems with varying reaction execution behavior. *Computational Biology and Chemistry*, 30(1):39–49, 2 2006.

[14] Angela Violi, Adel F. Sarofim, and Gregory A. Voth. Kinetic monte carlo–molecular dynamics approach to model soot inception. *Combustion Science and Technology*, 176(5-6):991–1005, 05 2004.

Bibliography III

[15] M.A. Katsoulakis 3 (2005) A. Chatterjee, D.G. Vlachos. *International Journal for Multiscale Computational Engineering*, 3(135), 2005.

[16] Aleksandar Donev, Vasily V. Bulatov, Tomas Oppelstrup, George H. Gilmer, Babak Sadigh, and Malvin H. Kalos. A first-passage kinetic monte carlo algorithm for complex diffusion–reaction systems. *Journal of Computational Physics*, 229(9):3214–3236, 2010.

[17] Kristen A. Fichthorn and W. H. Weinberg. Theoretical foundations of dynamical monte carlo simulations. *The Journal of Chemical Physics*, 95(2):1090–1096, 2017/01/03 1991.

[18] Santiago A. Serebrinsky. Physical time scale in kinetic monte carlo simulations of continuous-time markov chains. *Physical Review E*, 83(3):037701–, 03 2011.

[19] P A Maksym. Fast monte carlo simulation of mbe growth. *Semiconductor Science and Technology*, 3(6):594, 1988.

[20] Boris D Lubachevsky. Efficient parallel simulations of dynamic ising spin systems. *Journal of Computational Physics*, 75(1):103–122, 1988.

[21] K. M. Chandy and J. Misra. Distributed simulation: A case study in design and verification of distributed programs. *IEEE Transactions on Software Engineering*, SE-5(5):440–452, 1979.

[22] J. Misra. Distributed discrete-event simulation. *ACM Comput. Surv.*, 18:39, (986.

Bibliography IV

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

[23] Yunsic Shim and Jacques G. Amar. Semirigorous synchronous sublattice algorithm for parallel kinetic monte carlo simulations of thin film growth. *Physical Review B*, 71(12):125432–, 03 2005.

[24] Giorgos Arampatzis, Markos A. Katsoulakis, Petr Plecháč, Michela Taufer, and Lifan Xu. Hierarchical fractional-step approximations and parallel kinetic monte carlo algorithms. *Journal of Computational Physics*, 231(23):7795–7814, 2012.

[25] Ignacio Martin-Bragado, J. Abujas, P. L. Galindo, and J. Pizarro. Synchronous parallel kinetic monte carlo: Implementation and results for object and lattice approaches. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 352:27–30, 2015.

[26] G. Korniss, Z. Toroczkai, M. A. Novotny, and P. A. Rikvold. From massively parallel algorithms and fluctuating time horizons to nonequilibrium surface growth. *Physical Review Letters*, 84(6):1351–1354, 02 2000.

[27] G. Korniss, M. A. Novotny, H. Guclu, Z. Toroczkai, and P. A. Rikvold. Suppressing roughness of virtual times in parallel discrete-event simulations. *Science*, 299(5607):677, 01 2003.

[28] Mehran Kardar, Giorgio Parisi, and Yi-Cheng Zhang. Dynamic scaling of growing interfaces. *Physical Review Letters*, 56(9):889–892, 03 1986.

[29] L. Xu, M. Taufer, S. Collins, and D. G. Vlachos. Parallelization of tau-leap coarse-grained monte carlo simulations on gpus. In *2010 IEEE International Symposium on Parallel & Distributed Processing (IPDPS)*, pages 1–9, 2010.

Bibliography V

[30] Jens Nielsen, Mayeul d'Avezac, James Hetherington, and Michail Stamatakis. Parallel kinetic monte carlo simulation framework incorporating accurate models of adsorbate lateral interactions. *The Journal of Chemical Physics*, 139(22):224706, 2018/08/20 2013.

[31] Ivan Komarov and Roshan M. D'Souza. Accelerating the gillespie exact stochastic simulation algorithm using hybrid parallel execution on graphics processing units. *PLOS ONE*, 7(11):e46693–, 11 2012.

[32] G Korniss, M. A Novotny, and P. A Rikvold. Parallelization of a dynamic monte carlo algorithm: A partially rejection-free conservative approach. *Journal of Computational Physics*, 153(2):488–508, 1999.

[33] E. Martínez, P. R. Monasterio, and J. Marian. Billion-atom synchronous parallel kinetic monte carlo simulations of critical 3d ising systems. *Journal of Computational Physics*, 230(4):1359–1369, 2011.

[34] Weiliang Chen and Erik De Schutter. Parallel steps: Large scale stochastic spatial reaction-diffusion simulation with high performance computers. *Frontiers in Neuroinformatics*, 11:13, 2017.

[35] F. Jiménez and C. J. Ortiz. A gpu-based parallel object kinetic monte carlo algorithm for the evolution of defects in irradiated materials. *Computational Materials Science*, 113:178–186, 2016.

Bibliography VI

[36] Guido Klingbeil, Radek Erban, Mike Giles, and Philip K. Maini. Stochsimgpu: parallel stochastic simulation for the systems biology toolbox 2 for matlab. *Bioinformatics*, 27(8):1170–1171, 04 2011.

[37] Daniele D. Agostino, Giulia Pasquale, Andrea Clematis, Carlo Maj, Ettore Mosca, Luciano Milanesi, and Ivan Merelli. Parallel solutions for voxel-based simulations of reaction-diffusion systems. *BioMed Research International*, 2014:10, 2014.

[38] Lorenzo Dematté and Tommaso Mazza. On parallel stochastic simulation of diffusive systems. In Monika Heiner and Adelinde M. Uhrmacher, editors, *Computational Methods in Systems Biology*, pages 191–210, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[39] Michael Creutz. Microcanonical monte carlo simulation. *Physical Review Letters*, 50(19):1411–1414, 05 1983.

[40] Fang Chen, László Lovász, and Igor Pak. Lifting markov chains to speed up mixing, 1999.

[41] Persi Diaconis, Susan Holmes, and Radford M. Neal. Analysis of a nonreversible markov chain sampler. pages 726–752, 2000.

[42] Thomas P. Hayes and Alistair Sinclair. Liftings of tree-structured markov chains. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 6302 of *Lecture Notes in Computer Science*. Springer-Verlag Berlin Heidelberg, 2010.

Bibliography VII

[43] Konstantin S. Turitsyn, Michael Chertkov, and Marija Vucelja. Irreversible monte carlo algorithms for efficient sampling. *Physica D: Nonlinear Phenomena*, 240(4–5):410–414, 2 2011.

[44] Marija Vucelja. Lifting—a nonreversible markov chain monte carlo algorithm. *American Journal of Physics*, 84(12):958–968, 2017/01/06 2016.

[45] Simon Duane, A. D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics Letters B*, 195(2):216–222, 1987.

[46] Alexandre Bouchard-Côté, Sebastian J. Vollmer, and Arnaud Doucet. The bouncy particle sampler: A nonreversible rejection-free markov chain monte carlo method. *Journal of the American Statistical Association*, 113(522):855–867, 04 2018.

[47] Manon Michel and Stéphane Sénécal. *Forward Event-Chain Monte Carlo: a general rejection-free and irreversible Markov chain simulation method*. 02 2017.

[48] Joris Bierkens, Paul Fearnhead, and Gareth Roberts. *The Zig-Zag Process and Super-Efficient Sampling for Bayesian Analysis of Big Data*. 07 2016.

[49] Joris Bierkens and Gareth Roberts. A piecewise deterministic scaling limit of lifted metropolis-hastings in the curie-weiss model. *Ann. Appl. Probab.*, 27(2):846–882, 2017.

[50] E. A. J. F. Peters and G. de With. Rejection-free monte carlo sampling for general potentials. *Physical Review E*, 85(2):026703–, 02 2012.

Bibliography VIII

[51] Etienne P. Bernard, Werner Krauth, and David B. Wilson. Event-chain monte carlo algorithms for hard-sphere systems. *Physical Review E*, 80(5):056704–, 11 2009.

[52] Etienne P. Bernard and Werner Krauth. Two-step melting in two dimensions: First-order liquid-hexatic transition. *Physical Review Letters*, 107(15):155704–, 10 2011.

[53] Sebastian C Kapfer and Werner Krauth. Sampling from a polytope and hard-disk monte carlo. *Journal of Physics: Conference Series*, 454(1):012031, 2013.

[54] Sebastian C. Kapfer and Werner Krauth. Two-dimensional melting: From liquid-hexatic coexistence to continuous transitions. *Physical Review Letters*, 114(3):035702–, 01 2015.

[55] Alejandro Mendoza-Coto, Rogelio Díaz-Méndez, and Guido Pupillo. Event-driven monte carlo: Exact dynamics at all time scales for discrete-variable models. *EPL (Europhysics Letters)*, 114(5):50003, 2016.

[56] Manon Michel, Johannes Mayer, and Werner Krauth. Event-chain monte carlo for classical continuous spin models. *EPL (Europhysics Letters)*, 112(2):20003, 2015.

[57] Yoshihiko Nishikawa, Manon Michel, Werner Krauth, and Koji Hukushima. Event-chain algorithm for the heisenberg model: Evidence for $z\ensuremath{\simeq}1$ dynamic scaling. *Physical Review E*, 92(6):063306–, 12 2015.

Bibliography IX

[58] Manon Michel, Sebastian C. Kapfer, and Werner Krauth. Generalized event-chain monte carlo: Constructing rejection-free global-balance algorithms from infinitesimal steps. *The Journal of Chemical Physics*, 140(5):054116, 2018/08/13 2014.

[59] Tobias A. Kampmann, Horst-Holger Boltz, and Jan Kierfeld. Parallelized event chain algorithm for dense hard sphere and polymer systems. *J. Comput. Phys. 0021-9991*, 281(C):864–875, 2015.

Index I