Algorithms and computations in physics (Oxford Lectures 2025)

Werner Krauth*
Laboratoire de Physique, Ecole normale supérieure, Paris (France)
Rudolf Peierls Centre for Theoretical Physics & Wadham College
University of Oxford (UK)

Second lecture: 28 January 2025 (version: 27 January 2025)

In this second lecture, we consider Markov-chain sampling, from the adults' game on the Monte Carlo heliport to modern ideas on non-reversibility. We discuss a lot of theory, but also six ten-line pseudo-code algorithms, none of them approximate, and all of them as intricate as they are short.

Contents

2 Markov-chain sampling

We discuss Markov chains, initially in a practical context, but then in (almost) full mathematical rigor. We then walk through an exhibition of algorithms that illustrate some major themes: the Metropolis algorithm and its modern variants, non-reversibility and continuous-time Markov processes.

Background material for this second lecture is contained in the book by Levin et al. [2], which treats the basic theory of Markov chains, transition matrices, mixing and relaxation times. A recent educational paper with Tartero [3] provides more details on the example algorithms. The heliport game and the Metropolis–Hastings algorithm are treated in more depth in Ref. [4].

*werner.krauth@ens.fr, werner.krauth@physics.ox.ac.uk

2.1 Adults on the Monte Carlo heliport

Markov-chain sampling—believe it or not—comes from a game that adults play on the Monte Carlo heliport in the early evenings, with all the helicopters safely stowed away (see Fig. 2.1). Up to a rescaling of lengths, it realizes the same sample space $\Omega^{\square}$ as the children had on the beach.

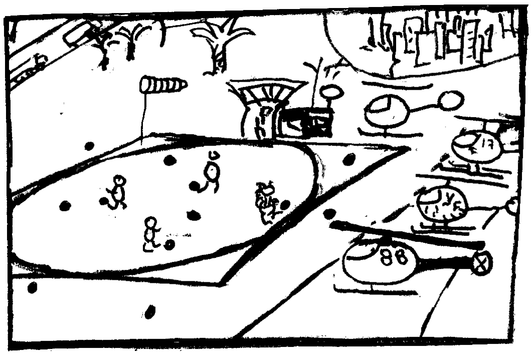

Figure 2.1: Adults throwing pebbles on the Monte Carlo heliport, and computing the number π. The game always starts at the clubhouse. A curious procedure is followed when a pebble falls outside the pad.

Adults fling pebbles inside the square but, because of its sheer size, they cannot possible place pebbles randomly. So they use a different algorithm. They start at the clubhouse (position $(1, 1)$),[1] and carry along with them as many pebbles as their handbags will hold, then throw a pebble randomly around them, go to where it has landed, then throw again, and again. When they throw a pebble outside the square of the heliport, they fetch it and have it placed on top of the pebble that marked their last position, then continue. To explain exactly what we mean with the adults' algorithm, we again resort to pseudo-code (see Alg. 2.1 (`markov-pi`)).

procedure `markov-pi`
$N_{\text{hits}} \leftarrow 0;\ \{x, y\} \leftarrow \{1, 1\}$
for $i = 1, \ldots, N$**:**
$$\left\{\begin{array}{l} \Delta_x \leftarrow \texttt{ran}(-\delta, \delta) \\ \Delta_y \leftarrow \texttt{ran}(-\delta, \delta) \\ \textbf{if } |x + \Delta_x| < 1 \textbf{ and } |y + \Delta_y| < 1\textbf{:}\quad \textbf{then} \\ \quad \left\{\begin{array}{l} x \leftarrow x + \Delta_x \\ y \leftarrow y + \Delta_y \end{array}\right. \\ \textbf{if } x^2 + y^2 < 1\textbf{:}\quad N_{\text{hits}} \leftarrow N_{\text{hits}} + 1 \end{array}\right.$$
output N_{hits}
——

Algorithm 2.1: `markov-pi`. Markov-chain Monte Carlo algorithm for computing π in the adults' game. The game starts at the clubhouse; the throwing range δ remains fixed.

At the end of the game, the pattern of pebbles looks weird (see Fig. 2.2), and certainly differs from that in the children's game in Lecture 1. However, when the adults count the number of pebbles inside the circle in the square, they again get a decent—if less precise— approximation of the number π, (see Table 2.1, and again [4, Sect. 1.1] for the full story). We remember in this context that probability theory, as codified in the Kolmogoroff axioms, assigns probabilities in continuous sample spaces not to single samples but to subsets of Ω called *events* (see Ref. [1, Chap. 1.3]).

[1] If they could start at a random position, there would be no point in their game

Run	N_{hits}	Estimate of π
1	3123	3.123
2	3118	3.118
3	3040	3.040
4	3066	3.066
5	3263	3.263

Table 2.1: Results of five runs of Alg. 2.1 (`markov-pi`) with $N = 4000$ and a throwing range $\delta = 0.3$

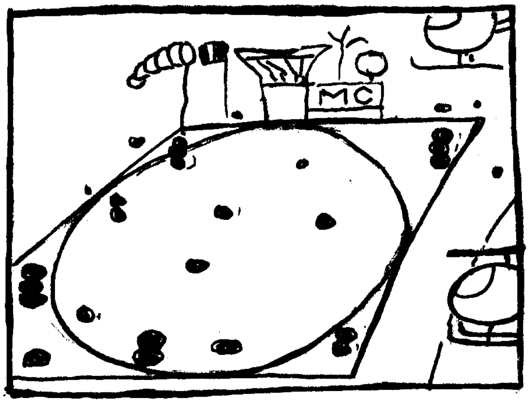

Figure 2.2: Monte Carlo heliport after the game. Piles of pebbles can be seen near the boundaries, and especially near the corners.

The players on the heliport implement the 1953 Metropolis algorithm [5]. In this lecture, we will discuss it, prove its correctness, analyze it, then overcome it and confront it with a number of "beyond-Metropolis" algorithms.

2.1.1 The transition matrix, balance conditions

To discuss the mathematical structure of the adults' game and other Markov chains, we imagine its sample space Ω discretized, with a finite number of samples $x \in \Omega$. The possible moves, from x to x', then have probabilities that constitute a transition matrix $P(x, x')$. The initial configuration, at the clubhouse $(x, y) = (1, 1)$ means that the probability distribution $\pi^{\{t=0\}}$, is a Kronecker δ function at $(1, 1)$ and not at all what we want it to be, namely constant on the landing pad. We study how $\pi^{\{t=0\}} = \delta(1, 1)$ gives rise to wider and wider distributions $\pi^{\{t=1\}}, \pi^{\{t=2\}}, \ldots,$ as time evolves, and then becomes flat, in the late stages of the heliport game, so to say. The connection between time $t-1$ and time t is made by what is called a transition matrix P, which has a double meaning. On the one hand, it encodes a Monte Carlo algorithm: $P(x, x')$ *is defined as* the (conditional) probability to move from x to x' in one step, The condition $\sum_{x'} P(x, x') = 1$ expresses the conservation of probabilities. On the other hand, the transition matrix gives the relationship between the probability distributions $\pi^{\{t-1\}}$ and $\pi^{\{t\}}$ at subsequent time steps $t-1$ and t:

$$\pi^{\{t\}}(x) = \sum_{x' \in \Omega} \pi^{\{t-1\}}(x') P(x', x). \tag{2.1}$$

At time $t = 0$, $\pi^{\{t=0\}}$ is a function that we can sample. Most of the time, it is a single configuration (in our example, it's a Delta function at the clubhouse).

In order for the distribution $\pi^{\{t\}}$ at late times to correspond to the distribution π (in the example of the heliport, for it to be uniform in the square), we may drop the time indices in eq. (2.1), and arrive at the *global-balance condition*

$$\pi(x) = \sum_{x' \in \Omega} \pi(x') P(x', x) \quad \forall x \in \Omega, \tag{2.2}$$

a necessary condition on the transition matrix P (in other words, the Monte Carlo algorithm) to converge towards π. For a transition matrix that is *irreducible*, the global-balance condition is satisfied for a unique stationary distribution π. "Irreducible" means that (for a finite Ω) the probability to move in a finite time from any x to any x' is finite.

Any irreducible transition matrix has a unique π, but this distribution is not necessarily the limit $\pi^{\{t\}}$ for $t \to \infty$ for all initial distributions $\pi^{\{0\}}$. Convergence towards π of an irreducible Markov chain requires that it is aperiodic, that is, that the return times from a configuration i back to itself $\{t \geq 1 : (P^t)(x, x) > 0\}$ are not all multiples of a period larger than one. For irreducible, aperiodic transition matrices, $P^t = (P^t)(x, x')$ is a positive matrix for some fixed t, and the Markov chain converges towards π from any starting distribution $\pi^{\{0\}}$.

In conclusion, Markov chains must satisfy the global-balance condition of eq. (2.2) in order to converge towards the imposed distribution π. In addition, they must satisfy an—easily verifiable—irreducibility requirement to make sure that they can go everywhere in the sample space Ω. Finally, they should be aperiodic, but that is easy to achieve.

2.1.2 Reversible and non-reversible Markov chains

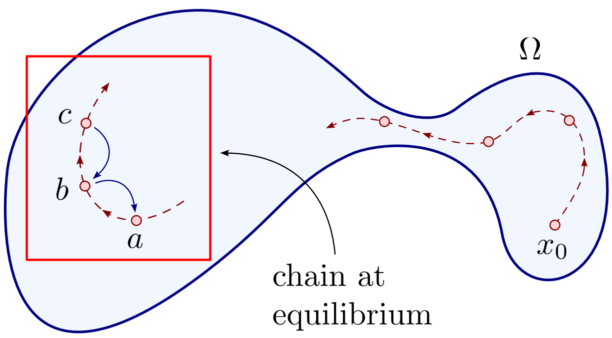

Figure 2.3: Motion of a Markov chain in equilibrium. For a reversible Markov chain in equilibrium, the trajectory $a \to b \to c$ appears with the same probability as the "reverse" trajectory $c \to b \to a$.

Reversible algorithms are those that satisfy the *detailed-balance condition*

$$\pi(x) P(x, x') = \pi(x') P(x', x) \quad \forall x, y \in \Omega. \tag{2.3}$$

Detailed balance implies global balance (eq. (2.3) yields eq. (2.2) by summing over x', considering that $\sum_{x' \in \Omega} P(x, x') = 1$). The detailed-balance condition imposes that in equilibrium, the path from x (at time $t - 1$) to x' (at time t) is equally likely as the time-reversed path from x' to x. This can be extended to paths that are arbitrary long (see Fig. 2.3), and explains why Markov chains that satisfy the detailed-balance condition are equivalently called *reversible*.

To set up a reversible transition matrix P for a given distribution π, we may choose

$$\pi(x) P(x, x') \propto \min\left[\pi(x), \pi(x')\right] \quad \text{for } x \neq x'. \tag{2.4}$$

The right-hand side of Eq. (2.4) is symmetric in x and x', so that the left-hand side must also be symmetric. Therefore, detailed balance is automatically satisfied. We divide both sides by

$\pi(x)$ and arrive at the equation famously proposed by Metropolis et al. in 1953:

$$P^{\text{Met}}(x, x') \propto \min\left[1, \frac{\pi(x')}{\pi(x)}\right] \quad \text{for } x \neq x'. \tag{2.5}$$

Let us discuss the difference between a transition matrix and a filter in order to make Eq. (2.5) explicit and remove the proportionality sign. The move from x to $x' \neq x$ proceeds in two steps. A possible move is first proposed with an *a priori* probability $\mathcal{A}(x, x')$ and is then accepted or rejected with a filter. In the Metropolis algorithm, the *a priori* probability is symmetric, $\mathcal{A}(x, x') = \mathcal{A}(x', x)$, and

$$\underbrace{P^{\text{Met}}(x, x')}_{\text{transition matrix}} = \underbrace{\mathcal{A}(x, x')}_{a\ priori\ \text{probability}} \overbrace{\mathcal{P}^{\text{Met}}(x, x')}^{\text{Metropolis filter}}. \tag{2.6}$$

For the Metropolis algorithm, a proposed move $x \to x'$ (with $x' \neq x$) is thus accepted with the Metropolis acceptance probability

$$\mathcal{P}^{\text{Met}}(x, x') = \min\left[1, \frac{\pi(x')}{\pi(x)}\right], \tag{2.7}$$

which is also called the *Metropolis filter* in order to differentiate it from the transition matrix. If the move $x \to x'$ is rejected, the particle remains at x, which determines the diagonal transition matrix elements $P(x, x)$ and guarantees that $\sum_{x'} P(x, x') = 1$, in other words, that the transition matrix is *stochastic.*

For the transition matrix P of a reversible Markov chain, the matrix $A_{ij} = \pi_i^{1/2} P_{ij} \pi_j^{-1/2}$ is symmetric, as trivially follows from the detailed balance of eq. (2.3). The spectral theorem then assures that A has only real eigenvalues and that its eigenvectors form an orthonormal basis. The transition matrix P has the same eigenvalues as A, as well as closely related (right) eigenvectors:

$$\sum_{j \in \Omega} \underbrace{\pi_i^{1/2} P_{ij} \pi_j^{-1/2}}_{A_{ij}} x_j = \lambda x_i \quad \Leftrightarrow \quad \sum_{j \in \Omega} P_{ij} \underbrace{\left[\pi_j^{-1/2} x_j\right]}_{\tilde{x}_j} = \lambda \underbrace{\left[\pi_i^{-1/2} x_i\right]}_{\tilde{x}_i}. \tag{2.8}$$

The eigenvectors $\tilde{x}$ of P must be multiplied with $\sqrt{\pi}$ to be mutually orthogonal. They provide a basis on which any initial probability distribution $\pi^{\{0\}}$ can be expanded. An irreducible and aperiodic transition matrix P (reversible or not) has one eigenvalue $\lambda_1 = 1$, and all others satisfy $|\lambda_k| < 1\ \forall k \neq 1$. The unit eigenvalue λ_1 corresponds to a constant right eigenvector of P because of the stochasticity condition $\sum_{j \in \Omega} P_{ij} = 1$, and to the left eigenvector π of P, because of the global-balance condition of eq. (2.2). Let us consider a reversible transition matrix with a non-degenerate spectrum (which must be real, as we just showed), then slightly perturb the elements of P, which will make it non-reversible. As the eigenvalues of a matrix continuously depends on its elements, it follows that a non-reversible transition matrix may very well have a real-valued spectrum of eigenvalues.

2.1.3 Metropolis–Hastings algorithm

In Alg. 2.1 (`markov-pi`), moves (Δ_x, Δ_y) are restricted to a small square of edge length 2δ, the throwing range, and as this throwing range around a position (x, y) is independent of the position, the a priori probability is symmetric, and even constant (see Fig. 2.4A). The small square could be replaced by a small disk without bringing in anything new (see Fig. 2.4B). A more interesting situation arises for asymmetric a priori probabilities: in the triangle algorithm of Fig. 2.4C, moves are sampled from an oriented equilateral triangle centered at a, with one edge parallel to the x-axis. This extravagant choice may lack motivation in the context of the adults' game, but contains a crucial ingredient of modern Monte Carlo algorithms, that we will study in later lectures.

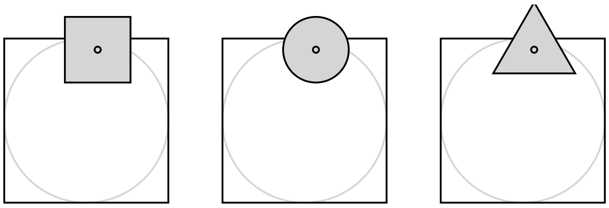

Figure 2.4: Throwing pattern in Alg. 2.1 (`markov-pi`) (A), with variants. The triangle algorithm (C) requires special attention.

The detailed-balance condition of eq. (2.3), in the presence of an asymmetric a priori probability $\mathcal{A}(x, x')$, such as the triangular one, gives

$$\pi(x)\mathcal{A}(x, x')\mathcal{P}(x, x') = \pi(x')\mathcal{A}(x', x)\mathcal{P}(x', x'). \tag{2.9}$$

In this equation, π is given (we want it to be uniform, on the heliport) and so is $\mathcal{A}$ (we want it to be uniform, in an equilateral triangle). The probability of moving from a to b must satisfy $\pi(a)\mathcal{P}(a \to b) = \pi(b)\mathcal{P}(b \to a)$, so that the acceptance probabilities (the filter) must obey:

$$\frac{\mathcal{P}^{\text{Met–H}}(x, x')}{\mathcal{P}^{\text{Met–H}}(x', x)} = \frac{\pi(x')}{\mathcal{A}(x, x')} \frac{\mathcal{A}(x', x)}{\pi(x)}.$$

leading to

$$\mathcal{P}^{\text{Met–H}}(x, x') = \min\left[1, \frac{\pi(x')}{\mathcal{A}(x, x')} \frac{\mathcal{A}(x', x)}{\pi(x)}\right], \tag{2.10}$$

also called the Metropolis–Hastings filter, which always requires one to take into consideration the back move (from x' to x), when one decides whether one should accept the move from x to x'.

2.2 Reversible Markov chains

The material in this section is taken from Ref. [3], which presents a dozen of distinct Markov-chain Monte Carlo algorithms to sample the Boltzmann distribution of the anharmonic oscillator. We present two reversible Markov chains (plus a patch) before continuing with two non-reversible algorithms, in Sec. 2.3. In what follows, we consider the probability distribution

$$\pi_{24}(x) = \exp\left(-x^2/2 - x^4/4\right), \tag{2.11}$$

This distribution (that we already encountered in Lecture 1) has—in principle—a normalization factor, but we will not worry about it. The harmonic part of this distribution is referred to as π_2 and the quartic term as π_4.

$$\pi_2(x) = \exp\left(-x^2/2\right), \quad \pi_4(x) = \exp\left(-x^4/4\right). \tag{2.12}$$

2.2.1 Metropolis algorithm

Algorithm 2.2 (`metropolis`) implements the symmetric *a priori* probability as a uniform displacement $\Delta = x' - x$ which is as likely as $-\Delta$. The Metropolis filter is implemented with a uniform random number Υ between 0 and 1, which we refer to as a "pebble." For large times

t, when the initial configuration is forgotten, the algorithm samples π_{24}. In all the following Markov-chain algorithms, this large t condition is understood.

procedure `metropolis`
input x (sample at time t)
$\Delta \leftarrow \texttt{ran}(-\delta, \delta)$
$x' \leftarrow x + \Delta$
$\Upsilon \leftarrow \texttt{ran}(0, 1)$
if $\Upsilon < \min\left[1, \dfrac{\pi_{24}(x')}{\pi_{24}(x)}\right]$**:** $x \leftarrow x'$
output x (sample at time $t+1$)
——

Algorithm 2.2: `metropolis`. Sampling π_{24} with the Metropolis algorithm.

2.2.2 Factorized Metropolis algorithm

The Metropolis algorithm is really famous, but it is not the end of history. A modern variant is the factorized Metropolis algorithm that we do not discuss here in detail, but only apply to the anharmonic oscillator, where it is written as:

$$\mathcal{P}_{24}^{\text{fact}}(x, x') = \min\left[1, \frac{\pi_2(x')}{\pi_2(x)}\right] \min\left[1, \frac{\pi_4(x')}{\pi_4(x)}\right]. \tag{2.13}$$

The factorized Metropolis algorithm satisfies detailed balance:

$$\begin{aligned} &\pi_{24}(x) P_{24}^{\text{fact}}(x, x') \\ &\propto \underbrace{\pi_2(x) \min\left[1, \frac{\pi_2(x')}{\pi_2(x)}\right]}_{\min[\pi_2(x), \pi_2(x')]:\ x \Leftrightarrow x'} \underbrace{\pi_4(x) \min\left[1, \frac{\pi_4(x')}{\pi_4(x)}\right]}_{\min[\pi_4(x), \pi_4(x')]:\ x \Leftrightarrow x'} \\ &\propto \pi_{24}(x') P_{24}^{\text{fact}}(x', x), \end{aligned} \tag{2.14}$$

where we have dropped the symmetric *a priori* probability $\mathcal{A}$. Algorithm 2.3 (`factor-metropolis`) implements the factorized Metropolis filter: Algorithm 2.3 (`factor-metropolis`) (which is

procedure `factor-metropolis`
input x
$\Delta \leftarrow \texttt{ran}(-\delta, \delta)$
$x' \leftarrow x + \Delta$
$\Upsilon \leftarrow \texttt{ran}(0, 1)$
if $\Upsilon < \min\left[1, \dfrac{\pi_2(x')}{\pi_2(x)}\right] \min\left[1, \dfrac{\pi_4(x')}{\pi_4(x)}\right]$**:**
 $\{\ x \leftarrow x'$
output x
——

Algorithm 2.3: `factor-metropolis`. Sampling π_{24} naively with the factorized Metropolis filter (see Ref. [3]).

naive) can be patched by replacing its random number Υ by two independent random numbers Υ_2 and Υ_4, as shown in Alg. 2.4 (`factor-metropolis(patch)`). There, a proposed move is accepted by *consensus* if all the factors accept it. In Alg. 2.4 (`factor-metropolis(patch)`), two independent decisions are taken,[2] one for the harmonic and one for the quartic factor, and the proposed move is finally accepted only if it is accepted by both factors. The output is identical to that of Alg. 2.3 (`factor-metropolis`).

procedure `factor-metropolis(patch)`
input x
$\Delta \leftarrow \texttt{ran}(-\delta, \delta)$
$x' \leftarrow x + \Delta$
$\Upsilon_2 \leftarrow \texttt{ran}(0,1)\,;\ \Upsilon_4 \leftarrow \texttt{ran}(0,1)$
if $\Upsilon_2 < \min\left[1, \frac{\pi_2(x')}{\pi_2(x)}\right]$ **and** $\Upsilon_4 < \min\left[1, \frac{\pi_4(x')}{\pi_4(x)}\right]$ **:**
  $\{\ x \leftarrow x'$ (move accepted by consensus)
output x
——

Algorithm 2.4: `factor-metropolis(patch)`. Patch of Alg. 2.3, implementing the consensus principle (see Ref. [3]).

2.2.3 More reversible Markov-chain algorithms ...

2.3 Non-reversible Markov chains

In a tradition that started with the Metropolis algorithm many decades ago, Markov chains are normally designed with the restrictive detailed-balance condition, although they are only required to satisfy global balance. In this section, we illustrate more recent attempts to overcome the detailed-balance condition in a systematic way, within the framework of "lifted" Markov chains. Background and references can be found in [3].

2.3.1 Lifting and the zig-zag algorithm

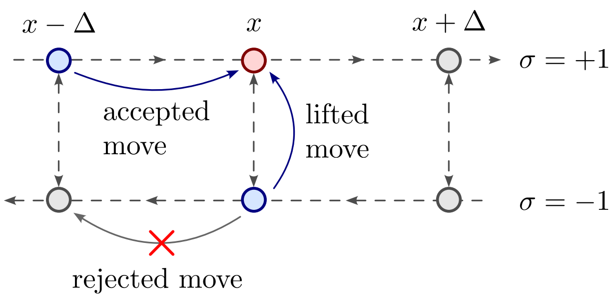

Figure 2.5: Discretized lifted Metropolis algorithm for the anharmonic oscillator. The flow into the lifted configuration $\{x, +1\}$ is indicated [see Eq. (2.18)].

The Metropolis algorithm proposes positive and negative displacements Δ for the anharmonic oscillator with symmetric *a priori* probabilities (see Alg. 2.2 (`metropolis`)). The filter then imposes that the net flow vanishes, so there will be as many particles going from x to $x+\Delta$ as in the reverse direction, even if, say, $\pi(x) \ll \pi(x+\Delta)$. To break detailed balance and only satisfy global balance, (while keeping π_{24} as a stationary distribution), we first suppose that the positions x lie on the grid $\{\dots, -2\Delta, -\Delta, 0, \Delta, 2\Delta, \dots\}$, with moves allowed only between nearest neighbors. Each configuration x is duplicated into a forward-moving one $\{x, +1\}$, and a

[2]one can view this as the sampling of two independent Boolean random variables, (see Ref. [6]), of which the final decision is the *conjunction*.

backward-moving one $\{x, -1\}$. From a lifted configuration $\{x, \sigma\}$, the lifted Metropolis algorithm proposes only a forward move if $\sigma = 1$, and only a backward move if $\sigma = -1$. In summary,

$$P^{\text{lift}}(\{x, \sigma\}, \{x + \sigma\Delta, \sigma\}) = \min\left[1, \frac{\pi_{24}(x + \sigma\Delta)}{\pi_{24}(x)}\right], \tag{2.15}$$

where $\sigma = \pm 1$. When this move is rejected by the Metropolis filter, the algorithm flips the direction and instead moves from $\{x, \sigma\}$ to $\{x, -\sigma\}$:

$$P^{\text{lift}}(\{x, \sigma\}, \{x, -\sigma\}) = 1 - \min\left[1, \frac{\pi_{24}(x + \sigma\Delta)}{\pi_{24}(x)}\right]. \tag{2.16}$$

This algorithm clearly violates detailed balance as there is thus no backward flow for $\sigma = +1$ and no forward flow for $\sigma = -1$. On the other hand, the lifted Metropolis algorithm satisfies the global-balance condition of Eq. (2.2) with the ansatz

$$\pi_{24}^{\text{lift}}(\{x, \sigma\}) = \frac{1}{2}\pi_{24}(x) \quad \text{for } \sigma = \pm 1. \tag{2.17}$$

For example, the flow into the lifted configuration $\{x, +1\}$ satisfies

$$\begin{aligned} &\pi_{24}(\{x, +1\}) \\ &= \pi_{24}(\{x - \Delta, +1\})P^{\text{lift}}(\{x - \Delta, +1\}, \{x, +1\}) \\ &+ \pi_{24}(\{x, -1\})P^{\text{lift}}(\{x, -1\}, \{x, +1\}). \end{aligned} \tag{2.18}$$

The two contributions on the right-hand side of Eq. (2.18) correspond on the one hand to the accepted moves from $\{x - \Delta, +1\}$, and on the other hand to the lifted moves from $\{x, -1\}$, when the move from $\{x, -1\}$ toward $\{x - \Delta, -1\}$ is rejected (see Fig. 2.5). Equation (2.18) can be transformed into

$$\begin{aligned} \pi_{24}(x) = \pi_{24}(x - \Delta) \min\left[1, \frac{\pi_{24}(x)}{\pi_{24}(x - \Delta)}\right] \\ + \pi_{24}(x)\left\{1 - \min\left[1, \frac{\pi_{24}(x - \Delta\sigma)}{\pi_{24}(x)}\right]\right\}, \end{aligned} \tag{2.19}$$

which is identically satisfied. We have shown that the lifted Metropolis algorithm satisfies the global-balance condition for the ansatz of Eq. (2.17), which splits $\pi_{24}(x)$ equally between $\{x, +1\}$ and $\{x, -1\}$. The sequence $\pi^{\{t\}}$ will actually converge to this stationary distribution.

In the lifted Metropolis algorithm, the particle, starting from $x_0 = 0$, climbs uphill in direction σ until a move is rejected by the filter, when it remains at its current position but reverses its velocity to $-\sigma$. The following downhill moves, again without rejections, are followed by another uphill climb, and so on, criss-crossing between the two wings of the potential U_{24}. It outputs configurations $\{x, \sigma\}$ such that, remarkably, the x-component samples π_{24}. This curious algorithm is implemented in Alg. 2.5 (`lifted-metropolis`), where we (almost) silently replaced the fixed grid of positions by a sampling of Δ.

2.3.2 Event-driven Markov processes

Markov chains in continuous time are called *Markov processes*. To approach these, we consider Algorithm 2.5 (`lifted-metropolis`) with a grid of positions $\{\dots, -2\Delta, -\Delta, 0, \Delta, 2\Delta, \dots\}$ and nearest-neighbor moves. In Alg. 2.5 (`lifted-metropolis`), we thus input a fixed Δ and scrap the line $\Delta \leftarrow \texttt{ran}(0, \delta)$, then study it in the limit of small Δ. We rescale time such that a displacement $\pm\Delta$ is itself undertaken in a time interval Δ. The particle in the anharmonic oscillator thus moves with unit absolute velocity, whose sense is reversed when there is a rejection.

procedure `lifted-metropolis`
input $\{x, \sigma\}$ (lifted sample at time t)
$\Delta \leftarrow \texttt{ran}(0, \delta)$ ($\delta > 0$)
$x' \leftarrow x + \sigma\Delta$ (x' in direction σ from x)
$\Upsilon \leftarrow \texttt{ran}(0, 1)$
if $\Upsilon < \min\left[1, \dfrac{\pi_{24}(x')}{\pi_{24}(x)}\right]$**:** $x \leftarrow x'$
else: $\sigma \leftarrow -\sigma$
output $\{x, \sigma\}$ (lifted sample at time $t+1$)
——

Algorithm 2.5: `lifted-metropolis`. Non-reversible lifted version of Alg. 2.2 (`metropolis`). The x-positions that are output by this program sample π_{24} (see Ref. [3]).

The downhill moves are all accepted, and even uphill moves are accepted with a probability close to one. We may simulate each of these steps, but it's preferable to *sample* the position of the next rejection. As an example, let us consider a sequence of uphill moves in positive direction from $x = 0$. The probability for accepting an entire sequence of n subsequent uphill moves, at positions $0, \Delta, \ldots, (n-1)\Delta$, and then rejecting the move $n+1$, is

$$\mathbb{P}(0 \to x_{\text{ev}}) = \underbrace{\mathrm{e}^{-\beta \Delta U_{24}(0) \cdots \Delta U_{24}[(n-1)\Delta]}}_{n \text{ accept}} \underbrace{\left[1 - \mathrm{e}^{-\beta \Delta U_{24}(n\Delta)}\right]}_{\text{reject, expand to 1st order}} \to \beta \mathrm{e}^{-\beta U_{24}} \mathrm{d}U_{24}. \tag{2.20}$$

In the small-Δ limit, the rejection is here expanded to first order, and ΔU is replaced by $\mathrm{d}U$. In our example of the anharmonic oscillator starting at $x = 0$, all the increments of ΔU_{24} up to position x add up to the potential $U_{24}(x)$. Equation (2.20) indicates that the value of U_{24} at which the velocity is reversed follows an exponential distribution in U_{24}. Remembering from Lecture 1 how to sample an exponential random variable, we obtain

$$U_{24}(x_{\text{ev}}) = -\beta^{-1} \log \texttt{ran}(0, 1)\,, \tag{2.21}$$

which can be inverted as $U_{24}(x_{\text{ev}}) = x_{\text{ev}}^2/2 + x_{\text{ev}}^4/4$, with

$$x_{\text{ev}} = \sigma \sqrt{-1 + \sqrt{1 - 4\beta^{-1} \log \texttt{ran}(0, 1)}}. \tag{2.22}$$

Algorithm 2.6 (`zig-zag`) implements this event-driven, continuous-time, Markov process and manages to move forward and backward. The equal-time samples again sample the Boltzmann distribution π_{24} (see Fig. 2.6).

procedure `zig-zag`
input $\{x, \sigma\}, t$ (lifted sample with $\sigma x \leq 0$)
$x_{\text{ev}} \leftarrow \sigma \sqrt{-1 + \sqrt{1 - 4\beta^{-1} \log \texttt{ran}(0, 1)}}$ (see Eq. (2.22))
$t_{\text{ev}} \leftarrow t + |x_{\text{ev}} - x|$
for $t^* = \text{int}(t) + 1, \ldots, \text{int}(t_{\text{ev}})$**:**
  $\{$ **print** $x + \sigma(t^* - t)$ (equal-time samples)
$x \leftarrow x_{\text{ev}};\ \sigma \leftarrow -\sigma;\ t \leftarrow t_{\text{ev}}$ ("zig-zag")
output $\{x, \sigma\}, t$
——

Algorithm 2.6: `zig-zag`. Continuous-time, event-driven version of Alg. 2.5 (`lifted-metropolis`).' The x-positions output by the `print` statement sample π_{24} (see Ref. [3]).

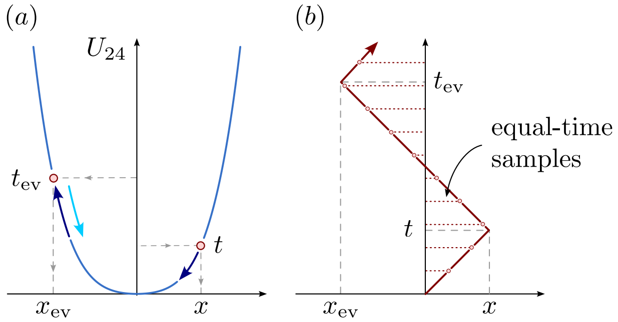

Figure 2.6: Zig-zag algorithm (continuous-time event-driven lifted Metropolis chain). (a): The particle swings about the origin, turning around at positions x_{ev} [sampled by Eq. (2.22)]. (b): Piecewise deterministic constant-velocity trajectory. Particle positions are sampled at equal time steps (see Ref. [3]).

2.3.3 More non-reversible Markov-chain algorithms...

There are many more ideas in Markov-chain Monte Carlo (for a starter, try Ref. [3]), even if we restrict ourselves, as we did here, to true samplers of π, without any correction, in the limit of large times. From our modest beginnings, we already see a world of infinite possibilities.

References

[1] L. Wasserman, *All of Statistics.* New York: Springer, 2004.

[2] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov Chains and Mixing Times.* American Mathematical Society, 2008.

[3] G. Tartero and W. Krauth, "Concepts in Monte Carlo sampling," *American Journal of Physics*, vol. 92, no. 1, p. 65–77, 2024.

[4] W. Krauth, *Statistical Mechanics: Algorithms and Computations.* Oxford University Press, 2006.

[5] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," *J. Chem. Phys.*, vol. 21, pp. 1087–1092, 1953.

[6] W. Krauth, "Event-Chain Monte Carlo: Foundations, Applications, and Prospects," *Front. Phys.*, vol. 9, p. 229, 2021.