The Computational Physics Section publishes articles that help students and instructors learn about the computational tools used in contemporary research. Interested authors are encouraged to send a proposal to the editors of the Section, Jan Tobochnik (jant@kzoo.edu) or Harvey Gould (hgould@clarku.edu). Summarize the physics and the algorithm you wish to discuss and how the material would be accessible to advanced undergraduates or beginning graduate students.

Concepts in Monte Carlo sampling

Gabriele Tartero[a] and Werner Krauth[b]
Laboratoire de Physique de l'Ecole Normale Supérieure, ENS, Université PSL, CNRS, Sorbonne Université, Université Paris-Diderot, Sorbonne Paris Cité, 24 rue Lhomond, F-75005 Paris, France

(Received 17 September 2023; accepted 21 September 2023)

We discuss contemporary ideas in Monte Carlo algorithms in the simplified setting of the one-dimensional anharmonic oscillator. After reviewing the connection between molecular dynamics and Monte Carlo, we introduce the Metropolis and the factorized Metropolis algorithms and lifted non-reversible Markov chains. We, furthermore, illustrate the concept of thinning, where moves are accepted by simple bounding potentials rather than the harmonic and quartic contributions to the anharmonic oscillator. We point out the multiple connections of our example algorithms with real-world sampling problems. This paper is self-contained, and Python implementations are provided.
https://doi.org/10.1119/5.0176853

I. INTRODUCTION

The Monte Carlo method is an important tool for producing samples x from a given probability distribution $\pi(x)$. In real-life applications, algorithms and computer implementations for this sampling problem can be highly complex. In contrast, we discuss a dozen distinct Monte Carlo algorithms in the severely stripped-down setting of a particle in a one-dimensional anharmonic potential

$$U_{24}(x) = \frac{x^2}{2} + \frac{x^4}{4}, \tag{1}$$

consisting of a harmonic term, $U_2 = x^2/2$, and a quartic one, $U_4 = x^4/4$. For concreteness, we also provide short example programs.

For the anharmonic oscillator, the distribution to be sampled is the Boltzmann distribution,

$$\pi_{24}(x) = \exp\left[-\beta U_{24}(x)\right], \tag{2}$$

where $\beta = (k_B T)^{-1}$ is the inverse of the temperature T and k_B denotes the Boltzmann constant. The connection between the potential U_{24} and the distribution π_{24} derives from the following. In classical mechanics, an isolated particle is governed by Newton's law and, in a one-dimensional confining potential, oscillates between two turning points. A certain function $\pi^{\text{iso}}(x)$ describes the fraction of time that the particle spends at position x during one period and, therefore, during a long time interval containing many periods. If the particle is in contact with a thermostat, this function turns into a probability distribution for finding the particle at a position x at large times t, and it is exactly the Boltzmann distribution $\pi_{24}(x)$ of Eq. (2), as we will discuss in Sec. II. The molecular dynamics method accesses this distribution through the numerical solution of Newton's equation in contact with a thermostat.

The Monte Carlo method more abstractly addresses the sampling problem than that of molecular dynamics, as it samples (obtains samples x from) the distribution $\pi_{24}(x)$ without simulating a physical process. The sequence of 12 algorithms that we present here will lead us from the beginning of the method, namely, direct sampling and the reversible Metropolis algorithm and its extensions (Sec. III), to non-reversible Markov-chain algorithms (Sec. IV) and to advanced approaches that sample the target distribution with a minimum of evaluations of the potential (Sec. V). Some mathematical details are collected separately in Appendix A. Our algorithms are presented in compact pseudo-code (as in Ref. 1) and implemented in short, freely accessible, Python programs (Appendix B). Their correctness is tested to high precision (Appendix C). A companion paper[2] will apply the concepts discussed here to real-life settings and address efficiency questions, whereas in this paper, we are concerned only with the correctness of the sampling algorithms.

II. FROM CLASSICAL TO STATISTICAL MECHANICS

If isolated from the environment, so that the energy is conserved, the anharmonic oscillator of Fig. 1 is a classical, periodic, one-dimensional deterministic system, and we may track the fraction of time per period that the particle spends near a given position x (Sec. II A). When interacting with a heat bath (for which we use a concrete realization), the motion is piecewise deterministic,[3] yet random. In this case, we may sample the Boltzmann distribution π_{24} by molecular dynamics modeling of the particle subject to Newton's laws and interacting with the thermostat (Sec. II B). We will then

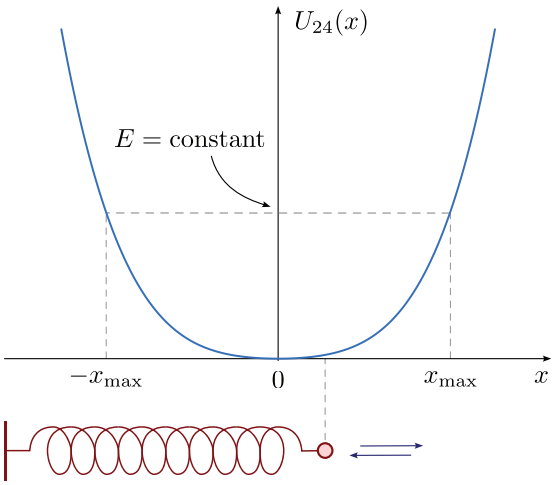


Fig. 1. Isolated anharmonic oscillator at energy E, subject to the potential U_{24} of Eq. (1).

provide a Monte Carlo algorithm that directly samples x from the Boltzmann distribution (Sec. II C).

A. The isolated anharmonic oscillator

We may hold the particle fixed—with initial velocity $v=0$—and then release it at time $t=0$ from a position $x_{\max}>0$. If it is isolated, the anharmonic oscillator conserves its energy E, given by the sum of the kinetic and potential energies at all times $t\geq 0$. It, thus, picks up velocity until it reaches the minimum of the potential at $x=0$, then slows down and turns around at $-x_{\max}$, where E equals the potential energy and the velocity again vanishes (see Fig. 2(a)). The energy E is then

$$E=\frac{x^2{}_{\max}}{2}+\frac{x^4{}_{\max}}{4}\iff x_{\max}=\sqrt{-1+\sqrt{1+4E}}, \tag{3}$$

as follows from solving a quadratic equation and taking a square root. In between the turning points $-x_{\max}$ and $x_{\max}$, the kinetic energy (with the mass equal to unity) is $\frac{1}{2}(\mathrm{d}x/\mathrm{d}t)^2$, and conservation of energy can be written as

$$E=\frac{1}{2}\left(\frac{\mathrm{d}x}{\mathrm{d}t}\right)^2+U_{24}(x)\iff \frac{\mathrm{d}x}{\mathrm{d}t}=\pm\sqrt{2[E-U_{24}(x)]}, \tag{4}$$

which gives

$$\mathrm{d}t=\pm\sqrt{\frac{1}{2[E-U_{24}(x)]}}\mathrm{d}x. \tag{5}$$

The period τ of the motion, i.e., the time between two realizations of a given position and velocity, corresponds to four times the interval from $x=0$ to $x_{\max}$,

$$\begin{aligned}\tau&=4\int_0^{x_{\max}}\mathrm{d}t=4\int_0^{\sqrt{-1+\sqrt{1+4E}}}\frac{1}{\sqrt{2[E-U_{24}(x)]}}\mathrm{d}x\\&=4\sqrt{\frac{2}{1+\sqrt{1+4E}}}K\left(\frac{1-\sqrt{1+4E}}{1+\sqrt{1+4E}}\right),\end{aligned} \tag{6}$$

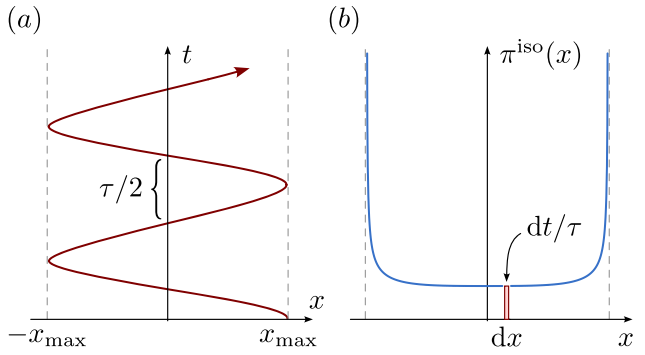


Fig. 2. Isolated anharmonic oscillator, as represented in Fig. 1. (a) Periodic trajectory with amplitude $2x_{\max}$ and period τ. (b) Normalized function π^{iso}. The fraction of time $\mathrm{d}t/\tau$ spent per period between x and $x+\mathrm{d}x$ is $\pi^{\mathrm{iso}}(x)\mathrm{d}x$.

where K is the complete elliptic integral of the first kind (see Fig. 3). For small E, the period τ agrees with that of the harmonic oscillator, which is independent of $x_{\max}$ and, thus, of E. For large E, in contrast, the period $\tau\sim E^{-1/4}$ approaches that of the quartic oscillator (see Appendix A for some mathematical details).

Equation (5) yields the fraction $\pi^{\mathrm{iso}}(x)$ of time that the particle spends between x and $x+\mathrm{d}x$ over a semi-period,

$$\pi^{\mathrm{iso}}(x)=\frac{2}{\tau}\sqrt{\frac{1}{2[E-U_{24}(x)]}}, \tag{7}$$

with $-x_{\max}<x<x_{\max}$. The function $\pi^{\mathrm{iso}}(x)$ is normalized but does not represent the probability for the particle to be at x at a fixed time t, because of the deterministic nature of the motion (see Fig. 2(b)).

To simulate the isolated anharmonic oscillator, we could numerically integrate the first-order ordinary differential equation on the right of Eq. (4) over a quarter period and then piece together the entire trajectory of Fig. 2(a). However, this method is specific to one-dimensional dynamical systems (Ref. 4, Sec. 11). To reflect the general case, we numerically integrate Newton's law for the force F,

$$F=m\frac{\mathrm{d}^2x(t)}{\mathrm{d}t^2},\quad \text{with } F=-\frac{\mathrm{d}U_{24}}{\mathrm{d}x}=-x-x^3. \tag{8}$$

By substituting the time differential $\mathrm{d}t$ by a very small finite interval Δt, appropriate for stepping from t to $t+\Delta t$, and to $t+2\Delta t$, and so on, we obtain

$$x(t+\Delta t)=x(t)+v(t)\Delta t, \tag{9}$$

$$v(t+\Delta t)=v(t)-(x+x^3)\Delta t. \tag{10}$$

Algorithm 0 (`isolated-dynamics`) implements one iteration of this naive algorithm, for which we start with an initial position $x(t=0)=x_{\max}$ and an initial velocity $v(t=0)=0$. The output can then be fed back into the input of the program. Like most simple isolated molecular dynamics codes, Alg. 0 is unstable—the energy will slowly increase with time and then diverge. To obtain good approximate results, we should use a small discretization Δt and not run the program for excessively large values of t.

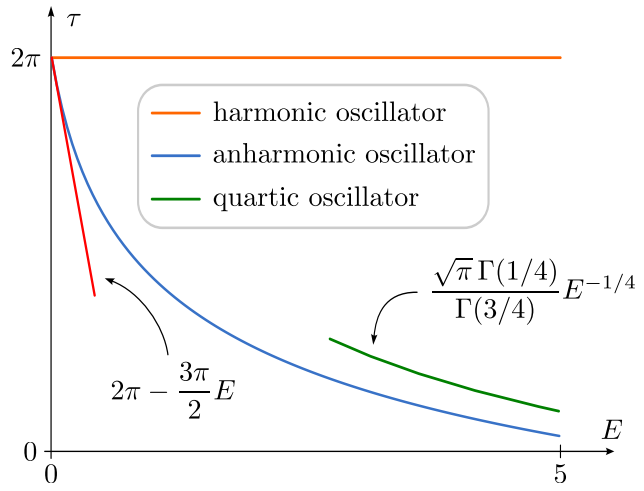


Fig. 3. Period τ of the isolated anharmonic oscillator as a function of the energy E. The period of the harmonic oscillator is independent of E, while that of the quartic oscillator scales as $E^{-1/4}$. Here, Γ denotes the Euler gamma function (see Appendix A).

Algorithm 0. `isolated-dynamics`. Naive integration of Newton's equations for the isolated anharmonic oscillator.

procedure `isolated-dynamics`
input x, v, t
$t \leftarrow t + \Delta t$
$x' \leftarrow x + v\Delta t$
$v \leftarrow v - (x + x^3)\Delta t$
$x \leftarrow x'$
output x, v, t

B. Introducing a thermal bath

Liquids, gases, and other systems described by statistical mechanics are generally composed of particles that interact and exchange energy and momentum. All systems interact with their environment and therefore do not conserve energy and momentum. For the anharmonic oscillator, this interaction may be modeled by an external heat bath at temperature T, represented by a box composed of a large number of hard spheres of mass $m = 1$ that move randomly with velocities given by the Maxwell distribution. For concreteness, we imagine the anharmonic oscillator to be in contact with the heat bath through a semi-permeable elastic thermostat, a stick that vibrates back and forth in an infinitesimal interval around $x = 0$, and that is also of mass one. At each collision of the thermostat with a heat bath particle, their two velocities are exchanged. We may imagine that the anharmonic oscillator, as it approaches $x = 0$, passes through the thermostat without interaction with probability 1/2, and otherwise bounces off with the velocity of the stick. The particle trajectory is then deterministic except at the origin (see Fig. 4). Statistical mechanics teaches us that, although all the particles in the heat bath are Maxwell-distributed, the thermostat behaves differently. In particular, because the latter lies at a fixed position (up to an infinitesimal interval), its velocity follows the distribution

$$\pi(v)\mathrm{d}v = \beta|v|\mathrm{e}^{-\beta v^2/2}\mathrm{d}v, \tag{11}$$

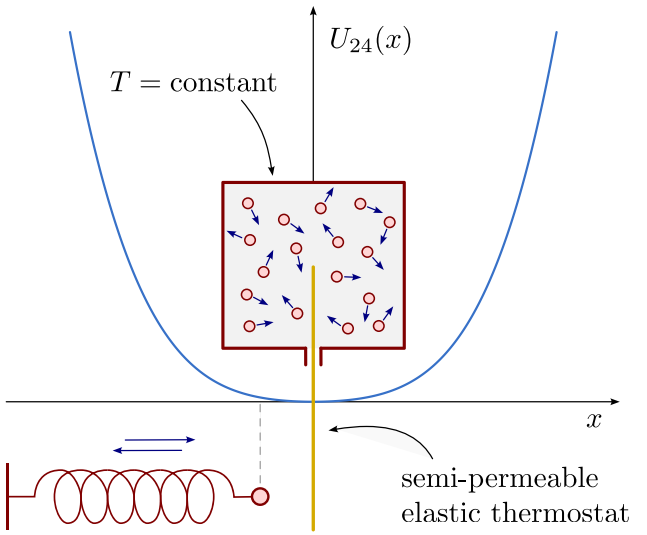


Fig. 4. Anharmonic oscillator of Eq. (1) interacting with a heat bath at temperature T through an elastic semi-permeable thermostat vibrating in an infinitesimal interval about $x = 0$.

often called the Maxwell boundary condition (see Ref. 1, Sec. 2.3.1). It differs by the prefactor $\beta|v|$ from the Maxwell distribution of one velocity component.

The velocity distribution of the thermostat in Eq. (11) can be sampled as

$$v = \pm\sqrt{\frac{-2\log \mathtt{ran}(0,1)}{\beta}}, \tag{12}$$

and the Maxwell boundary condition, realized with a single random number, exactly represents the infinite heat bath. After a few collisions (see Fig. 5(a)), the particle has forgotten its initial position $x(0)$, and it makes sense to speak of the probability distribution at time t. The latter is exactly given by $\pi_{24}(x)$ in the limit $\Delta t \to 0$, and substantially differs from π^{iso} of Fig. 2(b). It is naively sampled by Alg. 1 (`thermostat-dynamics`).

Algorithm 1. `thermostat-dynamics`. Naive solution of Newton's equations for the anharmonic oscillator with the semi-permeable thermostat at $x = 0$ (see Fig. 4).

procedure `thermostat-dynamics`
input x, v, t
$x' \leftarrow x + v\Delta t$
$t \leftarrow t + \Delta t$
$\Upsilon \leftarrow \mathtt{ran}(0,1)$
if $x \cdot x' < 0$ **and** $\Upsilon < 1/2$:
  $v \leftarrow -\mathtt{sign}(v)\sqrt{-2\beta^{-1}\log \mathtt{ran}(0,1)}$ (see Eq. (12))
else :
  $v \leftarrow v - (x + x^3)\Delta t$
  $x \leftarrow x'$
output x, v, t

We pause for a moment to compute the normalization $Z(\beta)$ of π_{24} in Eq. (2), that is, the partition function

$$Z(\beta) = \int_{-\infty}^{\infty} \mathrm{d}x\, \pi_{24}(x) = \frac{\mathrm{e}^{\beta/8}}{\sqrt{2}} K_{1/4}(\beta/8), \tag{13}$$

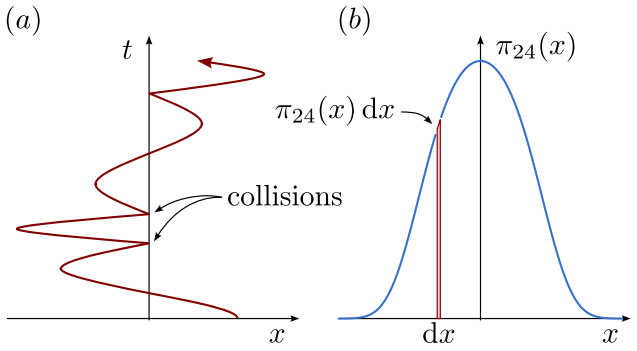


Fig. 5. Anharmonic oscillator in contact with the thermostat of Fig. 4. (a) Piecewise deterministic trajectory with random kicks at $x = 0$. (b) At large t, when the initial configuration $x(t = 0)$ is forgotten, the particle positions follow the Boltzmann distribution π_{24} of Eq. (2).

where $K_{1/4}$ denotes the Bessel function of the second kind (see Appendix A). For simplicity of notation, the division by the partition function is understood whenever we want π_{24} to represent a bona fide normalized probability distribution.

C. Direct Monte Carlo sampling

To sample the distribution π_{24}, we need not simulate a physical system—in our case, the anharmonic oscillator in contact with a heat bath. Let us first consider the simpler problem of the Gaussian distribution,

$$\pi_2(x) = \exp\left[-\beta U_2(x)\right] = \exp\left(-\beta x^2/2\right). \tag{14}$$

Samples x of $\pi_2(x)$ are known as Gaussian random numbers of zero mean and of standard deviation $1/\sqrt{\beta}$. They are readily available on computers, websites, and even pocket calculators (see Ref. 1, Sec. 1.2.5, for an algorithm using the method of sample transformation from uniform random numbers). With an additional uniform random number $y = \texttt{ran}[0, \exp(-\beta x^2/2)]$, they can be expanded into two-dimensional positions (x, y) of "pebbles," which are uniformly distributed in the area between the x-axis and the bell-shaped Gaussian curve of Eq. (14).

The probability distribution of the anharmonic oscillator satisfies $\pi_{24}(x) \le \pi_2(x)$ for all x (see Fig. 6), and those pebbles that lie below π_{24}—as they are uniformly distributed below the Gaussian curve—are also evenly spread out below π_{24}. As a consequence, the function π_{24} need not be normalized.

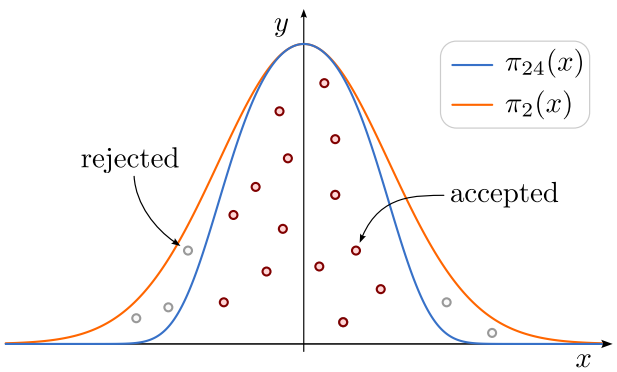


Fig. 6. Uniformly distributed pebbles below the Gaussian curve π_2. The x-values of the pebbles (x, y), with $y < \pi_{24}(x)$, sample the Boltzmann distribution π_{24} of Eq. (2).

Clearly, it suffices to reject any pebble (x, y) above π_{24} to be left with x positions distributed according to the Boltzmann distribution of the anharmonic oscillator. Algorithm 2 (`direct-sampling`) implements this direct-sampling idea.

Algorithm 2. `direct-sampling`. Sampling π_{24} through the rejection of Gaussians samples from Eq. (14).

procedure `direct-sampling`
while True :
$x \leftarrow \texttt{gauss}(0, 1/\sqrt{\beta})$
$y \leftarrow \texttt{ran}[0, \pi_2(x)]$
if $y < \pi_{24}(x)$: **break**
output x

III. REVERSIBLE MARKOV CHAINS

The probability of rejecting a pebble in Alg. 2 (`direct-sampling`) is not too high, and a sample of π_{24} can be quickly obtained from a sample of π_2. In real life, however, the difference between any distribution that we can sample (as π_2) and the one we want to sample (as π_{24}) becomes huge, thwarting the direct-sampling approach. In the alternative Markov-chain sampling approach, we start at time $t = 0$ with a sample x_0 from a distribution $\pi^{\{0\}}$ that we know how to sample. At the next step, the position x_1 samples a distribution $\pi^{\{1\}}$, and so on. If we introduce the transition matrix P such that $P(x', x)$ represents the probability to move from a sample x' to a sample x in one time step, the distribution at time $t + 1$ can be expressed as

$$\pi^{\{t+1\}}(x) = \sum_{x' \in \Omega} \pi^{\{t\}}(x') P(x', x) \quad \forall x \in \Omega, \tag{15}$$

where the sample space Ω represents the set of all configurations of the system. Markov-chain Monte Carlo requires that, at large t, x_t samples the distribution $\pi^{\{t\to\infty\}} = \pi$. For the sampling to take place, the transition matrix P must satisfy, for all $x \in \Omega$, the global-balance condition,

$$\pi(x) = \sum_{x' \in \Omega} \pi(x') P(x', x) \quad \text{(global balance)}, \tag{16}$$

which is nothing but the steady-state version of Eq. (15). The strategy for sampling π implied in Eqs. (15) and (16) represents a monumental investment, because we have to wait a long time until $\pi^{\{t\}} \sim \pi$ in order to obtain a single sample of π. It is not uncommon for this mixing time to correspond to weeks or even years of computer time.[5]

The algorithms in this section are more restrictive than required by Eq. (16). They satisfy, for all $x, x' \in \Omega$, the detailed-balance condition

$$\pi(x) P(x, x') = \pi(x') P(x', x) \quad \text{(detailed balance)}. \tag{17}$$

It suffices to sum Eq. (17) over all $x' \in \Omega$ (using the conservation of probabilities $\sum_{x'} P(x, x') = 1$) to see that detailed balance implies global balance.

Detailed-balance algorithms are time-reversible. This means that, at large t (in equilibrium), any segment of the chain (for example $[a \to b \to c]$ in Fig. 7) at subsequent time steps is sampled with the same probability $\mathbb{P}$ as the time-reversed segment. In our example, $\mathbb{P}(a \to b \to c)$ is pieced together from the probability $\pi(a)$ to sample a and the transition-matrix probabilities to move from a to b and then from b to c, so that

$$\begin{aligned}\mathbb{P}(a \to b \to c) &= \underbrace{\pi(a)P(a,b)}_{\pi(b)P(b,a)\text{ etc.}} P(b,c)\\ &= \pi(c)P(c,b)P(b,a)\\ &= \mathbb{P}(c \to b \to a),\end{aligned} \tag{18}$$

where we have twice used the detailed-balance condition. By construction, reversible algorithms thus have no net flows (the flow $a \to b \to c$ is cancelled by the flow $c \to b \to a$), which points to a very serious restriction imposed by the detailed-balance condition: they can usually only move around Ω diffusively, that is, slowly.

In the following, we will first discuss the seminal reversible algorithm due to Metropolis *et al*. (Sec. III A). We will then explore a variant of the Metropolis algorithm which introduces a crucial factorization (Sec. III B). We finally discuss the consensus principle at the origin of modern developments (Sec. III C).

A. The Metropolis chain

To sample the distribution π_{24} with a reversible transition matrix $P(x,x')$, we impose the detailed-balance condition $\pi(x)P(x,x') = \pi(x')P(x',x)$ for any pair x and x' in Ω. To this end, we may choose

$$\pi(x)P(x,x') \propto \min\left[\pi(x), \pi(x')\right] \quad \text{for } x \neq x'. \tag{19}$$

The right-hand side of Eq. (19) 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,

$$P^{\text{Met}}(x,x') \propto \min\left[1, \frac{\pi(x')}{\pi(x)}\right] \quad \text{for } x \neq x'. \tag{20}$$

Let us discuss the difference between a transition matrix and a filter in order to make Eq. (20) 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 a symmetric *a priori* probability $\mathcal{A}(x,x')$ and is then accepted or rejected with a filter

$$\underbrace{P^{\text{Met}}(x,x')}_{\text{transition matrix}} = \underbrace{\mathcal{A}(x,x')}_{\textit{a priori}\text{ probability}} \overbrace{\mathcal{P}^{\text{Met}}(x,x')}^{\text{Metropolis filter}}. \tag{21}$$

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

$$\mathcal{P}^{\text{Met}}(x,x') = \min\left[1, \frac{\pi(x')}{\pi(x)}\right]. \tag{22}$$

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 $\sum_{x'} P(x,x') = 1$.

Algorithm 3 (`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.

Algorithm 3. `metropolis`. Sampling π_{24} with the Metropolis algorithm.

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 $\Upsilon < \min\left[1, \frac{\pi_{24}(x')}{\pi_{24}(x)}\right]$: $x \leftarrow x'$
output x (sample at time $t+1$)

B. Factorizing the Metropolis filter

The Metropolis algorithm is famous, but it is not the end of history. A modern variant is useful for distributions π that factorize

$$\pi = \pi_a \pi_b \pi_c \cdots \pi_k = \prod_{\xi=a,\ldots,k} \pi_\xi. \tag{23}$$

For example, the Boltzmann distribution $\pi = \exp(-\beta U)$ takes the above form if its potential U can be written as the sum over pair potentials. The Metropolis filter of Eq. (22) is then

$$\begin{aligned}\mathcal{P}^{\text{Met}}(x,x') &= \min\left[1, \frac{\pi_a(x')\pi_b(x')\cdots\pi_k(x')}{\pi_a(x)\pi_b(x)\cdots\pi_k(x)}\right]\\ &= \min\left[1, \prod_{\xi=a,\ldots,k} \frac{\pi_\xi(x')}{\pi_\xi(x)}\right],\end{aligned} \tag{24}$$

and it is implemented in this way in countless computer programs. An alternative to Eq. (24) is the factorized Metropolis filter,[6]

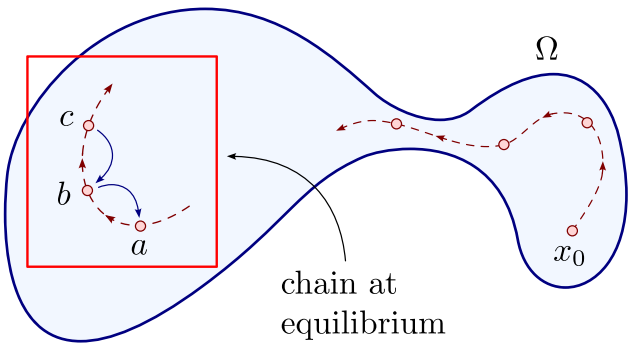


Fig. 7. Reversible Markov chain. In equilibrium, it satisfies $\mathbb{P}(a \to b \to c) = \mathbb{P}(c \to b \to a)$ for all $a, b, c \in \Omega$.

$$\mathcal{P}^{\text{fact}}(x,x') = \min\left[1,\frac{\pi_a(x')}{\pi_a(x)}\right]\cdots\min\left[1,\frac{\pi_k(x')}{\pi_k(x)}\right] = \prod_{\xi=a,\ldots,k}\min\left[1,\frac{\pi_\xi(x')}{\pi_\xi(x)}\right]. \tag{25}$$

If used naively, it gives lower acceptance probabilities than the Metropolis filter, but it also satisfies the detailed-balance condition. Let us prove this for the anharmonic oscillator, where

$$\mathcal{P}_{24}^{\text{fact}}(x,x') = \min\left[1,\frac{\pi_2(x')}{\pi_2(x)}\right]\min\left[1,\frac{\pi_4(x')}{\pi_4(x)}\right] \tag{26}$$

and

$$\pi_{24}(x) = \exp\left(-\frac{x^2}{2}-\frac{x^4}{4}\right) = \exp\left(-\frac{x^2}{2}\right)\exp\left(-\frac{x^4}{4}\right) = \pi_2(x)\pi_4(x), \tag{27}$$

illustrating that a potential that is a sum of terms yields a Boltzmann distribution that factorizes. Detailed balance is satisfied because of the following:

$$\begin{aligned}\pi_{24}(x)P_{24}^{\text{fact}}(x,x') &\propto \underbrace{\pi_2(x)\min\left[1,\frac{\pi_2(x')}{\pi_2(x)}\right]}_{\min[\pi_2(x),\pi_2(x')]:\,x\Longleftrightarrow x'}\underbrace{\pi_4(x)\min\left[1,\frac{\pi_4(x')}{\pi_4(x)}\right]}_{\min[\pi_4(x),\pi_4(x')]:\,x\Longleftrightarrow x'}\\ &\propto \pi_{24}(x')P_{24}^{\text{fact}}(x',x),\end{aligned} \tag{28}$$

where we have dropped the symmetric *a priori* probability $\mathcal{A}$. Algorithm 4 (`factor-metropolis`) samples π_{24}. It implements the factorized filter in a way that we will soon discover to be naive.

Algorithm 4. `factor-metropolis`. Sampling π_{24} naively with the factorized Metropolis filter.

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

C. The consensus principle

The factorized Metropolis algorithm will turn out to be particularly powerful in the presence of many factors, even an infinite number of them. This power is due to the consensus principle, which we now discuss, and which will avoid the evaluation of the lengthy product in Eq. (25).

For the anharmonic oscillator, the consensus principle simply relies on the fact that the filter

$$\mathcal{P}_{24}^{\text{fact}}(x,x') = \underbrace{\min\left[1,\frac{\pi_2(x')}{\pi_2(x)}\right]}_{p_2\ \text{(in Table I)}}\underbrace{\min\left[1,\frac{\pi_4(x')}{\pi_4(x)}\right]}_{p_4\ \text{(in Table I)}} \tag{29}$$

Table I. Consensus principle for the factorized Metropolis filter. A move $x \to x'$ that is accepted/rejected independently by the harmonic and the quartic factor with probabilities taken from Eq. (29). The acceptance "by consensus" reproduces the correct probability of Eq. (26).

Harmonic \ Quartic	Accept (p_4)	Reject ($1-p_4$)
Accept (p_2)	p_2p_4 ✓	$p_2(1-p_4)$
Reject ($1-p_2$)	$(1-p_2)p_4$	$(1-p_2)(1-p_4)$

is a product p_2p_4 of probabilities that may be interpreted as independent (see Table I). This independence holds even though the two factors are evidently correlated; for example, π_2 is small when π_4 is. In Alg. 5 (`factor-metropolis(patch)`), two independent decisions are taken, 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. 4 (`factor-metropolis`), and it again samples the Boltzmann distribution π_{24} and, thus, illustrates that the consensus principle is correct.

Algorithm 5. `factor-metropolis(patch)`. Patch of Alg. 4, implementing the consensus principle.

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

IV. GOING BEYOND REVERSIBILITY

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.[7,8] Our first lifted Markov chain, Alg. 6 (`lifted-metropolis`), takes fewer than a dozen lines of code, but is quite intricate (Sec. IV A). In recent applications, lifted Markov chains are often formulated for continuous time. For the anharmonic oscillator, this formulation yields the "zig-zag" algorithm,[9] where the particle moves back and forth as in molecular dynamics (as in Alg. 0 (`isolated-dynamics`)), but at a fixed velocity. Newton's equations are not solved, but π_{24} is still sampled exactly, and quite magically (Sec. IV B). The decision to reverse the velocity (from "zig" to "zag") may again be broken up into independent decisions of the harmonic and the quartic factors foreshadowing strategies that have profoundly impacted real-life sampling approaches (Sec. IV C).

A. Lifting the Metropolis chain

The Metropolis algorithm proposes positive and negative displacements Δ for the anharmonic oscillator with

symmetric *a priori* probabilities (see Alg. 3 (`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, $\pi(x) \ll \pi(x+\Delta)$.

We will now break detailed balance with a non-reversible "lifted" Markov chain[7,8] that only respects global balance, while having π_{24} as its stationary distribution. Let us suppose as a first step, that the positions x lie on the grid $\{\ldots, -2\Delta, -\Delta, 0, \Delta, 2\Delta, \ldots\}$, with moves allowed only between nearest neighbors. Each configuration x is duplicated ("lifted") into two configurations, a forward-moving one $\{x,+1\}$, and a 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,

$$P^{\text{lift}}(\{x,\sigma\},\{x+\sigma\Delta,\sigma\}) = \min\left[1, \frac{\pi_{24}(x+\sigma\Delta)}{\pi_{24}(x)}\right], \quad (30)$$

where $\sigma = \pm 1$. When this move is not accepted by the Metropolis filter, the algorithm flips the direction and instead moves from $\{x,\sigma\}$ to $\{x,-\sigma\}$ with probability

$$P^{\text{lift}}(\{x,\sigma\},\{x,-\sigma\}) = 1 - \min\left[1, \frac{\pi_{24}(x+\sigma\Delta)}{\pi_{24}(x)}\right]. \quad (31)$$

This algorithm clearly violates detailed balance as, for example,

$$P^{\text{lift}}(\{x,+1\},\{x+\Delta,+1\}) > 0, \quad (32)$$

$$P^{\text{lift}}(\{x+\Delta,+1\},\{x,+1\}) = 0. \quad (33)$$

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. (16) with the ansatz

$$\pi_{24}^{\text{lift}}(\{x,\sigma\}) = \frac{1}{2}\pi_{24}(x) \quad \text{for } \sigma = \pm 1. \quad (34)$$

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

$$\begin{aligned}\pi_{24}(\{x,+1\}) = {} & \pi_{24}(\{x-\Delta,+1\}) \\ & \times P^{\text{lift}}(\{x-\Delta,+1\},\{x,+1\}) \\ & + \pi_{24}(\{x,-1\})P^{\text{lift}}(\{x,-1\},\{x,+1\}).\end{aligned} \quad (35)$$

The two contributions on the right-hand side of Eq. (35) 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. 8). Equation (35) can be transformed into

$$\begin{aligned}\pi_{24}(x) = {} & \pi_{24}(x-\Delta)\min\left[1, \frac{\pi_{24}(x)}{\pi_{24}(x-\Delta)}\right] \\ & + \pi_{24}(x)\left\{1 - \min\left[1, \frac{\pi_{24}(x-\Delta\sigma)}{\pi_{24}(x)}\right]\right\},\end{aligned} \quad (36)$$

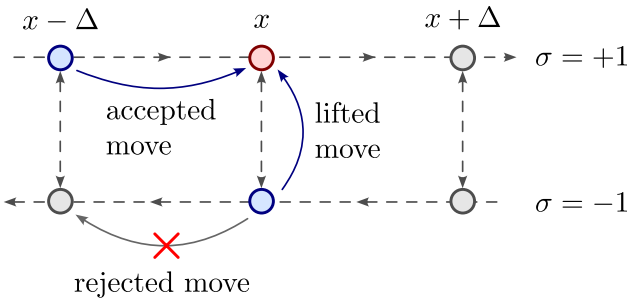


Fig. 8. Discretized lifted Metropolis algorithm for the anharmonic oscillator. The flow into the lifted configuration $\{x,+1\}$ is indicated [see Eq. (35)].

which is identically satisfied. We have shown that the lifted Metropolis algorithm satisfies the global-balance condition for the ansatz of Eq. (34), which splits $\pi_{24}(x)$ equally between $\{x,+1\}$ and $\{x,-1\}$. The sequence $\pi^{\{t\}}$ will actually converge to this stationary distribution under very mild conditions that are satisfied for the anharmonic oscillator.[10,11]

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}. Algorithm 6 (`lifted-metropolis`) implements a version of the lifted Metropolis algorithm where the displacements Δ are sampled from a positive interval. This lifted algorithm outputs configurations $\{x,\sigma\}$ such that, remarkably, the x-component samples π_{24}. In real-life applications, lifted Markov chains can often overcome the slow diffusive behavior of the reversible methods.

Algorithm 6. `lifted-metropolis`. Non-reversible lifted version of Alg. 3 (`metropolis`). The x-positions that are output by this program sample π_{24}.

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 $\Upsilon < \min\left[1, \frac{\pi_{24}(x')}{\pi_{24}(x)}\right]$: $x \leftarrow x'$
else: $\sigma \leftarrow -\sigma$
output $\{x,\sigma\}$ (lifted sample at time $t+1$)

B. From discrete to continuous time

So far, we have discussed Markov chains that move between configurations indexed by an integer time t, from x_t to x_{t+1}. We now consider algorithms in continuous time (technically speaking, we consider Markov processes). For simplicity, we revisit the lifted Metropolis algorithm with its grid of positions $\{\ldots, -2\Delta, -\Delta, 0, \Delta, 2\Delta, \ldots\}$ and with its nearest-neighbor moves, but consider the case of small Δ. It is then appropriate to 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.

The downhill moves are all accepted, and even uphill moves are accepted with a probability close to one. We may sample the position of the next rejection, rather than running through the sequence of individual moves, because an uphill move starting, say, in the positive direction from $x=0$ is accepted with probability $\exp[-\beta\Delta U_{24}(x=0)]$. Likewise, the probability for accepting an entire sequence of n uphill moves, at subsequent positions $0, \Delta, \ldots, (n-1)\Delta$, and then rejecting the move $n+1$, is

$$\mathbb{P}(0 \to x_{\rm ev}) = \underbrace{\mathrm{e}^{-\beta\Delta U_{24}(0)\cdots\beta\Delta U_{24}[(n-1)\Delta]}}_{n \text{ accepted moves}} \underbrace{[1-\mathrm{e}^{-\beta\Delta U_{24}(n\Delta)}]}_{\text{rejection}} \to \beta \mathrm{e}^{-\beta U_{24}} \mathrm{d}U_{24}. \tag{37}$$

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 (37) indicates that the value of U_{24} at which the velocity is reversed follows an exponential distribution in U_{24}.[12] Because an exponential random number can be obtained as a logarithm of a uniform random number (see Ref. 1, Sec. 1.2.4), we obtain

$$U_{24}(x_{\rm ev}) = -\beta^{-1} \log \texttt{ran}(0,1). \tag{38}$$

We invert $U_{24}(x_{\rm ev}) = x_{\rm ev}^2/2 + x_{\rm ev}^4/4$ and obtain

$$x_{\rm ev} = \sigma\sqrt{-1+\sqrt{1-4\beta^{-1}\log \texttt{ran}(0,1)}}. \tag{39}$$

To sample the Boltzmann distribution π_{24}, it now suffices to sample the turning points $x_{\rm ev}$ of the constant-velocity motion, alternatingly on the negative and positive branches of the potential, and then to sample the particle positions at equal time steps, as implemented in Alg. 7 (`zig-zag`). This event-driven continuous-time algorithm samples the Boltzmann distribution π_{24} (see Fig. 9). The event-driven version of Alg. 6 exists also for fixed, finite Δ, and it is often classified as "faster-than-the-clock" (see Ref. 1, Sec. 7.1.1).

Algorithm 7. `zig-zag`. Continuous-time version of Alg. 6 (`lifted-metropolis`) using an event-driven formulation. The x-positions output by the `print` statement sample π_{24}.

procedure `zig-zag`
input $\{x,\sigma\}, t$ (lifted sample with $\sigma x \le 0$)
$x_{\rm ev} \leftarrow \sigma\sqrt{-1+\sqrt{1-4\beta^{-1}\log \texttt{ran}(0,1)}}$ (see Eq. (39))
$t_{\rm ev} \leftarrow t + |x_{\rm ev} - x|$
for $t^* = \mathrm{int}(t)+1, \ldots, \mathrm{int}(t_{\rm ev})$:
  { **print** $x + \sigma(t^* - t)$ (equal-time samples)
$x \leftarrow x_{\rm ev}$; $\sigma \leftarrow -\sigma$; $t \leftarrow t_{\rm ev}$ ("zig-zag")
output $\{x,\sigma\}, t$

C. Extending the consensus principle

We now replace the Metropolis filter in Alg. 7 (`zig-zag`) (contained in the expression for $x_{\rm ev}$) by the factorized Metropolis filter, and then use the consensus principle.

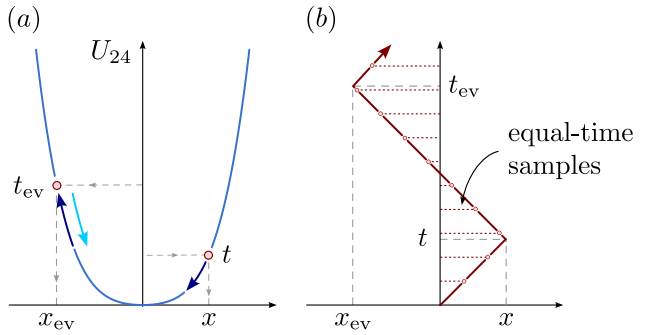


Fig. 9. Zig-zag algorithm (continuous-time event-driven lifted Metropolis chain). (a) The particle swings about the origin, turning around at positions $x_{\rm ev}$ [sampled in Eq. (39)]. (b) Piecewise deterministic constant-velocity trajectory. Particle positions are sampled at equal time steps.

Starting again at $x=0$, the particle now climbs up one hill for the harmonic factor and one for the quartic factor (see Fig. 10). For each factor, we can redo the argument of Eq. (37), with U_2 or U_4 instead of U_{24}. In analogy with Eqs. (38) and (39), we can thus sample two "candidate" events,

$$x_{\rm ev}^{(2)} = \sigma\sqrt{-2\beta^{-1}\log \texttt{ran}(0,1)}, \tag{40}$$

$$x_{\rm ev}^{(4)} = \sigma\sqrt[4]{-4\beta^{-1}\log \texttt{ran}(0,1)}, \tag{41}$$

with two independent random numbers. This means that the harmonic potential accepts all moves up to $x_{\rm ev}^{(2)}$, and the quartic potential accepts all moves up to $x_{\rm ev}^{(4)}$. Therefore, the consensus of the two factors is broken by the candidate event that comes first:

$$x_{\rm ev} = \sigma \min\left(|x_{\rm ev}^{(2)}|, |x_{\rm ev}^{(4)}|\right). \tag{42}$$

At that position, the velocity must be reversed. We may again collect positions x at equal time steps. This algorithm is implemented in Alg. 8 (`factor-zig-zag`), which samples the Boltzmann distribution π_{24}.

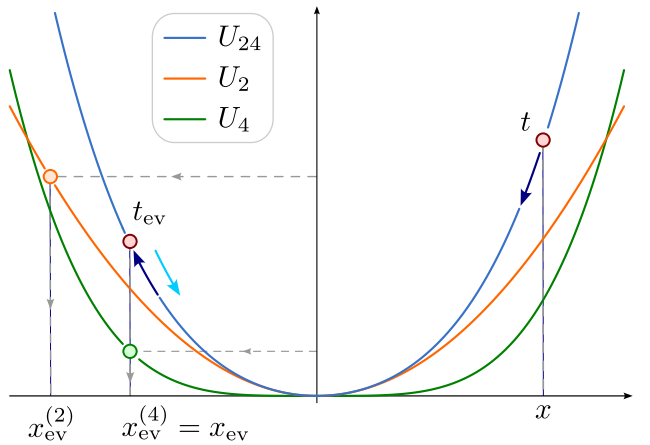


Fig. 10. Factorized zig-zag algorithm. Starting from x (here with $\sigma=-1$), the next event is given by the earliest event between $x_{\rm ev}^{(2)}$ and $x_{\rm ev}^{(4)}$ (here by $x_{\rm ev}^{(4)} = x_{\rm ev}$).

Algorithm 8. `factor-zig-zag`. Factorized zig-zag algorithm accepting moves in direction σ until the consensus of the harmonic and the quartic factor is broken at position x_{ev}.

procedure `factor-zig-zag`
input $\{x,\sigma\},t$ (lifted sample with $\sigma x \le 0$)
$x_{\text{ev}}^{(2)} \leftarrow \sigma\sqrt{-2\beta^{-1}\log \texttt{ran}(0,1)}$ (see Eq. (40))
$x_{\text{ev}}^{(4)} \leftarrow \sigma\sqrt[4]{-4\beta^{-1}\log \texttt{ran}(0,1)}$ (see Eq. (41))
$x_{\text{ev}} \leftarrow \sigma\min(|x_{\text{ev}}^{(2)}|,|x_{\text{ev}}^{(4)}|)$
$t_{\text{ev}} \leftarrow t + |x_{\text{ev}} - x|$
for $t^* = \text{int}(t)+1,\ldots,\text{int}(t_{\text{ev}})$:
  { **print** $x + \sigma(t^* - t)$ (sample of π_{24})
$x \leftarrow x_{\text{ev}}$; $\sigma \leftarrow -\sigma$; $t \leftarrow t_{\text{ev}}$ ("zig-zag")
output $\{x,\sigma\},t$

V. THINNING: OR AVOIDING EVALUATION

In molecular dynamics algorithms such as Alg. 1, forces must be computed precisely to keep the trajectory on track. In contrast, Monte Carlo algorithms are decision problems where proposed moves must be accepted with a filter, for example the Metropolis filter $\min[1,\exp(-\beta\Delta U)]$. As we discuss in this section, we can often base the accept/reject decision on a bounding potential $\widehat{U}$, and thus avoid computing U, ΔU, and their exponentials (Sec. V A). In the continuous-time setting, we simply evaluate the derivative of the bounding potential and of the potential U to eliminate all bias due to the bounding (Sec. V B).

By combining this "thinning" approach[13] with factorization, we may, for the anharmonic oscillator, base our decision to accept moves on the consensus of harmonic and quartic bounding potentials. At the end, we will have a Monte Carlo algorithm that evaluates a single factor potential, and only at the position where the proposed move is rejected by the bounding potential of that same factor (Sec. V C). In the companion paper,[2] we generalize this approach to realistic simulations of particles with long-range interactions that sample the Boltzmann distribution $\exp(-\beta U)$ without ever evaluating U.

A. Introducing the bounding potential

We say, that $\widehat{U}$ is a bounding potential of a potential U if, for any pair of configurations x and x', it satisfies

$$\min\left(1,\mathrm{e}^{-\beta\Delta\widehat{U}}\right) \le \min\left(1,\mathrm{e}^{-\beta\Delta U}\right) \quad \forall x,x' \in \Omega, \tag{43}$$

where $\Delta\widehat{U} = \widehat{U}(x') - \widehat{U}(x)$ and $\Delta U = U(x') - U(x)$. Equation (43) requires $\mathrm{d}\widehat{U}/\mathrm{d}x$ and $\mathrm{d}U/\mathrm{d}x$ to have the same sign everywhere, with $|\mathrm{d}\widehat{U}/\mathrm{d}x| \ge |\mathrm{d}U/\mathrm{d}x|$. Concretely, we define the harmonic and quartic bounding potentials recursively for $n = 0, \pm1, \pm2$, etc., as

$$\widehat{U}_2(n) = \begin{cases} 0 & \text{if } n = 0 \\ \widehat{U}_2(|n|-1) + |n| & \text{if } n \in \mathbb{Z}\backslash\{0\}, \end{cases}$$
$$\widehat{U}_4(n) = \begin{cases} 0 & \text{if } n = 0 \\ \widehat{U}_4(|n|-1) + |n^3| & \text{if } n \in \mathbb{Z}\backslash\{0\}. \end{cases} \tag{44}$$

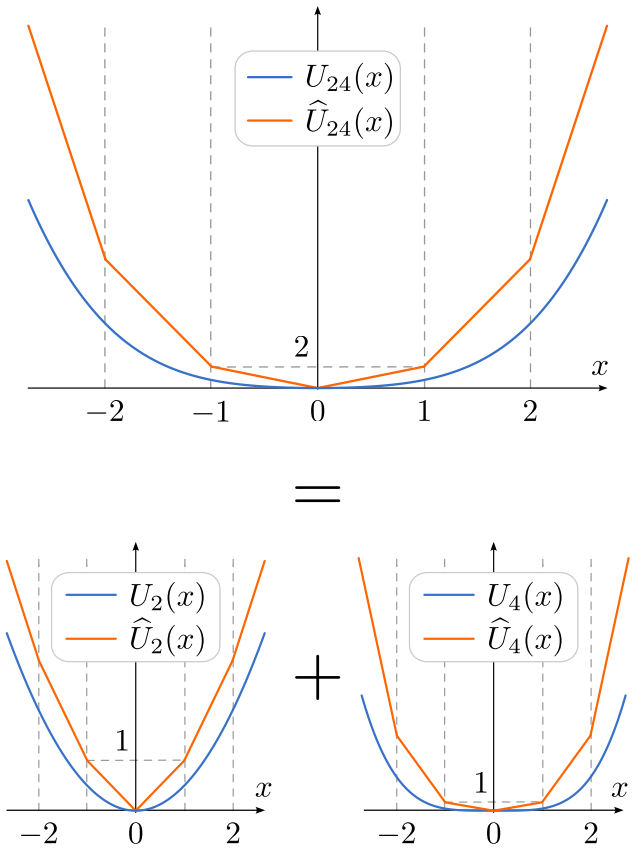


Fig. 11. Anharmonic bounding potential $\widehat{U}_{24}$ and its harmonic and quartic constituents $\widehat{U}_2$ and $\widehat{U}_4$.

These definitions are extended to non-integer arguments x by linear interpolation. The anharmonic bounding potential is then defined as $\widehat{U}_{24}(x) = \widehat{U}_2(x) + \widehat{U}_4(x)$ (see Fig. 11).

A bounding potential can simplify the decision to accept a move because a pebble $0 < \Upsilon < 1$ that falls below $\exp(-\beta\Delta\widehat{U})$ also falls below $\exp(-\beta\Delta U)$ (see Fig. 12(a)). In the remaining algorithms, we use a two-pebble strategy for the decision to accept or reject a move. The first pebble $0 < \Upsilon_1 < 1$ decides whether a move is accepted with respect to the bounding potential. Otherwise, if Υ_1 rejects the move,

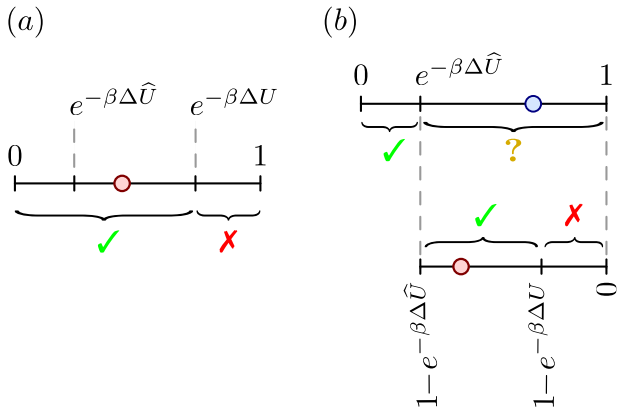


Fig. 12. Single-pebble and two-pebble decisions in the Metropolis algorithm. (a) A single pebble Υ illustrating that acceptance with respect to the bounding potential implies acceptance with respect to U. (b) The first pebble Υ_1 makes a decision with respect to the bounding potential. In the case of rejection, a second pebble Υ_2 definitely decides on the move.

we use a second pebble Υ_2 to decide whether the first-pebble rejection with respect to $\widehat{U}$ stands with respect to U (see Fig. 12(b)). A rescaling, with $0 < \Upsilon_2 < 1$, allows us to definitely reject the move if

$$\Upsilon_2 < \frac{1 - \mathrm{e}^{-\beta\Delta U}}{1 - \mathrm{e}^{-\beta\Delta\widehat{U}}}. \tag{45}$$

The two-pebble bounding-potential algorithm is implemented in Alg. 9 (`bounded-lifted`) for the anharmonic oscillator. It again samples the Boltzmann distribution π_{24}, although most positive decisions are taken on the basis of the bounding potential. The remaining evaluations of U or its derivative will disappear in subsequent algorithms.

Algorithm 9. `bounded-lifted`. Discrete-time bounded-lifted Metropolis algorithm using two-pebble decisions. The second pebble is used and the true potential U_{24} is evaluated only after a first-pebble rejection with respect to the bounding potential $\widehat{U}_{24}$.

procedure `bounded-lifted`
input $\{x, \sigma\}$ (lifted sample at time t)
$\Delta \leftarrow \mathtt{ran}(0, \delta)$ $(\delta > 0)$
$x' \leftarrow x + \sigma\Delta$; $\Upsilon_1 \leftarrow \mathtt{ran}(0,1)$;
if $\Upsilon_1 < \min(1, \mathrm{e}^{-\beta\Delta\widehat{U}_{24}})$:
  $\{ x \leftarrow x'$
else :
$$\begin{cases} \Upsilon_2 \leftarrow \mathtt{ran}(0,1) \\ \textbf{if } \Upsilon_2 > \dfrac{1 - \mathrm{e}^{-\beta\Delta U_{24}}}{1 - \mathrm{e}^{-\beta\Delta\widehat{U}_{24}}} : x \leftarrow x' \\ \textbf{else} : \sigma \leftarrow -\sigma \end{cases}$$
output $\{x, \sigma\}$ (lifted sample at time $t+1$)

B. Continuous-time thinning

The bounded-lifted Metropolis algorithm, Alg. 9 (`bounded-lifted`), generalizes to continuous time. For the anharmonic oscillator, we first consider $\sigma = +1$ and positive x between n and $n+1$, where the decision of Eq. (45), for the second pebble, becomes

$$\Upsilon_2 < \frac{1 - \mathrm{e}^{-\beta\Delta U_{24}}}{1 - \mathrm{e}^{-\beta\Delta\widehat{U}_{24}}} \rightarrow \Upsilon_2 < \frac{\mathrm{d}U_{24}/\mathrm{d}x}{\mathrm{d}\widehat{U}_{24}/\mathrm{d}x}. \tag{46}$$

The piecewise linear anharmonic bounding potential $\widehat{U}_{24}$ simplifies the event-driven formulation. Rather than to walk up the anharmonic potential until the change of potential satisfies $\Delta U_{24} = -\beta^{-1}\log \mathtt{ran}(0,1)$ (see Eq. (38) and Fig. 9), we now run up a bounding potential of constant slope $\widehat{q}$ with

$$\widehat{q} = \frac{\mathrm{d}}{\mathrm{d}x}\widehat{U}_{24}(x)|_{x\in S_n} = n + 1 + (n+1)^3, \tag{47}$$

where $S_n = [n, n+1)$ and $n \in \mathbb{N}$. The change in potential $\Delta\widehat{U}_{24}(x) = -\beta^{-1}\log \mathtt{ran}(0,1)$ then translates into the advance of the position as

$$x_{\mathrm{ev}} = x_0 - (\beta\widehat{q})^{-1}\log \mathtt{ran}(0,1). \tag{48}$$

The event rate $\beta\widehat{q}$ is constant in the sector S_n, but if x_{ev} falls outside of S_n, it is invalid. In this case, a "boundary event" is triggered, and the particle is placed at the right boundary of S_n, without changing the direction σ. Otherwise (if $x_{\mathrm{ev}} \in S_n$), the direction σ is reversed if the condition on the pebble Υ_2 in Eq. (46) is satisfied (see Fig. 13).

Our description of the continuous-time bounded-lifted Metropolis algorithm was for the case $\sigma = 1$, that is, for a pebble that climbs up the $x > 0$ branch of the potential. The general case is implemented in Alg. 10 (`bounded-zig-zag`), which again samples the Boltzmann distribution π_{24}.

Algorithm 10. `bounded-zig-zag`. Continuous-time bounded-lifted Metropolis algorithm. It need not invert the potential U_{24} [compare with Eq. (39)], foreshadowing the use of bounding potentials in real-world applications.

procedure `bounded-zig-zag`
input $\{x, \sigma\}, t$ (lifted sample)
if $\sigma x < 0$: $x_0 \leftarrow 0$; **else**: $x_0 \leftarrow x$ (starting point)
$n \leftarrow \mathrm{int}(|x_0|)$; $\widehat{q} \leftarrow n + 1 + (n+1)^3$; $\tilde{\sigma} \leftarrow \sigma$
$x_{\mathrm{ev}} \leftarrow x_0 + \sigma[-(\beta\widehat{q})^{-1}\log \mathtt{ran}(0,1)]$ (see Eq. (48))
if $|x_{\mathrm{ev}}| > n + 1$:
  $\{ x_{\mathrm{ev}} \leftarrow \sigma(n+1)$
else if $\mathtt{ran}(0,1) < |x_{\mathrm{ev}} + x_{\mathrm{ev}}^3|/\widehat{q}$:
  $\{ \tilde{\sigma} \leftarrow -\sigma$
$t_{\mathrm{ev}} \leftarrow t + |x_{\mathrm{ev}} - x|$
for $t^* = \mathrm{int}(t) + 1, \ldots, \mathrm{int}(t_{\mathrm{ev}})$:
  $\{$ **print** $x + \sigma(t^* - t)$ (equal-time samples)
$x \leftarrow x_{\mathrm{ev}}$; $\sigma \leftarrow \tilde{\sigma}$; $t \leftarrow t_{\mathrm{ev}}$ ("zig-zag")
output $\{x, \sigma\}, t$

C. Thinning with consensus

Algorithm 10 (`bounded-zig-zag`) avoids the inversion in Eq. (39) of the potential U_{24}, and only evaluates the derivative $\mathrm{d}U_{24}/\mathrm{d}x$ at $x = \widehat{x}_{\mathrm{ev}}$. At the end of our journey through advanced Markov chain Monte Carlo sampling, we combine the consensus principle underlying factorization with that of thinned, lifted Metropolis chains and sample $\pi_{24} = \exp(-\beta U_{24})$ without ever evaluating the potential U_{24} nor its derivative. The use of bounding potentials generalizes to applications in particle systems with long-range interactions. In the anharmonic oscillator, we illustrate the basic idea[14] with the harmonic and quartic factor potentials U_2 and U_4 and their bounding potentials $\widehat{U}_2$ and $\widehat{U}_4$.

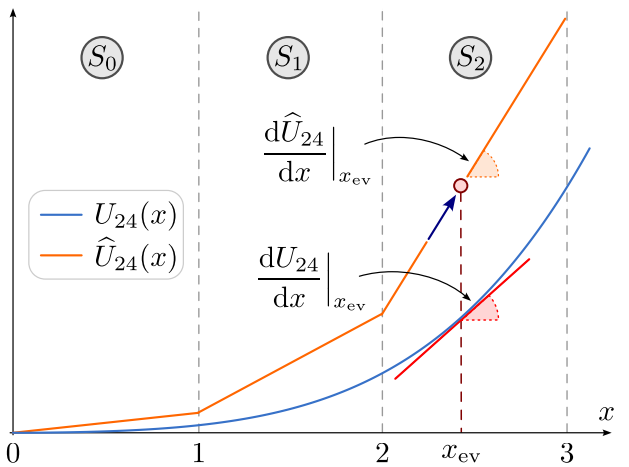


Fig. 13. Continuous-time version of the bounded-lifted Metropolis algorithm. The proposed event x_{ev} is confirmed by comparing the derivatives of the true potential U_{24} and the bounding potential $\widehat{U}_{24}$ [see Eq. (46)].

With factorization, two candidate events $x_{\rm ev}^{(2)}$ and $x_{\rm ev}^{(4)}$ can be sampled by means of Eq. (48), with bounding event rates $\beta(n+1)$ and $\beta(n+1)^3$, respectively. When both events fall outside the sector S_n where the bounding rates are valid, a boundary event is triggered. Otherwise, the earliest candidate event $x_{\rm ev} \in S_n$ (either $x_{\rm ev}^{(2)}$ or $x_{\rm ev}^{(4)}$) is confirmed with one of the probabilities

$$\left.\frac{\mathrm{d}U_2/\mathrm{d}x}{\mathrm{d}\widehat{U}_2/\mathrm{d}x}\right|_{x_{\rm ev}^{(2)}} = \frac{x_{\rm ev}}{n+1}, \quad \left.\frac{\mathrm{d}U_4/\mathrm{d}x}{\mathrm{d}\widehat{U}_4/\mathrm{d}x}\right|_{x_{\rm ev}^{(4)}} = \frac{x_{\rm ev}^3}{(n+1)^3}. \tag{49}$$

This bounded-lifted, and in addition factorized, Metropolis algorithm, largely analogous to Alg. 10, is implemented in Alg. 11 (`bounded-factor-zig-zag`). Remarkably, it evaluates the derivative of only one factor potential. Most of the decisional burden is carried by the bounding potentials, for example, which factor to choose for the next event. The decision-problem footprint of Monte Carlo algorithms thus appears clearly, as there are different ways to reach a statistically correct decision. In molecular dynamics, in contrast, only a single Newtonian trajectory exists.

Algorithm 11. `bounded-factor-zig-zag`. Factorized version of Alg. 10, with one candidate event for each factor. For each event, only one factor derivative is evaluated.

procedure `bounded-factor-zig-zag`
input $\{x,\sigma\},t$ (lifted sample at time t)
if $\sigma x<0$: $x_0\leftarrow 0$; **else**: $x_0\leftarrow x$ (starting point)
$n\leftarrow \mathrm{int}(|x_0|)$; $\widehat{q}^{(2)}\leftarrow n+1$; $\widehat{q}^{(4)}\leftarrow (n+1)^3$; $\tilde{\sigma}\leftarrow\sigma$
$x_{\rm ev}^{(2)}\leftarrow x_0+\sigma[-(\beta\widehat{q}^{(2)})^{-1}\log \mathtt{ran}(0,1)]$ (see Eq. (48))
$x_{\rm ev}^{(4)}\leftarrow x_0+\sigma[-(\beta\widehat{q}^{(4)})^{-1}\log \mathtt{ran}(0,1)]$ (see Eq. (48))
if $\min(|x_{\rm ev}^{(2)}|,|x_{\rm ev}^{(4)}|)>n+1$: $x_{\rm ev}\leftarrow\sigma(n+1)$
else if $|x_{\rm ev}^{(2)}|<|x_{\rm ev}^{(4)}|$:
  { $x_{\rm ev}\leftarrow x_{\rm ev}^{(2)}$; **if** $\mathtt{ran}(0,1)<|x_{\rm ev}|/\widehat{q}^{(2)}$: $\tilde{\sigma}\leftarrow-\sigma$
else:
  { $x_{\rm ev}\leftarrow x_{\rm ev}^{(4)}$; **if** $\mathtt{ran}(0,1)<|x_{\rm ev}|^3/\widehat{q}^{(4)}$: $\tilde{\sigma}\leftarrow-\sigma$
$t_{\rm ev}\leftarrow t+|x_{\rm ev}-x|$
for $t^*=\mathrm{int}(t)+1,\ldots,\mathrm{int}(t_{\rm ev})$:
  { **print** $x+\sigma(t^*-t)$ (equal-time samples)
$x\leftarrow x_{\rm ev}$; $\sigma\leftarrow\tilde{\sigma}$; $t\leftarrow t_{\rm ev}$ ("zig-zag")
output $\{x,\sigma\},t$

Algorithm 11 (`bounded-factor-zig-zag`) samples as many candidate events as there are factors (in our case, $\widehat{x}_{\rm ev}^{(2)}$ and $\widehat{x}_{\rm ev}^{(4)}$ for the harmonic and quartic factors), thus adopting a strategy that runs into trouble when there are too many factors. A patch of Alg. 11 illustrates, in a nutshell, how factors can be bundled in the continuous-time setting, where the total event rate is the sum of the individual factor rates (see Table II). In the anharmonic oscillator, the total bounding event rate is the sum of the harmonic and the quartic bounding rates, giving us the next event with a single random number. It then remains to decide whether this event is a harmonic-bounding or a quartic-bounding event, as implemented in Alg. 12 (`bounded-factor-zig-zag(patch)`). Even for a large number of factors, we can make this decision in a few steps, using the famous Walker algorithm.[15] It is this very program that is used in state-of-the-art programs to handle millions of factors in constant time,[14] as we will discuss further in Ref. 2.

Table II. Consensus probabilities of Table I for Alg. 11 (`bounded-factor-zig-zag`) and its patch. The total event rate (the total rate of rejection by consensus) is the sum of the factor event rates, as terms of order $(\mathrm{d}t)^2$ drop out.

Harmonic \ Quartic	Accept $(1-\beta\widehat{q}^{(4)}\mathrm{d}t)$	Reject $(\beta\widehat{q}^{(4)}\mathrm{d}t)$
Accept $(1-\beta\widehat{q}^{(2)}\mathrm{d}t)$	$1-\beta(\widehat{q}^{(2)}+\widehat{q}^{(4)})\mathrm{d}t$	$\beta\widehat{q}^{(4)}\mathrm{d}t$
Reject $(\beta\widehat{q}^{(2)}\mathrm{d}t)$	$\beta\widehat{q}^{(2)}\mathrm{d}t$	0

Algorithm 12. `bounded-factor-zig-zag(patch)`. Patch of Alg. 11 illustrating the bundling of two factors into a single candidate event.

procedure `bounded-factor-zig-zag(patch)`
input $\{x,\sigma\},t$ (lifted sample at time t)
if $\sigma x<0$: $x_0\leftarrow 0$; **else**: $x_0\leftarrow x$ (starting point)
$n\leftarrow \mathrm{int}(|x_0|)$
$\widehat{q}^{(2)}\leftarrow n+1$; $\widehat{q}^{(4)}\leftarrow (n+1)^3$; $\widehat{q}\leftarrow\widehat{q}^{(2)}+\widehat{q}^{(4)}$; $\tilde{\sigma}\leftarrow\sigma$
$x_{\rm ev}\leftarrow x_0+\sigma[-(\beta\widehat{q})^{-1}\log \mathtt{ran}(0,1)]$ (see Eq. (48))
if $|x_{\rm ev}|>n+1$: $x_{\rm ev}\leftarrow\sigma(n+1)$
else if $\mathtt{ran}(0,\widehat{q})<\widehat{q}^{(2)}$:
  { **if** $\mathtt{ran}(0,1)<|x_{\rm ev}|/\widehat{q}^{(2)}$: $\tilde{\sigma}\leftarrow-\sigma$
else:
  { **if** $\mathtt{ran}(0,1)<|x_{\rm ev}|^3/\widehat{q}^{(4)}$: $\tilde{\sigma}\leftarrow-\sigma$
$t_{\rm ev}\leftarrow t+|x_{\rm ev}-x|$
for $t^*=\mathrm{int}(t)+1,\ldots,\mathrm{int}(t_{\rm ev})$:
  { **print** $x+\sigma(t^*-t)$ (equal-time samples)
$x\leftarrow x_{\rm ev}$; $\sigma\leftarrow\tilde{\sigma}$; $t\leftarrow t_{\rm ev}$ ("zig-zag")
output $\{x,\sigma\},t$

VI. CONCLUSION

We have introduced a number of modern developments in Monte Carlo sampling that go beyond direct sampling and the Metropolis algorithm. New Monte Carlo algorithms build on notions such as factorization, non-reversibility, and thinning. They increasingly find applications in physics and other sciences. The one-dimensional anharmonic oscillator has hopefully allowed us to lay bare the foundations of these nontrivial theoretical developments and concentrate on the correctness of the sampling algorithms. Questions of efficiency will be the subject of the companion paper.[2]

ACKNOWLEDGMENTS

The authors thank K. J. Wiese for helpful discussions. The authors thank the Mathematical Research Institute MATRIX in Australia where part of this research was performed.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

APPENDIX A: MATHEMATICAL DETAILS

We present some mathematical details that, for the sake of conciseness, were omitted in the main text.

As stated in Eq. (6), the period of the isolated anharmonic oscillator at energy E is

$$\tau(E) = 4\sqrt{\frac{2}{1+\sqrt{1+4E}}}\,K\left(\frac{1-\sqrt{1+4E}}{1+\sqrt{1+4E}}\right), \tag{A1}$$

where K is the complete elliptic integral of the first kind. This nontrivial integral follows from the theory of elliptic functions (see e.g., Ref. 16, Chap. 19 for a discussion on this subject). It can be obtained indirectly using the `Integrate` function of Mathematica, as illustrated in the Mathematica notebook available in the `MCMCNutshell` software package (see Appendix B). For $E \to 0$, the amplitude $x_{\max}$ of the oscillation is small. Consequently, the anharmonic potential $U_{24}(x)$ of Eq. (1) can be safely replaced by the harmonic one in this regime,

$$U_{24}(x) \sim U_2(x) = x^2/2 \quad \text{for } x_{\max} \to 0. \tag{A2}$$

By expanding $\tau(E)$ in Eq. (A1) about $E = 0$, we obtain

$$\tau(E) = 2\pi - \frac{3\pi}{2}E + \mathcal{O}(E^2) \tag{A3}$$

(see Appendix B for the use of the Mathematica `Series` function).

For small E, the period of the anharmonic oscillator coincides with that of the harmonic one, $\tau = 2\pi$, because the quartic term in the potential is negligible for $|x| \ll 1$. For large E, the quartic term dominates

$$U_{24}(x) \sim U_4(x) = x^4/4 \quad \text{for } x_{\max} \gg 0. \tag{A4}$$

In this case, we expand τ for large E and find

$$\tau(E) = \frac{\sqrt{\pi}\,\Gamma(1/4)}{\Gamma(3/4)}E^{-1/4} + \mathcal{O}(E^{-3/4}), \tag{A5}$$

where Γ denotes the Euler gamma function (see Appendix B for the Mathematica notebook). The dominant term of Eq. (A5) coincides with the period of the quartic oscillator, computed using the equivalent of Eq. (6), with amplitude $x_{\max} = (4E)^{1/4}$.

The partition function $Z(\beta)$ of the harmonic oscillator in Eq. (13) can be easily computed by means of the Mathematica `Integrate` function.

APPENDIX B: COMPUTER PROGRAMS, MATHEMATICA NOTEBOOK

This paper is accompanied by the `MCMCNutshell` software package, which is available as an open-source project on GitHub, and which can also be found on the AJP website.[17,18] The package contains Python implementations for $\beta = 1$ of all the algorithms presented above and a Mathematica notebook (in .nb and .pdf formats).

APPENDIX C: NUMERICAL TESTS

Except for Alg. 0 (`isolated-dynamics`), the 11 Monte Carlo algorithms and one molecular dynamics algorithm all sample the Boltzmann distribution π_{24} of Eq. (2). To check the correctness of our implementations (see Appendix B), we fixed an arbitrary nonzero value of $\bar{x} = 0.63$ for $\beta = 1$, computed for each algorithm the empirical probability with which the samples x satisfy $x < \bar{x}$, and compared it with the exact result

Table III. Estimated probability $\mathbb{P}(x < 0.63)$ for the anharmonic oscillator computed by the algorithms we discussed (single-σ error bars).

Algorithm	$\mathbb{P}(x < 0.63)$
1 `thermostat-dynamics`	0.8038 ± 0.0025
2 `direct-sampling`	0.8029 ± 0.0001
3 `metropolis`	0.8029 ± 0.0026
4 `factor-metropolis`	0.8004 ± 0.0035
5 `factor-metropolis(patch)`	0.8029 ± 0.0015
6 `lifted-metropolis`	0.8033 ± 0.0003
7 `zig-zag`	$0.802\,92 \pm 0.000\,09$
8 `factor-zig-zag`	0.8030 ± 0.0001
9 `bounded-lifted`	0.8036 ± 0.0004
10 `bounded-zig-zag`	$0.802\,97 \pm 0.000\,09$
11 `bounded-factor-zig-zag`	$0.802\,97 \pm 0.000\,07$
12 `bounded-factor-zig-zag(patch)`	0.8029 ± 0.0001

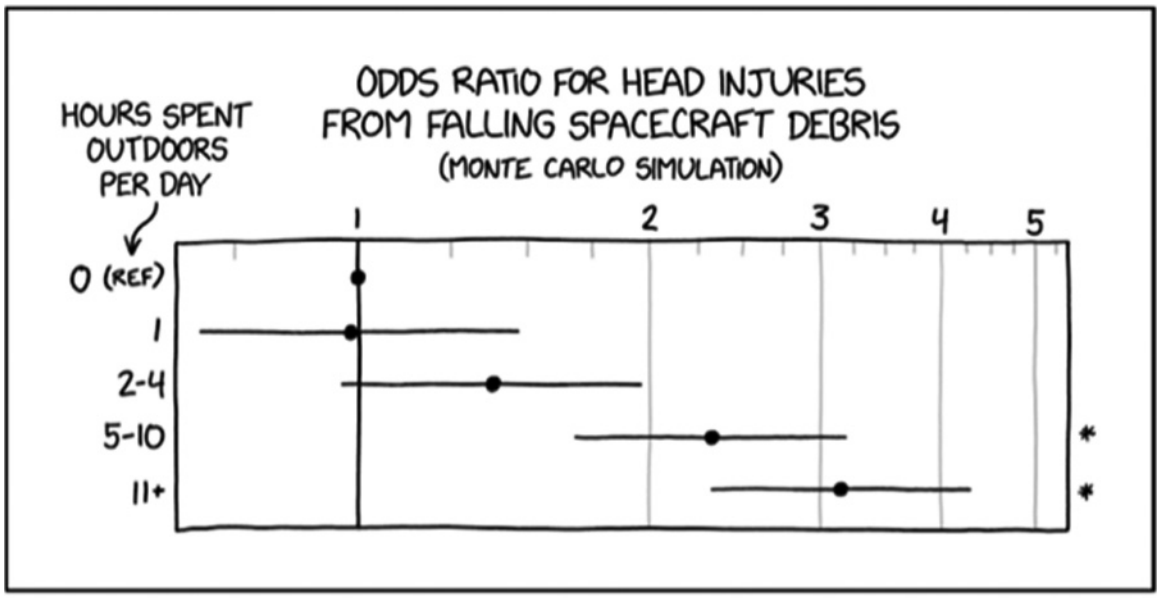
$$\mathbb{P}(x < 0.63) = Z^{-1}\int_{-\infty}^{0.63} \pi_{24}(x')\mathrm{d}x' = 0.8030254, \tag{C1}$$

computed with the Mathematica `NIntegrate` function at high precision (see Appendix B for the Mathematica notebook). Single-standard-deviation error bars were obtained from the bunching method (see Ref. 1, Sec. 1.3.5) except for Alg. 2 (`direct-sampling`), where we performed a standard Gaussian analysis. For all 12 algorithms, our results are consistent with Eq. (C1) to within three standard deviations (see Table III). The default parameter values of the implementations in `MCMCNutshell` were used. They typically correspond to run times of a few minutes.

[a]ORCID: 0009-0007-4376-2223.
[b]Author to whom correspondence should be addressed: werner.krauth@ens.fr, ORCID: 0000-0003-0183-6726.

[1]W. Krauth, *Statistical Mechanics: Algorithms and Computations* (Oxford U. P., 2006).
[2]G. Tartero, S. Vionnet, and W. Krauth, "Fast sampling of Lennard-Jones systems without cutoffs," (unpublished 2023).
[3]M. H. A. Davis, "Piecewise-deterministic Markov processes: A general class of non-diffusion stochastic models," J. R. Stat. Soc. Ser. B **46**(3), 353–376 (1984).
[4]L. D. Landau and E. M. Lifshitz, *Mechanics: Volume 1* (Elsevier Science, Amsterdam, 1982).
[5]B. Li, Y. Nishikawa, P. Höllmer, L. Carillo, A. C. Maggs, and W. Krauth, "Hard-disk pressure computations—a historic perspective," J. Chem. Phys. **157**, 234111 (2022).
[6]M. Michel, S. C. Kapfer, and W. Krauth, "Generalized event-chain Monte Carlo: Constructing rejection-free global-balance algorithms from infinitesimal steps," J. Chem. Phys. **140**, 054116 (2014).
[7]P. Diaconis, S. Holmes, and R. M. Neal, "Analysis of a nonreversible Markov chain sampler," Ann. Appl. Probab. **10**(3), 726–752 (2000).
[8]F. Chen, L. Lovász, and I. Pak, "Lifting Markov chains to speed up mixing," in *Proceedings of the 17th Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, New York, 1999). p. 275.
[9]J. Bierkens, P. Fearnhead, and G. Roberts, "The Zig-Zag process and super-efficient sampling for Bayesian analysis of big data," Ann. Stat. **47**, 1288–1320 (2019).
[10]D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov Chains and Mixing Times* (American Mathematical Society, 2008).
[11]W. Krauth, "Event-chain Monte Carlo: Foundations, applications, and prospects," Front. Phys. **9**, 229 (2021).

[12]E. A. J. F. Peters and G. de With, "Rejection-free Monte Carlo sampling for general potentials," Phys. Rev. E **85**(2), 026703 (2012).
[13]P. A. W. Lewis and G. S. Shedler, "Simulation of nonhomogeneous Poisson processes by thinning," Nav. Res. Logist. Q. **26**(3), 403–413 (1979).
[14]S. C. Kapfer and W. Krauth, "Cell-veto Monte Carlo algorithm for long-range systems," Phys. Rev. E **94**(3), 031302 (2016).
[15]A. J. Walker, "An efficient method for generating discrete random variables with general distributions," ACM Trans. Math. Software **3**(3), 253–256 (1977).
[16]NIST Digital Library of Mathematical Functions, <http://dlmf.nist.gov/>
[17]On GitHub, the MCMCNutshell package is part of the JeLLyFysh organization, and can be found at <https://github.com/jellyfysh/MCMCNutshell>.
[18]See supplementary material online for Python programs and a Mathematica notebook.



You say this daily walk will reduce my risk of death from cardiovascular disease by 30%, but also increase my risk of death by bear attack by 300%? That's a 280% increase! I'm not a sucker. I'm staying inside. (Source: https://xkcd.com/2599)