Algorithms and computations in physics (Oxford Lectures 2024)

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

Version 16 Jan 2024

First lecture (16 January 2024): Direct sampling, from children's game in Monte Carlo to the strong law of large numbers.

Contents

1 Direct sampling 1
1.1 Children on the beach in Monaco . . . 1
1.1.1 Sample spaces and random variables . . . 2
1.1.2 Pseudocode, pseudo-random numbers . . . 2
1.1.3 From Punch-card coding to Python and to ChatGPT . . . 3
1.2 Direct sampling—discrete distributions . . . 4
1.2.1 Rejection sampling . . . 4
1.2.2 Tower sampling . . . 4
1.2.3 Walker's method, $\mathcal{O}(1)$ sampling in the Saturday night problem . . . 5
1.3 Direct sampling—continuous distributions . . . 6
1.3.1 Sample transformation—simple examples . . . 6
1.3.2 Sample transformation—Gaussian, multidimensional Gaussians . . . 7
1.3.3 Rejection method . . . 9
1.4 Direct sampling: fundamental aspects . . . 10
1.4.1 Laws of large numbers—importance sampling . . . 11
1.4.2 Universal approach to the integrated distribution function . . . 13
1.4.3 What we learn about π by throwing pebbles . . . 13

1 Direct sampling

[1]

1.1 Children on the beach in Monaco

Direct sampling [1] is exemplified by a children's game popular in Monaco, south of France, whence the name "Monte Carlo" method. Children fling pebbles (small stones) into a square.

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

[1] The word *sample* has the same origin as *example*

They then count how many of them fall into the circle inscribed in the square (see [2, Sect. 1.1] for the full story).

1.1.1 Sample spaces and random variables

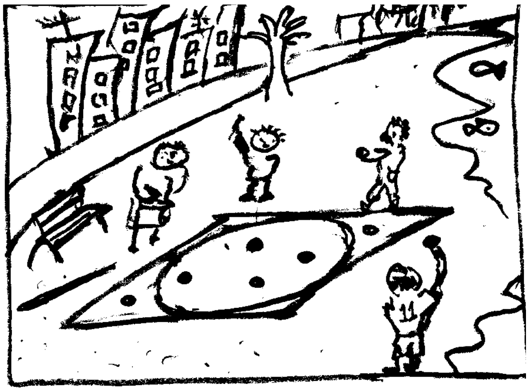

Figure 1: Children on the Monte Carlo beach, computing the number π.

The square in Fig. 1 constitutes a sample space Ω. [2] Modern probability theory, since Kolmogoroff, then assigns probabilities, not in general to individual samples (x, y), but to subsets of the sample space Ω. A small subset of size $\mathrm{d}x\ \mathrm{d}y$ around a point (x, y) then has the probability

$$\pi^{\square}(x,y)\mathrm{d}x\ \mathrm{d}y\ = \begin{cases} \text{const } \mathrm{d}x\ \mathrm{d}y & \text{if } x,y \in \Omega^{\square} \\ 0 & \text{else} \end{cases}. \tag{1}$$

where, in this case, we know that const $= 1/4$. The pebbles that the children throw are (exact, perfect) samples of the distribution π, and the act of throwing the pebbles is denoted as "sampling (from) the distribution π".

On the Monte Carlo beach, a circle is inscribed inside the square. While all pebbles inside the square are referred to as *trials*, those inside the circle are *hits*. The function on the sample space defines a "random variable" [3]. $\mathcal{O} : \Omega \to \mathbb{R}$:

$$\mathcal{O}((x,y)) = \begin{cases} 1 & \text{if } (x,y) \in \odot \\ 0 & \text{else} \end{cases} \tag{2}$$

Clearly, $\mathcal{O}$ is a Bernoulli random variable that we can in principle define without considering the sample space

$$\mathcal{O} = \begin{cases} 1 & P = \pi/4 \\ 0 & P = 1 - \pi/4 \end{cases}, \tag{3}$$

illustrating that sample spaces tend to disappear in discussions of probability theory [1, p. 27]. Unfortunately, we have forgotten all about π, and thus need to estimate it from the pebble throws, that is, from eq. (2). All of a sudden, we have a problem, not of probability theory, but of statistics, and are back with the sample space, that is, the square on the Monte Carlo beach.

1.1.2 Pseudocode, pseudo-random numbers

The children's game is the first of 150 algorithms that we will discuss in the present lecture series. We will specify them through self-explanatory pseudo-code (see Alg. 1 (`direct-pi`)).

[2] Sample space: the set of all outcomes of the pebble-throw experiments

[3] *random* comes from an old French word that still survives in modern French *randonnée*, which means *hike*

In Alg. 1 (`direct-pi`), as throughout this course, we take for granted the existence of uniform random numbers $\texttt{ran}(a, b)$ in the interval between a and b. During 80 years, now, such numbers have been generated as pseudo-random numbers, in other words deterministic procedures that look random, and that condense deep concepts from number theory and cryptography. Up to the year 2000, roughly, problems with random numbers were common, and users on high alert, running batteries of checks on their results for different classes of random numnbers. In modern times, the quality of random-number generators has much improved [3], although they continue to have flaws and will themselves be superseded by even more sophisticated variants. What will remain is the pseudo-code, as in Algorithm 1, and we will stick to it.

```
procedure direct-pi
$N_{\text{hits}} \leftarrow 0$ (initialize)
for $i = 1, \ldots, N$:
    $x \leftarrow \texttt{ran}(-1, 1)$
    $y \leftarrow \texttt{ran}(-1, 1)$
    if $x^2 + y^2 < 1$:  $N_{\text{hits}} \leftarrow N_{\text{hits}} + 1$
output $N_{\text{hits}}$
——
```

Algorithm 1: `direct-pi`. Using the children's game with N pebbles to compute π.

1.1.3 From Punch-card coding to Python and to ChatGPT

The pseudocode of Alg. 1 (`direct-pi`), and of all algorithms that follow must be communicated to a computer. This, over the decades, first consisted in punching holes into cards, of which one had huge piles. Later, pseudocode was to be translated into one or the other computer language, with Python becoming more and more popular, although the language itself changes over time. Today, pseudo-code as Alg. 1 (`direct-pi`) may be simply copied and pasted into a window of an AI program See this ChatGPT output. This is extremely helpful, and it may make interfacing with a computer appear as effortless as flinging 4×10^8 pebbles on a modern laptop rather than on the Monte Carlo beach. Nevertheless, the concept of effortless creation certainly remains a mirage. In whichever way one goes about it, actual running the short example programs of this course will be a powerful way to confront oneself with its subjects. As this has become so easy, why not implement them all?

Run	N_{hits}	Estimate of π
1	3156	3.156
2	3150	3.150
3	3127	3.127
4	3171	3.171
5	3148	3.148

Table 1: Results of five runs of Alg. 1 (`direct-pi`) with $N = 4000$. Approximations of π are obtained by shifting a decimal point.

Implementing Alg. 1 (`direct-pi`), not in a computer, but on the beach, the children record the proportion of "hits", that is, of the fraction of pebbles inside the circle (see Table 1.1.3, for $N = 4000$). Without knowing it, they have computed a two-dimensional integral:

$$\underbrace{\frac{N_{\text{hits}}}{\text{trials}} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{O}_i}_{\text{sampling}} \simeq \underbrace{\langle \mathcal{O} \rangle = \frac{\int_{-1}^{1} \mathrm{d}x \ \int_{-1}^{1} \mathrm{d}y \ \pi(x, y)\mathcal{O}(x, y)}{\int_{-1}^{1} \mathrm{d}x \ \int_{-1}^{1} \mathrm{d}y \ \pi(x, y)}}_{\text{integration}} . \tag{4}$$

On the left of eq. (4), there is the "running average" of the random variable $\mathcal{O}$, evaluated at the samples (x_i, y_i). The probability distribution $\pi(x, y)$ is absent, rather than being evaluated, it is sampled. By virtue of the law of large numbers, this running average converges (in a way we will need to discuss precisely, later) to the expectation (mean value) of $\mathcal{O}$, expressed as a ratio of two two-dimensional integrals. The same approach allows one to evaluate (that is, to sample) high-dimensional integrals in physics and other domains, if only we can think of how to do the sampling. What would happen in the limit $N \to \infty$ is easy to grasp, but we will later pursue the more relevant question what precisely we learn from 3156 hits for 4000 trials.

1.2 Direct sampling—discrete distributions

Sampling defines the Monte Carlo method (and some other fields, like machine learning), and direct sampling is the first step on this journey. In this section, we treat the case of a discrete distribution in a sample space $\Omega = \{1, \ldots, K\}$, with non-uniform weights $\{\pi_1, \ldots, \pi_K\}$. The sampling is non-trivial already for moderate values of K, as we experience ourselves each week in the "Saturday night problem", when the sample 1 and its probability π_1 correspond to studying (this course), 2 corresponds to cleaning the house, 3 to getting some exercise, and so on. There are only a few choices but, clearly, it takes us so long to decide what to do, after all.

The optimal way to go about sampling a finite distribution has not been known for a very long time [4], but we will first treat the standard approaches, rejection sampling, and tower sampling, which are interesting in their own right.

1.2.1 Rejection sampling

Rejection sampling is the simplest approach one can think of: place all the items into a big box such that they do not overlap, then throw pebbles into the box and stop when one of them hits an item. It helps to make the box rectangular, and to place the items side by side (see Alg. 2 (`reject-finite`)).

```
    procedure reject-finite
    $\pi_{\max} \leftarrow \max_{k=1}^{K} \pi_k$
1   $k \leftarrow$ nran$(1, K)$
    $\Upsilon \leftarrow$ ran$(0, \pi_{\max})$
    if $\Upsilon > \pi_k$:  goto 1
    output $k$
    ——
```

Algorithm 2: `reject-finite`. Sampling a finite distribution $\{\pi_1, \ldots, \pi_K\}$ with a rejection algorithm.

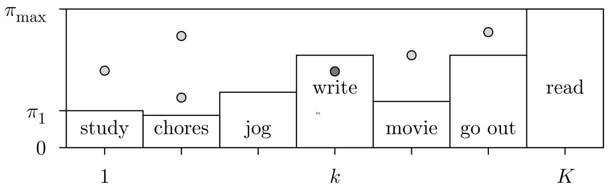

Figure 2: Saturday night problem solved by Alg. 2 (`reject-finite`).

1.2.2 Tower sampling

In what we call "tower sampling", the K probabilities are stacked onto each other, and all the probabilities are added up as $\Pi_0 = 0$, $\Pi_1 = \pi_1$, $\Pi_2 = \pi_1 + \pi_2$, and so on. Then we throw a uniform pebble $\texttt{ran}(0, \Pi_K)$.

```
    procedure tower-sample
    input $\{\pi_1, \ldots, \pi_K\}$
    $\Pi_0 \leftarrow 0$
    for $l = 1, \ldots, K$:  $\Pi_l \leftarrow \Pi_{l-1} + \pi_l$
    $\Upsilon \leftarrow \texttt{ran}(0, \Pi_K)$
  * find $k$ with $\Pi_{k-1} < \Upsilon < \Pi_k$
    output $k$
    ——
```

Algorithm 3: `tower-sample`. Tower sampling of a finite distribution $\{\pi_1, \ldots, \pi_K\}$ without rejections. The search indicated in the line marked with a $*$ makes its complexity to be $\mathcal{O}(\log K)$ per sample, once we have constructed the tower, in $\mathcal{O}(K)$ steps.

Tower sampling can be applied to discrete distributions with a total number K in the hundreds, thousands, or even millions. It is used when the naive rejection method of Fig. 2 fails because of too many rejections. Tower sampling becomes impracticable only when the probabilities $\{\pi_1, \ldots, \pi_K\}$ can no longer be listed. This rejection-free method is not optimal, but easy to implement and of theoretical interest.

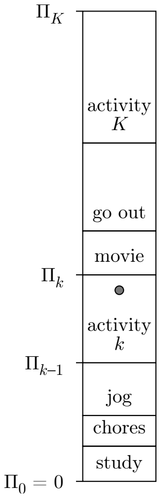

Figure 3: Saturday night problem solved by tower sampling.

1.2.3 Walker's method, $\mathcal{O}(1)$ sampling in the Saturday night problem

Tower sampling of the distribution $\pi_1, \ldots, \pi_K$ requires $\mathcal{O}(\log K)$ operations for each time step (after an initial preparation of $\mathcal{O}(K)$. Although a logarithm is not a high price to pay, it can be avoided by the extremely clever Walker's method [4] (see Fig. 4). To construct the table, we separate the pile of larger-than-average p_i ("tall") from the pile of smaller-than-average ones ("small"). At each time $t = 1, \ldots, K$, we set together a small and a large probability and cut them off at the average. What remains is put back into the corresponding pile. As we used up two p_i (one from the small, one from the tall), and and put back one, the total size of the two piles has decreased by one, and neither of the piles will empty. To sample from the table, we sample $i = \texttt{nran}(1, K)$, and a pebble to find out which of the two elements we belong to. Walker's algorithm is extremely useful when the p_i do not change over time.

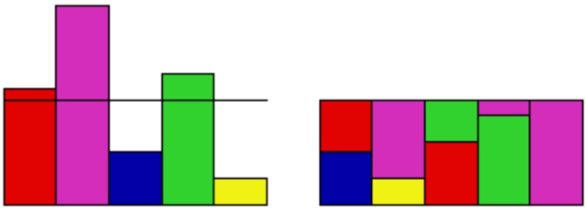

Figure 4: Saturday night problem solved by Walker's algorithm. There are at most two pieces in each parquet

1.3 Direct sampling—continuous distributions

We now consider the sampling problem for continuous distributions, but start by approaching the continuum limit from the aforementioned discretized problem. We discover a deep relation between integration and sampling, and between the substitution of variables and what we call a *sample transformation.*

1.3.1 Sample transformation—simple examples

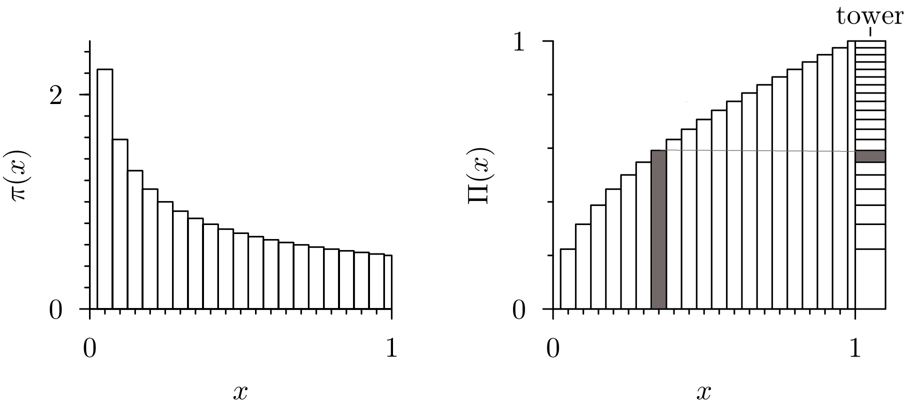

Figure 5: Tower sampling for a discretized version of $\pi(x) = (\gamma + 1)x^\gamma$ in the interval $x \in (0, 1]$. The case $\gamma = -\frac{1}{2}$ is shown).

We consider the continuum limit of tower sampling. As an example, let us sample random numbers $0 < x < 1$ distributed according to an algebraic function $\pi(x) = (\gamma + 1)x^\gamma$ (with $\gamma > -1$) (see Fig. 5). A pebble in the tower must be identified with its corresponding x position. In the continuum limit, we find

$$\begin{aligned} \pi(x) &= (\gamma+1)x^\gamma \text{ for } 0 < x < 1, \\ \Pi(x) &= \int_0^x \mathrm{d}x\ \pi(x') = x^{\gamma+1} = \texttt{ran}(0,1)\,, \\ x &= \texttt{ran}(0,1)^{1/(\gamma+1)}\,. \end{aligned} \tag{5}$$

The transformation method is best "seen" as a sample transformation: For the above algebraic function, we can transform the integral over a flat distribution into the integral of the target distribution:

$$\int_0^1 \mathrm{d}\Upsilon \xrightarrow[\text{transform}]{\text{integral}} \text{const} \int_0^1 \mathrm{d}x\ x^\gamma.$$

We now treat Υ ("Upsilon") as a sample $\Upsilon = \texttt{ran}(0,1)$, and it transforms as follows:

$$\mathrm{d}\Upsilon \ = \text{const} \cdot \mathrm{d}x \ x^{\gamma}.$$

$$\texttt{ran}(0,1) = \Upsilon = \text{const}' \cdot x^{\gamma+1} + \text{const}''.$$

Finally (checking that the bounds of $\texttt{ran}(0,1)$ correspond to $x = 0$ and $x = 1$), this results in

$$x = \texttt{ran}(0,1)^{1/(\gamma+1)}\,, \tag{6}$$

in agreement with eq. (5).

As a second example of sample transformation, we consider exponentially distributed random numbers, so that $\pi(x) \propto \mathrm{e}^{-\lambda x}$ for $x \geq 0$. We again write

$$\int_0^1 \mathrm{d}\Upsilon \ = \text{const} \int_0^\infty \mathrm{d}x \ \mathrm{e}^{-\lambda x} \tag{7}$$

and see on the left-hand side of eq. (7), that $\Upsilon = \texttt{ran}(0,1)$, so that

$$\mathrm{d}\Upsilon \ = \text{const} \cdot \mathrm{d}x \ \mathrm{e}^{-\lambda x},$$
$$\texttt{ran}(0,1) = \Upsilon = \text{const}' \cdot \mathrm{e}^{-\lambda x} + \text{const}''.$$

Checking the bounds $x = 0$ and $x = \infty$, this leads to

$$-\frac{1}{\lambda} \log \texttt{ran}(0,1) = x. \tag{8}$$

Each "negative logarithm of $\texttt{ran}(0,1)$" will from now be recognized as an exponential random number.

1.3.2 Sample transformation—Gaussian, multidimensional Gaussians

In our exploration of sample transformations, we move ahead to more complex, and physically relevant, cases involving Gaussian random numbers x that are a staple in many fields of science. Restricting ourselves to unit variance $\sigma^2 = 1$, they are distributed as

$$\pi(x) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{x^2}{2}\right].$$

Subroutines for Gaussian random numbers are readily available, and we may lack the curiosity to look under the hood of the corresponding algorithms. However, in our "School-of-seeing" approach, Gaussians again illustrate sample transformation and connect with what we learned a long time ago.

To evaluate the error integral

$$\int_{-\infty}^{\infty} \frac{\mathrm{d}x}{\sqrt{2\pi}} \mathrm{e}^{-x^2/2} = 1, \tag{9}$$

we recall that we should square eq. (9):

$$\left[\int_{-\infty}^{\infty} \frac{\mathrm{d}x}{\sqrt{2\pi}} \exp\left(-x^2/2\right)\right]^2 = \int_{-\infty}^{\infty} \frac{\mathrm{d}x}{\sqrt{2\pi}} \mathrm{e}^{-x^2/2} \int_{-\infty}^{\infty} \frac{\mathrm{d}y}{\sqrt{2\pi}} \mathrm{e}^{-y^2/2} \tag{10}$$

$$= \int_{-\infty}^{\infty} \frac{\mathrm{d}x \ \mathrm{d}y}{2\pi} \exp\left[-(x^2+y^2)/2\right], \tag{11}$$

introduce polar coordinates ($\mathrm{d}x\ \mathrm{d}y\ = r\mathrm{d}r\ \ \mathrm{d}\phi$),

$$\ldots = \int_0^{2\pi} \frac{\mathrm{d}\phi}{2\pi} \int_0^\infty r\,\mathrm{d}r\ \exp\left(-r^2/2\right),$$

and finally substitute $r^2/2 = \Upsilon$ ($r\,\mathrm{d}r\ = \mathrm{d}\Upsilon$)

$$\ldots = \underbrace{\overbrace{\int_0^{2\pi} \frac{\mathrm{d}\phi}{2\pi}}^{\phi=\mathtt{ran}(0,2\pi)}}_{1}\ \underbrace{\overbrace{\int_0^\infty \mathrm{d}\Upsilon\ \mathrm{e}^{-\Upsilon}}^{\Upsilon=-\log \mathtt{ran}(0,1)}}_{1}. \tag{12}$$

Equation (12) famously implies eq. (9), but it also shows us the way to obtaining independent Gaussian samples x and y. It suffices to sample the uniform distribution of ϕ and the exponential distribution for Υ, as indicated, and then to transform everything back to have two independent Gaussian random variables x and y (see Alg. 4 (`gauss`), the computation of sines and cosines can be avoided [2, Sect. 1.2.5]).

```
procedure gauss
input $\sigma$
$\phi \leftarrow \mathtt{ran}(0, 2\pi)$
$\Upsilon \leftarrow -\mathtt{log}\ \mathtt{ran}(0, 1)$
$r \leftarrow \sigma\sqrt{2\Upsilon}$
$x \leftarrow r\mathtt{cos}\ \phi$
$y \leftarrow r\mathtt{sin}\ \phi$
output $\{x, y\}$
——
```

Algorithm 4: `gauss`. Two independent Gaussian random numbers obtained by sample transformation.

As we just learned to sample the one-dimensional Gaussian integral, we can also sample the d-dimensional Gaussian integral,

$$1 = \int \ldots \int \underbrace{\mathrm{d}x_1 \ldots \mathrm{d}x_d}_{\mathrm{d}V} \left(\frac{1}{\sqrt{2\pi}}\right)^d \exp\left[-\frac{1}{2}(x_1^2 + \cdots + x_d^2)\right], \tag{13}$$

by running Alg. 4 (`gauss`) $d/2$ times. This obtains $\{x_1, \ldots, x_d\}$. We now substitute

$$\mathrm{d}x_1 \ldots \mathrm{d}x_d = r^{d-1}\,\mathrm{d}r\,\mathrm{d}\Omega$$

to obtain:

$$1 = \left(\frac{1}{\sqrt{2\pi}}\right)^d \overbrace{\int_0^\infty \mathrm{d}r\ r^{d-1} \exp\left(-r^2/2\right)}^{\text{some radial distribution}} \quad \underbrace{\int \mathrm{d}\Omega}_{\text{uniform solid angle}} \quad . \tag{14}$$

The $\{x_1, \ldots, x_N\}$ are samples, and so is $r = \sqrt{x_1^2 + \cdots + x_N^2}$, and so is the solid angle Ω [4]. With little rearrangements of the radius, we either obtain uniformly distributed pebbles within the unit d-dimensional hypersphere (Algorithm 5) or on its surface (Algorithm 6, see also Fig. 6).

The Gaussian is unique in combining into a d-dimensional isotropic integral. The transformation from eq. (13) to eq. (14), that is, Alg. 6 (`direct-surface`), was already known to Maxwell,

[4] not to be confused with the sample space

```
procedure direct-sphere
$\Sigma \leftarrow 0$
for $k = 1, \ldots, d$:
    $x_k \leftarrow$ gauss($\sigma$)
    $\Sigma \leftarrow \Sigma + x_k^2$
$\Upsilon \leftarrow$ ran$(0,1)^{1/d}$
for $k = 1, \ldots, d$:
    $x_k \leftarrow \Upsilon x_k/\sqrt{\Sigma}$
output $\{x_1, \ldots, x_d\}$
——
```

Algorithm 5: `direct-sphere`. Uniform random vector inside the d-dimensional unit sphere. The output is independent of σ.

```
procedure direct-surface
$\sigma \leftarrow 1/\sqrt{d}$
$\Sigma \leftarrow 0$
for $k = 1, \ldots, d$:
    $x_k \leftarrow$ gauss($\sigma$)
    $\Sigma \leftarrow \Sigma + x_k^2$
for $k = 1, \ldots, d$:
    $x_k \leftarrow x_k/\sqrt{\Sigma}$
output $\{x_1, \ldots, x_d\}$
——
```

Algorithm 6: `direct-surface`. Random vector on the surface of the d-dimensional unit sphere. For large d, Σ approaches one (see Fig. 6).

and was the key to his invention of the Maxwell distribution. Later generalized by Boltzmann, it became the core of all of statistical mechanics. So, maybe it was worth looking under the hood of the Gaussian algorithm.

1.3.3 Rejection method

We have a closer look at rejection sampling, after Sec. 1.2.1, but now for a continuum distribution. In preparation of Lecture 2 and following Ref. [5], we consider the Boltzmann distribution of the anharmonic oscillator

$$\int_{-\infty}^{\infty} \mathrm{d}x \underbrace{\exp\left(-\frac{x^2}{2} - \frac{x^4}{4}\right)}_{\pi_{24}(x)}, \tag{15}$$

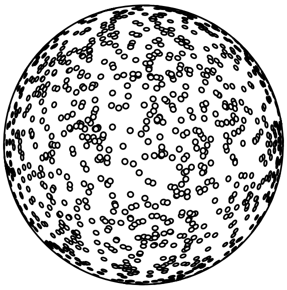

Figure 6: Random samples on the surface of the 3-dimensional sphere, from Alg. 6 (`direct-surface`)

that we need not normalize, and likewise for the Gaussian π_2.

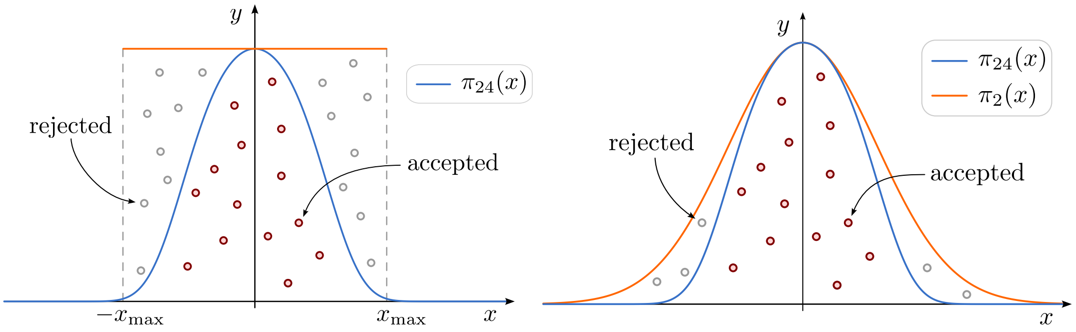

Figure 7: Naive and state-of-the-art algorithms to sample the Boltzmann distribution for the anharmonic oscillator as uniformly distributed pebbles. (a): The naive algorithm introduces a cutoff $x_{\max}$ and discards many pebbles. (b): Sampling pebbles (x, y) below the Gaussian curve and then discarding those samples (x, y) above $\pi_{24}(x)$ is a winning strategy (see eq. (17)

It is straightforward to enclose π_{24} in a rectangular box between $x = \pm x_{\max}$ and to adapt Alg. 2 (`reject-finite`) (see Fig. 7a). But this is wasteful of pebbles and potentially dangerous because of the cutoff. The box in Fig. 7a is nothing but a bounding function that satisfies two conditions: it can be sampled and it dominates the distribution of interest. For the anharmonic oscillator of eq. (15), the Gaussian is an alternative bounding function, so that we can write:

$$\int_{-\infty}^{\infty} \mathrm{d}x \ \exp\left(-\frac{x^2}{2} - \frac{x^4}{4}\right) = \underbrace{\int_{-\infty}^{\infty} \mathrm{d}x \ \exp\left(-\frac{x^2}{2}\right)}_{\text{sample}} \underbrace{\exp\left(-\frac{x^4}{4}\right)}_{\text{accept/reject}} \tag{16}$$

This suggests that to sample the anharmonic oscillator, we may sample a Gaussian, and then accept the pebble x with probability $\exp\left(-x^4/4\right)$ (see Alg. 7 (`direct-anharm`)). But to convince us that this brashly introduced algorithm is actually correct, we sample the Gaussian in x, then spread pebbles out evenly on the y-axis with a $\texttt{ran}\big(0, \exp\left(-x^2/2\right)\big)$. We now have uniform pebbles in the two-dimensional region below the Gaussian. It suffices to reject all pebbles above π_{24}, and this is what Alg. 7 (`direct-anharm`) implements.

To see how to proceed in general, we write:

$$\underbrace{\int_{-\infty}^{\infty} \mathrm{d}x \ \exp\left(-\frac{x^2}{2}\right)}_{\text{sample}} \underbrace{\exp\left(-\frac{x^4}{4}\right)}_{\text{accept/reject}} = \underbrace{\int_{-\infty}^{\infty} \mathrm{d}x \ \exp\left(-\frac{x^2}{2}\right)}_{\text{sample } x} \underbrace{\frac{\exp\left(-\frac{x^2}{2} - \frac{x^4}{4}\right)}{\exp\left(-\frac{x^2}{2}\right)}}_{\text{sample } y \text{, see Fig. 7b}}. \tag{17}$$

In conclusion, to sample a distribution, such as π_{24}, we may divide by and multiply it with another distribution, such as π_2, under the two conditions that we know to sample the latter, and that it is always larger than the former. Remarkably, neither distribution (neither π_2 nor π_{24}) need be normalized. The procedure works if the proportion of pebbles we must discard is not too large.

1.4 Direct sampling: fundamental aspects

Many fundamental aspects of sampling already manifest themselves in the direct-sampling framework, and then translate, *mutatis mutandis*, to the much more complicated Markov chains of subsequent lectures. For example, the strong law of large numbers that we need to understand for direct samples will turn into the famous ergodic theorem for Markov chains. We also discuss

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

Algorithm 7: `direct-anharm`. Sampling π_{24} through the rejection of Gaussians samples from Eq. (17).

importance sampling that permeates all of Monte Carlo and discuss the frequentist interpretation of probabilities at the core of the method.

1.4.1 Laws of large numbers—importance sampling

To discuss the convergence of Markov chains, we consider a frankly difficult sampling problem, the γ integral:

$$I(\gamma) = \int_0^1 \mathrm{d}x \; x^\gamma = \frac{1}{\gamma + 1} \quad \text{for } \gamma > -1 \tag{18}$$

(see [2, Sect. 1.4.2] for the full context). We attempt to compute the integral in a sample space $\Omega^{[0,1]}$, the unit interval between 0 and 1.

$$I(\gamma) = \int_0^1 \mathrm{d}x \; x^\gamma = \int_0^1 \underbrace{(1 \mathrm{d}x \;)}_{x=\texttt{ran}(0,1)} \; \overbrace{x^\gamma}^{\mathcal{O}} \tag{19}$$

As we discussed before, the random variable $\mathcal{O}$ has its own probability distribution:

$$\pi(\mathcal{O}) = (\alpha - 1)\mathcal{O}^{-\alpha}, \tag{20}$$

with $\alpha = 1 - 1/\gamma$. Its mean value of the random variable $\mathcal{O}$ can be equivalently written with $\pi(\mathcal{O})$ or in the original sample space:

$$\langle \mathcal{O} \rangle = (\alpha - 1) \int_1^\infty \mathrm{d}\mathcal{O} \; \mathcal{O}\mathcal{O}^{-\alpha} = \int_0^1 \mathrm{d}x x^\gamma . \tag{21}$$

The same holds for any higher moments.

After these preliminaries, let us now actually compute the γ integral with a running average of a sum of uniform random numbers to the power of γ (see Algorithm 8). This calculation works well for large enough $\gamma \gtrsim -0.5$ but fails for $-1 < \gamma < -0.5$. In the graph of the running average Σ, we notice a sequence of "false plateaux", which seem to indicate convergence, and might prompt us to write research papers. However, in between the plateaux, there are huge jumps caused by particularly large samples and that we should by no means eliminate. Overall, the *strong* law of large numbers applies to the sum of independent random variables with finite mean, that is, to Alg. 8 (`direct-gamma`). It guarantees almost sure convergence of any single trajectory of Σ, which will never again step out of the red box shown in Fig. 8, and which exists for any vertical size ϵ.

Although, in Alg. 8 (`direct-gamma`), there is law and order, that is, almost sure convergence, the wide spread of the random variable $\mathcal{O}$, from ~ 1 to $\sim 1,000.000$, thwarts direct attempts to compute the integral of eq. (18). The founding concept of importance sampling was invented to

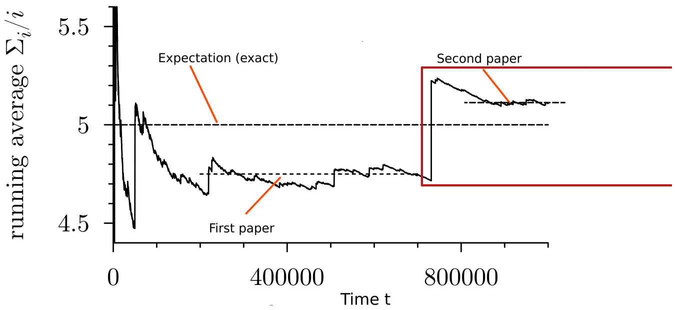

Figure 8: Running average of Alg. 8 (`direct-gamma`) for $\gamma = -0.8$. The strong law of large numbers guarantees the existence of boxes (for any ϵ), which contain the running averages up to infinite times.

```
procedure direct-gamma
$\Sigma \leftarrow 0$
for $i = 1, \ldots, N$:
    $x_i \leftarrow$ ran$(0, 1)$
    $\Sigma \leftarrow \Sigma + x_i^\gamma$ (running average: $\Sigma/i$)
output $\Sigma/N$
——
```

Algorithm 8: `direct-gamma`. Computing the γ-integral in eq. (18) by direct sampling.

contain the spread of $\mathcal{O}$. It consists in writing eq. (22) as

$$I(\gamma) = \int_0^1 \mathrm{d}x \; x^\gamma = \int_0^1 \underbrace{(x^\zeta \mathrm{d}x \;)}_{x=\mathtt{ran}(0,1)^{1/(\zeta+1)}} \overbrace{x^{\gamma-\zeta}}^{\mathcal{O}} \tag{22}$$

If $\gamma \simeq \zeta$, the fluctuations in the random variable are manageable, and the variance of $\mathcal{O}$ remains finite. This is implemented in Algorithm 9, which samples $x_i \sim x^\zeta$, then takes them to the power $\gamma - \zeta$. However, there is an normalization issue, and the program computes $I(\gamma)/I(\zeta)$. Running Algorithm 9 for a sequence of (γ, ζ) values implements importance sampling, and illustrates the concept of thermodynamic integration. Inputting only the value of $I(0) = 1$, and performing a carefully designed sequence of runs of Algorithm 9, we can compute $I(-0.8) \sim 5$ (from the product of the last column in Table 1.4.1), which appeared previously impossible.

```
procedure direct-gamma-zeta
$\Sigma \leftarrow 0$
for $i = 1, \ldots, N$:
    $x_i \leftarrow$ ran$(0, 1)^{1/(\zeta+1)}$   $(\pi(x_i) \propto x_i^\zeta)$
    $\Sigma \leftarrow \Sigma + x_i^{\gamma-\zeta}$
output $\Sigma/N$
——
```

““

Algorithm 9: `direct-gamma-zeta`. Using importance sampling to compute the ratio $I(\gamma)/I(\zeta)$.

γ	$\Sigma/N \pm$ Error	$1/(\gamma+1)$
2.0	0.334 ± 0.003	$0.333\ldots$
1.0	0.501 ± 0.003	0.5
0.0	1.000 ± 0.000	1
-0.2	1.249 ± 0.003	1.25
-0.4	1.682 ± 0.014	$1.666\ldots$
-0.8	3.959 ± 0.110	5.0

Table 2: Output of Alg. 8 (`direct-gamma`) for various values of γ ($N = 10\,000$, standard empirical error shown). The computation for $\gamma = -0.8$ is in trouble.

γ	ζ	Σ/N	$\frac{\zeta+1}{\gamma+1}$
-0.4	0.0	1.685 ± 0.017	1.66
-0.6	-0.4	1.495 ± 0.008	1.5
-0.7	-0.6	1.331 ± 0.004	1.33
-0.8	-0.7	1.508 ± 0.008	1.5

Table 3: Output of Alg. 9 (`direct-gamma-zeta`) with $N = 10\,000$. All pairs $\{\gamma, \zeta\}$ satisfy $2\gamma - \zeta > -1$ so that $\langle \mathcal{O}^2 \rangle < \infty$.

1.4.2 Universal approach to the integrated distribution function

1.4.3 What we learn about π by throwing pebbles

References

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

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

[3] M. Matsumoto and T. Nishimura, “Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator,” *ACM Transactions on Modeling and Computer Simulation*, vol. 8, no. 1, p. 3–30, 1998.

[4] A. J. Walker, “An Efficient Method for Generating Discrete Random Variables with General Distributions,” *ACM Trans. Math. Softw.*, vol. 3, no. 3, pp. 253–256, 1977.

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