

Numerical Methods for Soft Matter: Exercises

Lorenzo Rizzi

1 Exercise 1

1.1 Inversion method

Use the inversion method to design an algorithm that samples random numbers according to the PDF:

$$f(x) = (1-x)a + xb \quad \text{with } x \in [0, 1] \quad (1)$$

with $a, b \geq 0$. Simulate the designed algorithm with $a = 0$ and $b = 7$ and compare the obtained histograms with the analytical expressions

Resolution

We will use the inversion method to sample from the target distribution $f(x)$. But first of all, we need to normalize it:

$$C = \int_0^1 f(x) dx = \int_0^1 [(1-x)a + xb] dx = \int_0^1 (x(b-a) + a) dx = \frac{1}{2}(b-a) + a = \frac{1}{2}(b+a)$$

Finally, the normalized distribution looks like:

$$\rho(x) = \frac{f(x)}{C} = \frac{2(1-x)a + 2bx}{b+a}$$

Now we can compute the cumulative distribution function $F(x)$:

$$F(x) = \frac{1}{C} \int_0^x ds(s(b-a) + a) = \frac{2}{a+b} \left(\frac{1}{2}(b-a)x^2 + ax \right) = \frac{b-a}{b+a}x^2 + \frac{2a}{a+b}x$$

which is injective in $[0, 1]$, as desired from a CDF. To invert it, let $F(x) = y$:

$$(a+b)y = (b-a)x^2 + 2ax$$
$$x^2 + \frac{2a}{b-a}x - \frac{b+a}{b-a}y = 0$$

and:

$$x = \frac{-\frac{2a}{b-a} \pm \sqrt{\frac{4a^2}{(b-a)^2} - 4\frac{b+a}{b-a}}}{2} = \frac{-\frac{2a}{b-a} \pm \sqrt{\frac{4a^2}{(b-a)^2} - 4\frac{b+a}{b-a}}}{2}$$

simplifying further, we finally get:

$$F^{-1}(x) = \frac{\sqrt{a^2(1-x) + b^2x} - a}{b-a}$$

To sample from $\rho(x)$, one just has to extract a random real number $\xi_i \in [0, 1]$ uniformly and perform:

$$a_i = F^{-1}(\xi_i)$$

and a_i will be distributed according to ρ . A snippet of the function used to sample from the target distribution is shown in the following:

```
1     def sampleCustom(n,a,b):
2         #First of all, draw n uniform random number between 0 and 1
3         u = np.random.uniform(0,1,n)
4         #Then, use the inverse CDF to transform the uniform random number
5         z = (np.sqrt(a**2 * (1-u) + b**2 * u) - a) / (b - a)
6         return z
```

The generation of the uniformly distributed numbers is obtained through Numpy's function `np.random.uniform` that should use MT19937 (Mersenne twister algorithm) under the hood. To verify whether our function properly generate numbers distributed according to ρ , we can build a histogram starting from the simulated data and confront it with the actual PDF (Fig.1)

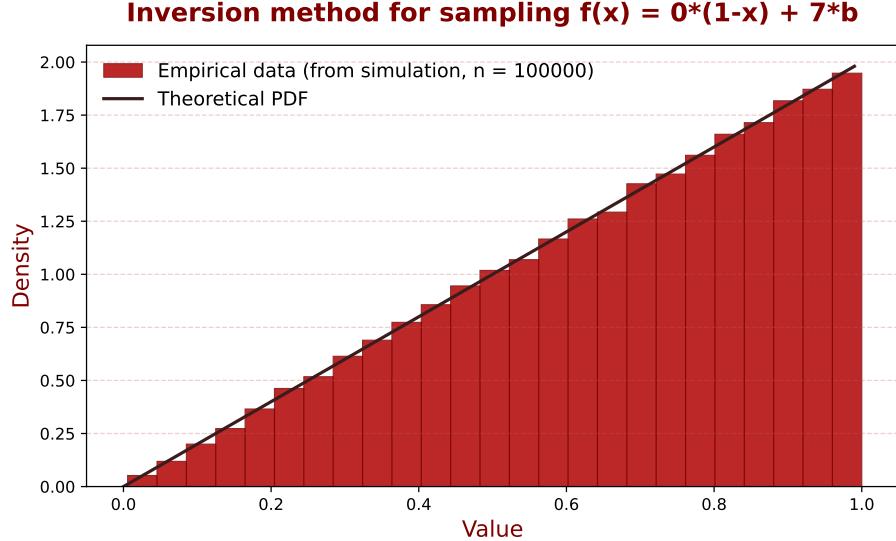


Figure 1: *Empirical samples obtained through the inversion method. Here, we generated $n = 100000$ samples with parameters $a = 0, b = 7$. The black solid line represents the theoretical pdf, which nicely fit the generated dataset*

1.2 Estimating areas by MC

Part 1 Generate random points uniformly distributed within an ellipse of major and minor axes, respectively $a = 5$ and $b = 1$, and compare the analytic value of its area with the Monte Carlo estimate based on the hit-miss method as a function of the number of “throws”

Resolution

We will use the *hit and miss algorithm*. We will generate pairs of points (x, y) uniformly distributed in the rectangle $[-5, 5] \times [-1, 1]$, i.e. $x \sim \mathcal{U}[-5, 5]$, $y \sim \mathcal{U}[-1, 1]$. If $\frac{x^2}{a^2} + \frac{y^2}{b^2} \geq 1$, we discard the pair (x, y) , otherwise we’ll keep it. Results are shown in Fig.2

To estimate the area of the ellipse, we just have to count the number of ”hits”. The estimate will then be:

$$A_n = 4 \frac{\# \text{points in}}{n} ab$$

whereas the actual value of A is $A = \pi ab$. In Fig.2, right panel, we compare the analytic value of the area A with the Monte Carlo estimate based on the hit-miss method as a function of the number of “throws” (n), A_n . For each value of n , we repeat the hit and miss procedure $k = 10$ times and average over the results to get a more robust scaling law. The absolute error is obtained as $\epsilon_n = |A - A_n|$, where A_n is the average over the $k = 10$ repetitions. As expected, the scaling behaviour closely resembles the correct one, i.e.:

$$\epsilon_n \sim \frac{1}{\sqrt{n}}$$

This is due to the central limit theorem.

Part 2 Use the hit-and-miss method to generate points within the region Ω coming from the intersection of the two parabolas $y = x^2$ and $y = 4x - x^2$ and compare the estimate of the area with the analytical expression, again as a function of the number of “throws”

Resolution

The procedure is basically the same as the one in Part 1. The results are shown in Fig.3.

1.3 Sampling by transformation of coordinates

The bilinear function

$$f(x, y) = a_0(1 - x)(1 - y) + a_1x(1 - y) + a_2y(1 - x) + a_3xy$$

interpolates between four values w_i at the four corners of the unitary square $[0, 1]^2$. Normalize the function to get the corresponding PDF $\rho(x, y)$; Write an algorithm that samples random points

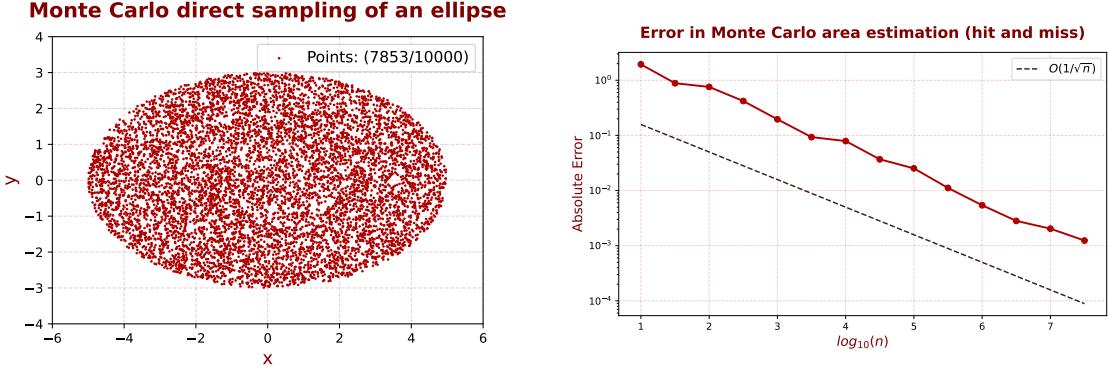


Figure 2: On the left, random points uniformly distributed in an ellipse, using hit and miss method. On the right, scaling law on the absolute error between A_n and A , with n ranging from 10^0 to $10^{7.5}$. The dotted line represents a $O(n^{-1/2})$ scaling behaviour.

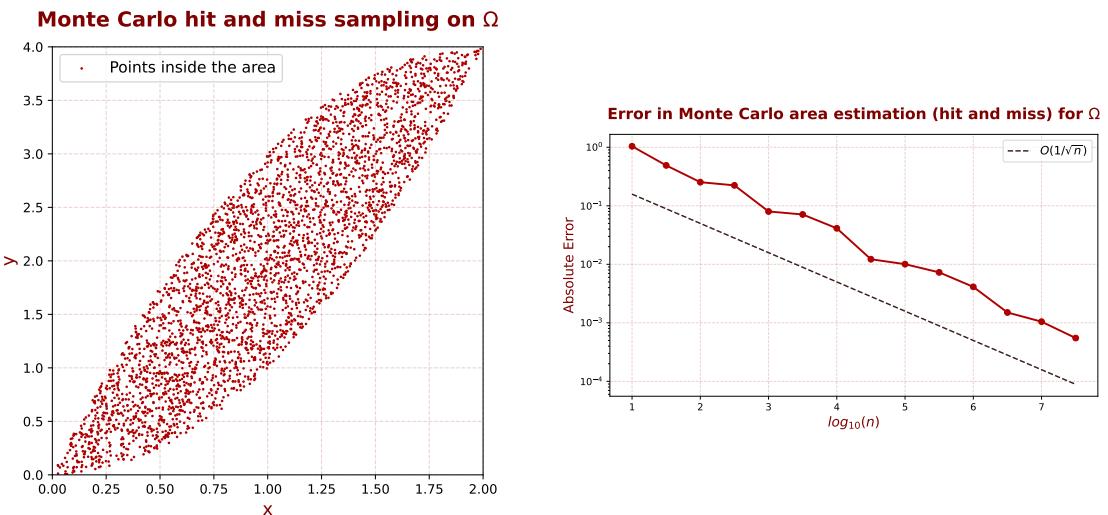


Figure 3: On the left, random points uniformly distributed in the plane region between the curve $y = x^2$ and $y = 4x - x^2$, using hit and miss method. On the right, scaling law on the absolute error between A_n and A , with n ranging from 10^0 to $10^{7.5}$. The dotted line represents a $O(n^{-1/2})$ scaling behaviour.

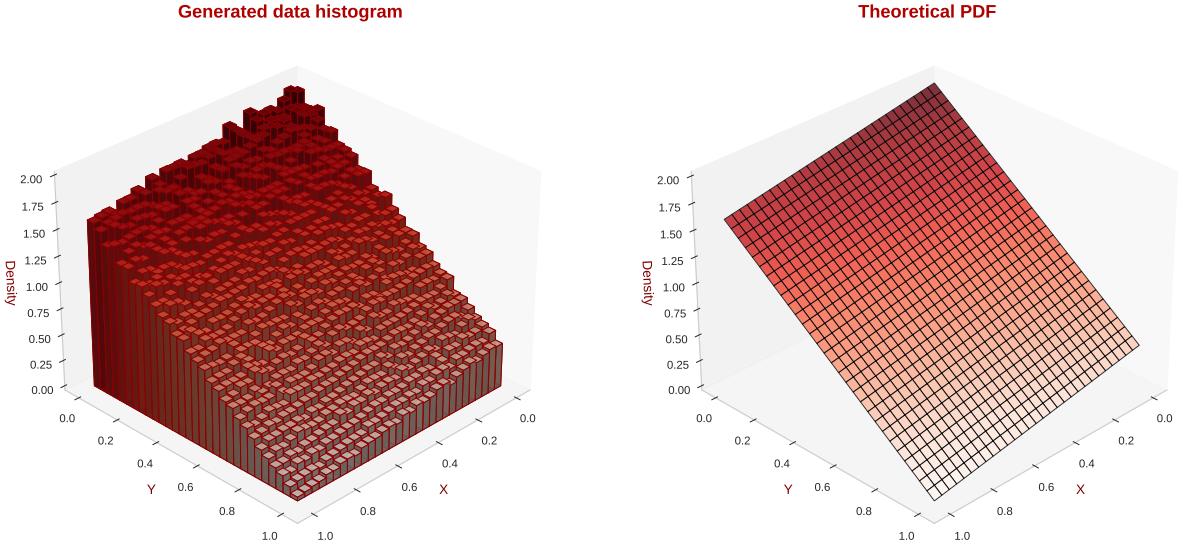


Figure 4: Empirical samples obtained with the marginal and conditional distributions, using inverse method for 1D PDF sampling. We first sample x from its own marginal distribution (a bilinear function) and then we sample y conditioned on x (again, a bilinear function)

(x, y) distributed according to $\rho(x, y)$

Resolution

Let's first normalize the PDF:

$$C = \int_0^1 \int_0^1 dx dy f(x, y) = \int_0^1 \int_0^1 dx dy [a_0(1-x)(1-y) + a_1x(1-y) + a_2y(1-x) + a_3xy] = \frac{1}{4}(a_0 + a_1 + a_2 + a_3)$$

If the PDF were separable, then we could have just sampled from the two factorized 1-point PDFs.

Unfortunately, this is not the case. To sample a pair (x, y) from the target distribution $\rho = \frac{f(x, y)}{C}$, we must first build the marginal distribution on X :

$$\begin{aligned} p_X(x) &= \frac{1}{C} \int_0^1 dy [a_0(1-x)(1-y) + a_1x(1-y) + a_2y(1-x) + a_3xy] \\ &= \frac{1}{2C} [(a_0 + a_2)(1-x) + (a_1 + a_3)x] = \frac{2(a_0 + a_2)(1-x) + 2(a_1 + a_3)x}{a_0 + a_1 + a_2 + a_3} \end{aligned} \quad (2)$$

Note that this equation is fundamentally the same as the one in Eq.1:

$$p_X(x) = \frac{2(a_0 + a_2)}{a_0 + a_1 + a_2 + a_3}(1-x) + \frac{2(a_1 + a_3)}{a_0 + a_1 + a_2 + a_3}x \quad (3)$$

so we can reuse the same algorithm as the one implemented in exercise 1 with $a = 2(a_0 + a_2)$, $b = 2(a_1 + a_3)$. Now we need to sample y from $P(y|x)$, which is given by:

$$p(y|x) = \frac{p_{X,Y}(x, y)}{p_X(x)} = \frac{2a_0(1-x)(1-y) + 2a_1(1-y)x + 2a_2y(1-x) + 2a_3xy}{(a_0 + a_2)(1-x) + (a_1 + a_3)x}$$

which can be rewritten as:

$$p(y|x) = (1-y) \frac{2a_0(1-x) + 2a_1x}{(a_0 + a_2)(1-x) + (a_1 + a_3)x} + y \frac{2a_2(1-x) + 2a_3x}{(a_0 + a_2)(1-x) + (a_1 + a_3)x}$$

which, again, has the same form as in Eq.1 with $a = 2a_0(1-x) + 2a_1x$ and $b = 2a_2(1-x) + 2a_3x$. Using the previous algorithm for both x and y , we can finally sample a pair (x, y) according to $\rho(x, y)$. Results are shown in Fig.4 with $n = 100000$, $a_0 = 10$, $a_1 = 8$, $a_2 = 2$, $a_3 = 0$

1.4 Rejection method

Using a pseudo-random generator that samples numbers according to the normal distribution, apply the rejection method to generate pseudo-random numbers with the distribution

$$\rho(x) = \frac{e^{-x^2/2}}{M(1+x^2)} \quad (4)$$

where M is the normalization constant

Resolution

We can't just use the inverse method here because the inverse cumulative distribution $F^{-1}(x)$ has no analytical form. In fact, the normalization constant M :

$$M = \int dx \frac{e^{-x^2/2}}{1+x^2}$$

can't be solved as a function of standard analytical functions¹. Therefore, we have to rely on Von Neumann's rejection method. In order to implement this algorithm, we first have to build a PDF $g(x)$ according to two criterions:

- Sampling from g should be easy and straightforward
- Given a $c \in \mathbb{R}$, we must have $\rho(x) \leq cg(x), \forall x \in \mathbb{R}$.

Let's choose out of simplicity a gaussian PDF:

$$g(x) \sim N(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-(x-\mu)^2}{2\sigma^2}$$

The target distribution $\rho(x)$ is symmetric around the origin, so let us impose $\mu = 0$. Having a better look at Eq.4, we see that $\rho(x)$ is just a normal distribution with zero mean and unitary variance weighted by a factor $\frac{1}{1+x^2} \geq 0$. This implies that, if we choose $\sigma = 1$ in the definition of $g(x)$, we have naturally that:

$$\begin{aligned} e^{-x^2/2} &\geq \frac{e^{-x^2/2}}{1+x^2} \\ \sqrt{2\pi}g(x) &\geq M\rho(x) \\ \frac{\sqrt{2\pi}}{M}g(x) &\geq \rho(x) \end{aligned} \tag{5}$$

Thus $c = \frac{\sqrt{2\pi}}{M}$. Note that we don't know the value of M , but this is not a problem.

To sample from $\rho(x)$, we now have to:

- Generate x according to $g(x) = \mathcal{N}(0, 1)$
- Generate $\xi \in [0, 1]$. If $\xi \leq \frac{\rho(x)}{cg(x)} = \frac{1}{1+x^2}$, keep the value of x , otherwise discard. Repeat

```

1  def sampleRho(n : int):
2      samples = []
3      while len(samples) < n:      # Keep going until we have n samples
4          x = np.random.normal(0, 1) # Sample from g(x)
5          xi = np.random.uniform(0, 1) # Sample from uniform[0,1]
6          ratio = (1 / (1 + x**2)) # Compute the ratio
7          if xi < ratio:
8              samples.append(x)      #Store the sample
9      return np.array(samples)

```

Results are shown in Fig.5. To compare the histogram with the actual correct PDF, I have estimated M numerically: $M \approx 1.643544$. As it is easy to see, the implemented algorithm correctly generate points from the desired distribution. The generation is supposed to be sufficiently efficient, since the envelope function $cg(x)$ nicely cover the original one.

¹One could rewrite M in terms of the error function, but we'd still have the problem of inverting a function that is both exponential and quadratic in x .

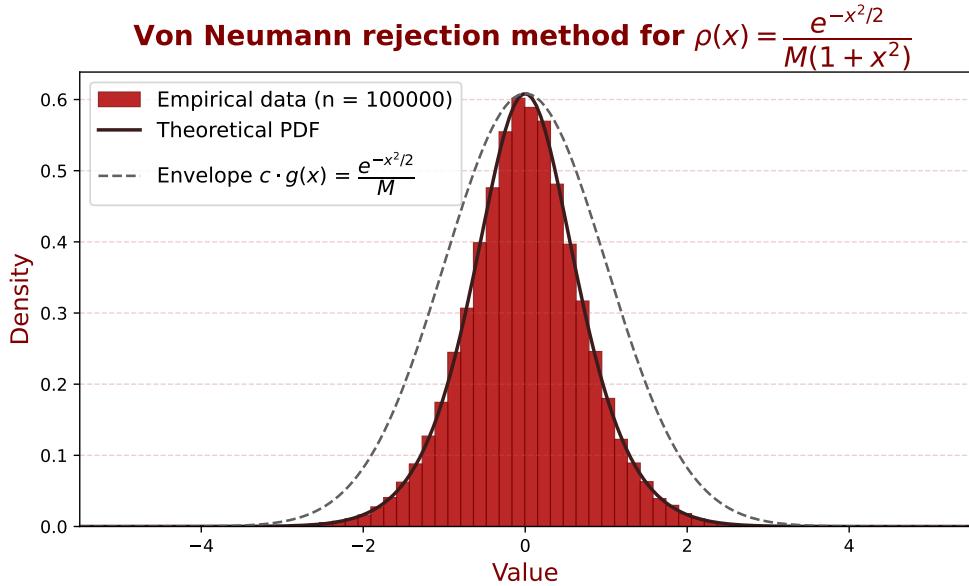


Figure 5: Empirical samples drawn from $\rho(x)$ using Von Neumann rejection method. The envelope function $c \cdot g(x)$ perfectly covers the target distribution, we couldn't have done better ($cg(0) = \rho(0)$)

2 Importance sampling

2.1 Exercise 1

Consider the function $f(x) = 10 \exp(-2|x-5|)$ and suppose we want to compute the average $\langle f(x) \rangle$ with $x \sim \mathcal{U}(0, 10)$. This is equivalent to computing the integral

$$I = \int_0^{10} \exp(-2|x-5|) dx$$

Use the direct method to estimate the integral. Show that with a good choice of the $g(x)$, the importance sampling method provides a reduction of the variance w.r.to the direct method. (Hint: Show first why the normal distribution $g(x) \sim N(5, 1)$ could be a good reweighting function)

Resolution

We want to estimate the integral:

$$I = \int_0^{10} \exp(-2|x-5|) dx$$

This is an easy integral that can be also solved analytically:

$$I = 1 - e^{-10} \approx 0.999955$$

Let's estimate it using direct Monte Carlo sampling. First, we rewrite the integral under a new measure $dP = \frac{1}{10}dx = p(x)dx$ where $p(x)$ is the PDF of an uniform random variable in $[0, 10]$:

$$\begin{aligned} I &= \int_0^{10} \exp(-2|x-5|) dx = 10 \int_0^{10} \exp(-2|x-5|) dP = \\ &= 10 \int_0^{10} \exp(-2|x-5|) p(x) dx = 10 \langle \exp(-2|x-5|) \rangle_{\mathcal{U}} = \\ &= \langle f(x) \rangle_{\mathcal{U}} \end{aligned} \tag{6}$$

where the average of $f(x) = 10 \exp(-2|x-5|)$ is performed sampling x from a $\mathcal{U}[0, 10]$. Thus, to estimate the thermal average $\langle f(x) \rangle_{\mathcal{U}}$ ² we generate $\{x_i\}_{i=1,\dots,n}$ with $x_i \sim \mathcal{U}[0, 10]$ and we compute:

$$I_n = \frac{1}{n} \sum_{i=0}^n f(x_i) \quad (= \bar{f}_n)$$

²"Thermal" here simply means *integral average*, the true statistical one. The whole trick of the direct Monte Carlo method consists in rewriting the initial integral as a statistical average and then approximate it through an arithmetic mean

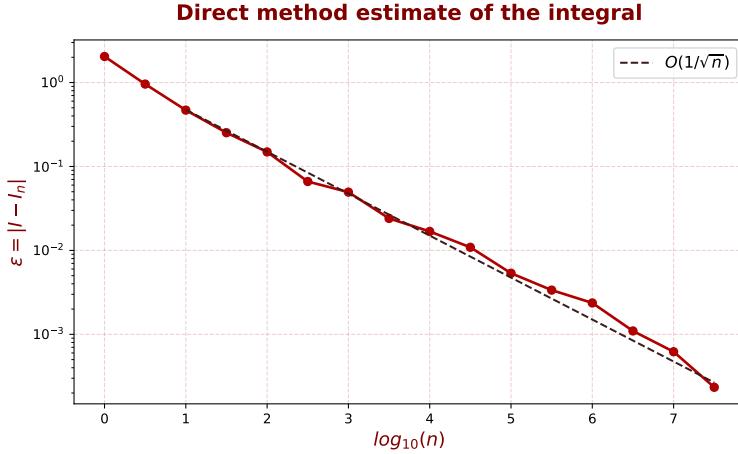


Figure 6: Direct sampling, Monte Carlo integration. Given a value of n , we compute $I_n^{(k)}$ for $k = 1, \dots, 20$. The final dots are given as averages over $I_n^{(k)}$. As n grows, the distribution gets closer and closer to a gaussian one with the right mean value and the error decreases as $O(n^{-1/2})$. The standard deviation σ is the intercept in the semi-logarithmic plot, which, in fact, is remarkably close to 2

We can also write an expression for the associated statistical error. In fact, \bar{f}_n is itself a random variable since every run of the algorithm is stochastic. By the law of large numbers, when $n \rightarrow \infty$ it should precisely converge to $\langle f \rangle$. By the central limit theorem, we know that its probability distribution function (once n is fixed) should get closer and closer to a normal distribution centered around $\langle f \rangle$ and whose variance is σ^2/n , with σ^2 being the variance of $f(x)$ when $x \sim \mathcal{U}[0, 10]$:

$$\sigma^2 = \int_0^{10} (f(x)^2 - \langle f(x) \rangle^2) dx$$

Usually, we don't know how to compute $\langle f(x) \rangle$ (it's part of the problem), so we can rely on the estimator for the variance (which is itself a random variable)

$$S_n^2 = \frac{1}{n} \sum_{i=0}^n (f(x_i)^2 - (I_n)^2) \approx \sigma^2$$

so that:

$$\langle f(x) \rangle_{\mathcal{U}} \approx \bar{f} \pm \frac{S_n}{\sqrt{n}} \Rightarrow I \approx I_n \pm \frac{S_n}{\sqrt{n}} \quad (7)$$

The behaviour of the absolute error $\epsilon = |I_n - I|$ is shown in Fig.6. Actually, since this is a solvable problem, we can compute the σ^2 exactly:

$$\sigma^2 = \int_0^{10} (f(x)^2 - \langle f(x) \rangle^2) dx \approx 4$$

This value can explain the values of the error that we observe in Fig.6. For example, when $n = 10^4$, we expect an error (i.e. deviation from $\langle f(x) \rangle$, the true value) of about $\sigma/\sqrt{n} \approx 2 \cdot 10^{-2}$, in close agreement with the figure. The overall trend decays as $O(n^{-1/2})$, which is expected because of Eq.7.

What can we do to further increase the accuracy of our numerical integration? The decaying law $O(n^{-1/2})$ is intrinsic to the direct Monte Carlo sampling and can't be further improved. However, we can use the importance sampling method to decrease the variance σ^2 . Let $g(x)$ be a PDF from which we can easily sample:

$$\begin{aligned} I &= \int_0^{10} \exp(-2|x-5|) dx = \int_0^{10} \frac{\exp(-2|x-5|)}{g(x)} g(x) dx = \int_0^{10} \frac{\exp(-2|x-5|)}{g(x)} dP_g = \\ &= \left\langle \frac{\exp(-2|x-5|)}{g(x)} \right\rangle_g = \left\langle \rho(x) \frac{f(x)}{g(x)} \right\rangle_g = \langle F(x) \rangle_g \end{aligned} \quad (8)$$

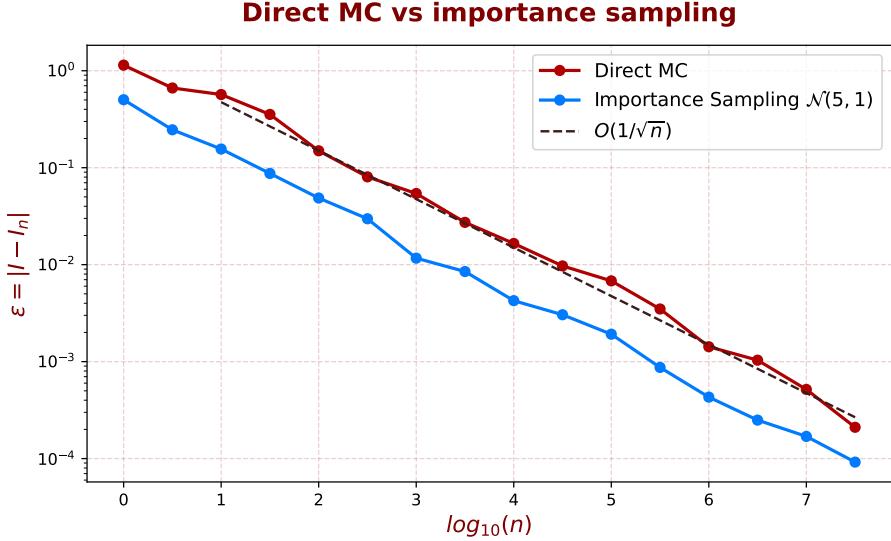


Figure 7: Comparison between direct MC sampling and importance sampling, using $g \sim \mathcal{N}(5, 1)$.

where $dP_g = g(x)dx$ is the probability that the random variable whose PDF is $g(x)$ belongs to the interval $[x, x + dx]$, $\langle \cdot \rangle_g$ is the average according to the distribution g and

$$F(x) = \rho(x) \frac{f(x)}{g(x)} = \frac{\exp(-2|x-5|)}{g(x)}$$

How do we choose g ? The idea is to reduce as much as possible the variance of the resulting distribution. We'd like $g(x)$ to be as close as possible to $f\rho$, while still being able to sample from it. We thus select a normal distribution $\mathcal{N}(\mu, \sigma^2)$ trying to align the mean value μ to the actual maximum of $f(x)$, therefore we choose $\mu = 5$. As for the variance, we will set $\sigma = 1$. Now that we have g , we simply have to sample from it $x_i \sim \mathcal{N}(5, 1)$ and average over the n values of $I_n = \frac{1}{n} \sum_i^n F(x_i) = \frac{1}{n} \sum_i^n \frac{\exp(-2|x_i-5|)}{g(x)}$. The results are shown in Fig. 7. As expected, the slope of the two curves look almost identical: we can't do anything to improve the scaling law $\epsilon \sim (n^{-1/2})$ since this is a fundamental property of the Monte Carlo integration. However, the implementation of importance sampling lead us the a (constant) reduction of the variance $\sigma_g^2(F)$, which decrease the error ϵ at a fixed value of n of about 75%

2.2 Exercise 2

Estimate the integral:

$$I = \int_0^{\pi/2} \cos^2(x) dx \quad (9)$$

by first using the direct method, then using the importance sampling technique with $g(x)$ proportional to $1 + bx^2 + cx^4$. In the second case determine the optimal values of the parameters b and c to generate samples according to $g(x)$ and establish the number of iterations needed to get an accuracy of %1

Resolution

The integral I is actually analytically solvable and its exact value is:

$$I = \frac{\pi}{4} \approx 0.7854$$

Let's first try with the direct MC method. To do so, let's rewrite I as an (uniform) average that we can estimate through repeated sampling:

$$I = \int_0^{\pi/2} \cos^2(x) dx = \frac{\pi}{2} \int_0^{\pi/2} \cos^2(x) dx = \frac{\pi}{2} \langle \cos^2(x) \rangle_U$$

where $\langle \cos^2(x) \rangle_U$ is the average over a uniform distribution $U[0, \pi/2]$. Given n iterations of the direct MC algorithm, the estimate of the integral will be:

$$I_n = \frac{\pi}{2n} \sum_i^n \cos^2(x_i) \quad \text{where } x \sim U[0, \pi/2]$$

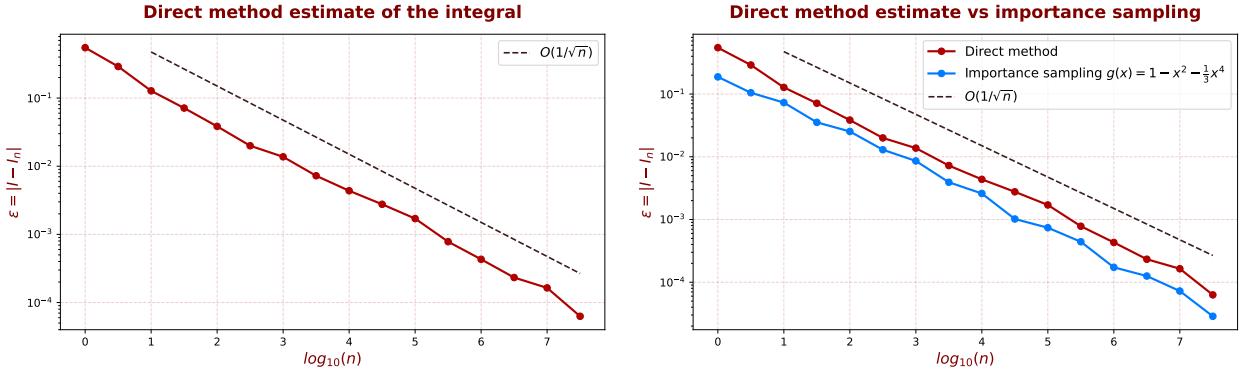


Figure 8: (Left) Direct MC integration for $\int \cos^2(x)dx$. Each point at fixed n is obtained over an average of $k = 20$ independent realizations (Right) Comparison between direct method and importance sampling. The slope of the two curves is the same, as it should be, but the intercept is different. In particular, we can observe that the error (standard deviation) is indeed reduced by about 50% when we sample according to $g(x)$.

As we did before, we are going to run our algorithm for several value of n to study the scaling behaviour of the estimate (which should converge, hopefully, to the correct value as n gets larger and larger). Results are shown in Fig.8 and are in agreement with our expectations, i.e. the error converges to 0 according to $O(n^{-1/2})$. The variance on $f(x) = \cos^2(x)$ given $x \sim \mathcal{U}[0, \pi/2]$ can be computed analytically and is equal to:

$$\begin{aligned} \sigma_{\mathcal{U}}^2(f) &= \frac{2}{\pi} \int_0^{\pi/2} \left(\cos^2(x) - \frac{\pi}{4} \right)^2 dx = \frac{2}{\pi} \int_0^{\pi/2} \cos^4(x) dx - \frac{2\pi}{\pi} \int_0^{\pi/2} \cos^2(x) dx + \frac{2\pi^3}{\pi 32} = \\ &= \frac{3}{8} - \frac{\pi}{4} + \frac{\pi^2}{16} \approx 0.2 \Rightarrow \sigma_{\mathcal{U}}(f) \approx 0.4 \end{aligned} \quad (10)$$

which is indeed the intercept in the left panel of Fig.8. To reduce the variance, let's perform importance sampling with $g(x)$:

$$g(x) = \frac{1}{Z}(1 + bx^2 + cx^4)$$

with

$$Z = \int_0^{\pi/2} (1 + bx^2 + cx^4) dx = \frac{\pi}{2} + b \frac{\pi^3}{24} + c \frac{\pi^5}{160}$$

The variance of $F(x) = f(x) \frac{\rho(x)}{g(x)}$ with respect to the distribution $g(x)$ is:

$$\sigma_g^2 = \int (F(x) - \langle F(x) \rangle_g)^2 g(x) dx$$

and we'd like to reduce it as much as possible. Ideally, we'd like $g(x) \propto f(x)\rho(x) \propto f(x)$ since ρ is uniform. Taylor expanding $f(x) = \cos^2(x)$ around $x = 0$,

$$\cos^2(x) = 1 - x^2 + \frac{1}{3}x^4 + O(x^6)$$

so that a reasonable choice for b, c would be:

$$b = -1, c = \frac{1}{3}$$

so, finally:

$$g(x) = \frac{1 - x^2 + \frac{1}{3}x^4}{\frac{\pi}{2} - \frac{\pi^3}{24} + \frac{\pi^5}{480}}$$

Practically speaking, we are going to sample n independent samples $\{x_i\}_{i=1,\dots,n}$ from $g(x)$ and compute:

$$I_n = \frac{1}{n} \sum_{i=0}^n \frac{\cos^2(x_i)}{g(x_i)}$$

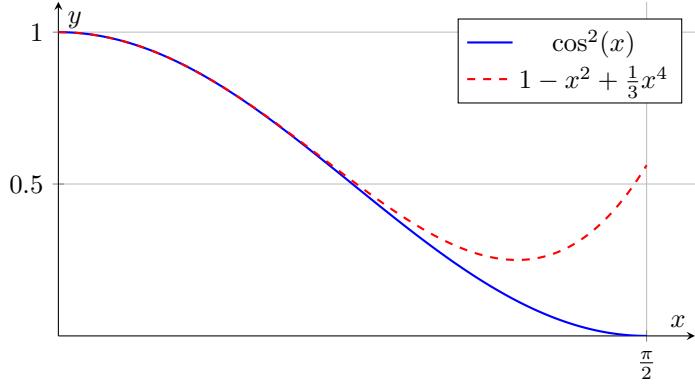


Figure 9: The blue continuous line is the function we want to integrate, $f(x) = \cos^2(x)$. The red dotted line is the proposal $g(x) = 1 + bx^2 + cx^4$ (unnormalized) where $b = -1, c = \frac{1}{3}$

sampling from $g(x)$ is not straightforward since its inverse cumulative is not easy to isolate. However, the function is well behaved on $[0, \pi/2]$ (Fig. 9), so we can use Von Neumann rejection method using $y_M = 1, y_m = 0$. Results are shown in the right panel of Fig. 8, where the direct method and the importance sampling implementation with $g(x)$ are shown in the same plot. To achieve an accuracy of about $1\% = 0.01$, one should set n equal to about 10^3 when using importance sampling, $n \approx 10^{3.5} \approx 3000$ with the direct method.

3 Lecture 4: Markov chains

3.1 Exercise 1

The state probability vector of a Markov chain μ_n satisfies the recurrence relation:

$$\mu_n = \mu_{n-1} \mathbf{P} \quad (11)$$

Show that this equation is equivalent to writing

$$\mu_n(i) = \left(1 - \sum_{j \neq i} p_{ji}\right) \mu_{n-1}(i) + \sum_{j \neq i} p_{i,j} \mu_{n-1}(j)$$

Explain the meaning of this equation in terms of a dynamical equation for the state vector.

Resolution

Let's start by rewriting Eq.11 by elements:

$$\mu_n(i) = \sum_{j \in S} \mu_{n-1}(j) p_{ji} = \sum_{j \neq i} \mu_{n-1}(j) p_{ji} + \mu_{n-1}(i) p_{ii}$$

To ensure probability conservation, \mathbf{P} must be stochastic, i.e. normalized across rows:

$$\sum_j p_{i,j} = 1 = p_{ii} + \sum_{j \neq i} p_{i,j}$$

substituting p_{ii} into the previous formulae, we get:

$$\begin{aligned} \mu_n(i) &= \sum_{j \in S} \mu_{n-1}(j) p_{ji} = \sum_{j \neq i} \mu_{n-1}(j) p_{ji} + \mu_{n-1}(i) \left(1 - \sum_{j \neq i} p_{i,j}\right) \\ \mu_n(i) &= \mu_{n-1}(i) \left(1 - \sum_{j \neq i} p_{i,j}\right) + \sum_{j \neq i} \mu_{n-1}(j) p_{ji} \end{aligned}$$

The first term can be interpreted as follows. Since p_{ij} represents the probability of transitioning from state i to state j , the quantity $\sum_{j \neq i} p_{i,j}$ is just the probability (in the time quanta $\Delta t = 1$) of transitioning to any state j different than the starting state i . Therefore the first term simply represents the contribution to the probability of being in state i at time $n - 1$ ($\mu_n(i)$) due to just standing there, i.e. no transitions has occurred from $n - 1$ to n . The second term, of course, represents the contribution due to the flux coming from all other states different from i itself. Each state $j \neq i$ will contribute with a factor $p_{ji} \mu_{n-1}(j)$, i.e. the probability of being in j at time $n - 1$ times the probability of transitioning from j to i .

3.2 Exercise 2

Classify the following Markov chain \mathbf{P} in terms of its digraph (to be drawn), irreducibility, and aperiodicity

$$\mathbf{P} = \begin{pmatrix} 0.3 & 0.7 & 0 \\ 0.2 & 0.4 & 0.4 \\ 0 & 0.6 & 0.4 \end{pmatrix} \quad (12)$$

Resolution

The digraph for this Markov process is drawn in Fig.10.

- The Markov chain \mathbf{P} is homogeneous (it does not depend on time)
- The Markov chain \mathbf{P} is irreducible, since we can reach all of the three states starting from an arbitrary one. State 2 and 1 are directly connected and states 3, 1 can be joined together passing by 2.
- The Markov chain \mathbf{P} is aperiodic. In fact, starting from each state at time $t = 0$, then $\tau = 1$ is always a valid return time thanks to the presence of the non-zero loops. Therefore, the period for each state (the gcd between all return times) is simply 1 and the chain is aperiodic. Since \mathbf{P} is also irreducible, then it is also regular

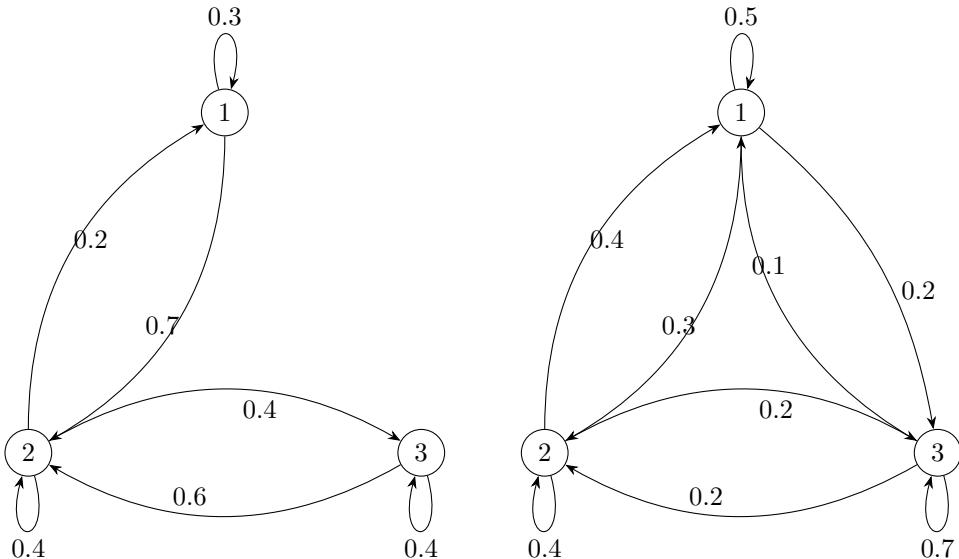


Figure 10: *Markov chain as digraphs, exercises 1 and 2*

3.3 Exercise 3

Find the fixed points of the following Markov chain and determine if they are unique.

$$\mathbf{P} = \begin{pmatrix} 0.5 & 0.3 & 0.2 \\ 0.4 & 0.4 & 0.2 \\ 0.1 & 0.2 & 0.7 \end{pmatrix} \quad (13)$$

The Markov chain (right panel of Fig.10) is homogeneous and irreducible, since any state is reachable from any other states. Again, thanks to the non-zero loops, the chain is also aperiodic, thus it's regular

If a Markov chain is regular, then we know there exists a unique invariant distribution $\vec{\pi}$ to which any initial state converges. It can be found by imposing:

$$\vec{\pi} = \vec{\pi}\mathbf{P}$$

$$(\pi_1 \ \pi_2 \ \pi_3) = (\pi_1 \ \pi_2 \ \pi_3) \begin{pmatrix} 0.5 & 0.3 & 0.2 \\ 0.4 & 0.4 & 0.2 \\ 0.1 & 0.2 & 0.7 \end{pmatrix} \quad (14)$$

$$(\pi_1 \ \pi_2 \ \pi_3) = (0.5\pi_1 + 0.4\pi_2 + 0.1\pi_3 \quad 0.3\pi_1 + 0.4\pi_2 + 0.2\pi_3 \quad 0.2\pi_1 + 0.2\pi_2 + 0.7\pi_3) \quad (15)$$

$$\begin{cases} -0.5\pi_1 + 0.4\pi_2 + 0.1\pi_3 = 0 \\ 0.3\pi_1 - 0.6\pi_2 + 0.2\pi_3 = 0 \\ 0.2\pi_1 + 0.2\pi_2 - 0.3\pi_3 = 0 \end{cases} \quad (16)$$

The normalized solution is:

$$\pi_1 = \frac{14}{45} \quad \pi_2 = \frac{13}{45} \quad \pi_3 = \frac{18}{45}$$

3.4 Exercise 4

Determine if the following Markov chain \mathbf{P} is irreducible and aperiodic, and find its period if it is not aperiodic:

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0.5 & 0 & 0.5 \end{pmatrix} \quad (17)$$

Resolution

The digraph is represented in Fig.11. The markov chain is irreducible: it's possible to go from any node to any other nodes. Regarding the period, let's take as an example the state 1. Starting from there, the possible return times are 2 ($1 \rightarrow 2 \rightarrow 1$), 4 ($1 \rightarrow 2 \rightarrow 1 \rightarrow 2 \rightarrow 1$), 5 ($1 \rightarrow 2 \rightarrow 3 \rightarrow$

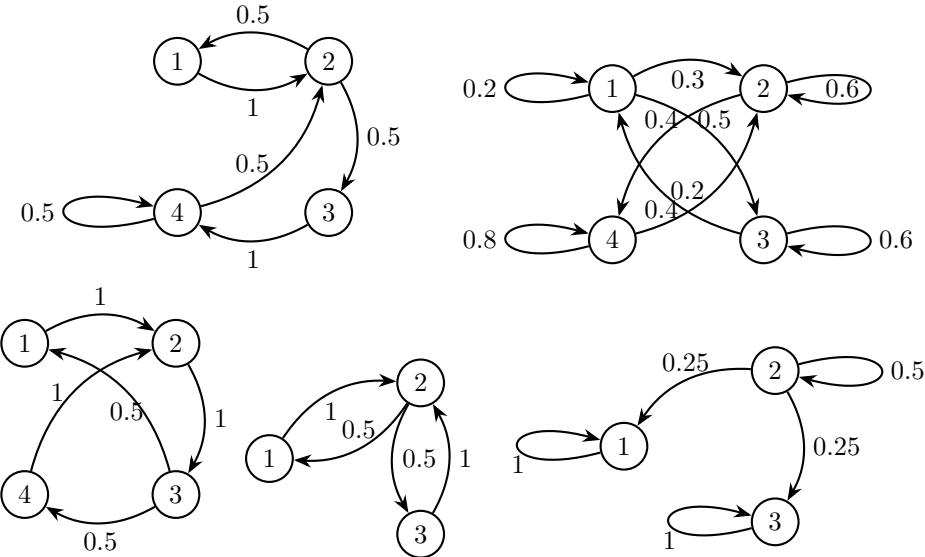


Figure 11: (Top Left) Markov chain digraph for exercise 4 (Top Right) Markov chain digraph for exercise 5 (Bottom Left) Markov chain digraph for exercise 6 (Bottom center) Markov chain digraph for exercise 7 (Bottom right) Markov chain digraph for exercise 8

$4 \rightarrow 2 \rightarrow 1$), thus the gcd is 1. Node 4 has for sure period 1 thanks to the loop. Node 3 has return times $3(3 \rightarrow 4 \rightarrow 2 \rightarrow 3)$, $4(3 \rightarrow 4 \rightarrow 4 \rightarrow 2 \rightarrow 3)$ thanks again to the loop, so the gcd is 1 again. The same logic goes on for node 2, since we can travel from node 2 to node 4 and loop over it an arbitrary amount of time. Therefore, all nodes have period 1 and the Markov chain is aperiodic

3.5 Exercise 5

Determine if the following Markov chain \mathbf{P} is irreducible and aperiodic, and find its period if it is not aperiodic:

$$\mathbf{P} = \begin{pmatrix} 0.2 & 0.3 & 0.5 & 0 \\ 0 & 0.6 & 0 & 0.4 \\ 0.4 & 0 & 0.6 & 0 \\ 0 & 0.2 & 0 & 0.8 \end{pmatrix} \quad (18)$$

Resolution

The digraph is represented in Fig.11. The markov chain isn't irreducible: in fact, you can't go from node 2 to node 1 or node 3

Regarding the period, the presence of self-loops implies that each node has period 1, so the Markov chain as a whole is aperiodic.

3.6 Exercise 6

Draw the digraph corresponding to the following stochastic matrix:

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0.5 & 0 & 0 & 0.5 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad (19)$$

Is the corresponding Markov chain irreducible? Is periodic? If yes, compute the corresponding period

Resolution

The digraph is represented in Fig.11. It is irreducible, since we can reach any states starting from any arbitrary initial states. Regarding the period,

- Starting from state 1, the Markov chain necessarily continues to state 2 and then 3. From node 3, we can return to state 1 (return time of 3) or continue to state 4. From node 4, the only way out is to transition to node 2 and again 3, for a total of 3 more steps. Hence, the period of node 1 is 3

- Since the chain is irreducible, all states will have period 3

In conclusion, this Markov chain is irreducible but periodic with period 3.

3.7 Exercise 7

Determine if the following Markov chain is irreducible and aperiodic, and find its period if it is not aperiodic. Show it by using both the digraph and the matrix.

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & 0 \\ 0.5 & 0 & 0.5 \\ 0 & 1 & 0 \end{pmatrix} \quad (20)$$

The Markov chain is irreducible, any state is reachable starting from any other state. Regarding the period,

- Let's start in node 1. We are forced to transition to node 2. From here, we can return in node 1 (return time of 2) or migrate to node 3. From node 3, we must return to node 2 and if we want to get back in node 1, we would do so with a return time of 4. It's easy to show that the all return times of state 1 must be even, so the its period is 2.
- Since the chain is irreducible, all states will have period 2. Therefore the chain as a whole is not aperiodic (and has period 2).

We can prove the same result using the matrix representation of the Markov chain. In fact, it's easy to show that:

$$\begin{pmatrix} 0 & 1 & 0 \\ 0.5 & 0 & 0.5 \\ 0 & 1 & 0 \end{pmatrix}^{2n+1} = \begin{pmatrix} 0 & 1 & 0 \\ 0.5 & 0 & 0.5 \\ 0 & 1 & 0 \end{pmatrix} \quad (21)$$

thus $(\mathbf{P}^{2n+1})_{ii} = 0, \forall i$, meaning that the probability to start from i and be back in i at time $t = 2n + 1$ is zero if t is odd. Therefore, return times must be even and the period is 2.

3.8 Exercise 8

Draw the digraph related to the following Markov Matrix

$$\mathbf{P} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ 0 & 0 & 1 \end{pmatrix} \quad (22)$$

Show that this Markov chain has more than one stationary distribution and find to which limiting matrix of \mathbf{P}^n as $n \rightarrow \infty$

Resolution

The digraph is shown in Fig.11. It's quite easy to verify that 1 is an absorbing state since $p_{1,1} = 1$, therefore the chain is not irreducible (it's not possible to start in 1 and end anywhere else). The same reasoning goes for node 3.

Since all three nodes have non-zero self-loops, then the Markov chain is aperiodic, but since it's not irreducible, we can't claim it is also regular. This means that we don't have any guarantee on the uniqueness and existence of the invariant distribution $\vec{\pi}$. Solving for $\vec{\pi}$ as we did in Exercise 3, we obtain the linear system:

$$\begin{cases} 0.25\pi_2 = 0 \\ -0.5\pi_2 = 0 \\ -0.75\pi_2 = 0 \end{cases} \quad (23)$$

and any generic normalized state $\vec{\pi} = (p, 0, 1-p)$ is a valid invariant distribution. This is reasonable, since states 1 and 3 are absorbing: if $\vec{\mu}_0 = \delta_{n,1}$, the system will converge to $\vec{\pi} = \delta_{n,1}$ (it will just stay there), and the same goes when $\vec{\mu}_0 = \delta_{n,3}$ that will converge to itself. To better understand this behaviour, let's eigen decompose the matrix \mathbf{P} :

$$\mathbf{P} = \mathbf{Q}\boldsymbol{\Pi}\mathbf{Q}^{-1} = \begin{pmatrix} 0 & -1 & 2 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0.5 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -0.5 & 1 & -0.5 \\ 0 & 0 & 1 \\ -0.5 & 0 & 0.5 \end{pmatrix} \quad (24)$$

And, as expected, the eigenvalue $\lambda = 1$, the one defining the invariant distribution, is degenerate. Now since $\mathbf{P}^n = \mathbf{Q}\Pi^n\mathbf{Q}^{-1}$ and $(0.5)^\infty \rightarrow 0$, then:

$$\vec{\pi} = \lim_{n \rightarrow \infty} \vec{\mu}_0 \mathbf{P}^n = \vec{\mu}_0 \mathbf{Q} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{Q}^{-1} = \vec{\mu}_0 \begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 0 & 0.5 \\ 0 & 0 & 1 \end{pmatrix} = (p_1^{(0)} + 0.5p_2^{(0)}, 0, 0.5p_2^{(0)} + p_3^{(0)}) \quad (25)$$

which is precisely what we expect (up to a normalization factor). The stationary value of π_2 is 0, since once state 2 is exited it is never to be returned. All the probability will be split between the two absorbing states depending on the initial condition. In fact, rewriting $p_2^{(0)} = 1 - p_1^{(0)} - p_3^{(0)}$, we obtain:

$$\begin{cases} \pi_1 = \frac{1}{2}(1 + p_1^{(0)} - p_3^{(0)}) \\ \pi_2 = 0 \\ \pi_3 = \frac{1}{2}(1 + p_3^{(0)} - p_1^{(0)}) \end{cases} \quad (26)$$

3.9 Exercise 9

Find the matrix of a Markov chain describing a biased random walk in two dimensions ($d = 2$), where the bias is along the positive x-axis, and its stationary distribution.

Resolution

We are going to assume that the random walker makes a move every Δt seconds. At each step, the walker independently moves both in the horizontal and vertical direction by a discrete amount Δr , so that the topology is effectively \mathbb{Z} and we can label each lattice point with an integer pair (i, j) (and two consecutive points are a distance Δr away from each other). The walker has no bias in the vertical direction, so it can go up or down with the same probability $1/2$. At variance, we must impose a bias in the horizontal direction: at each Δt , the walker will move to the right with probability p and to the left with probability $1 - p$.

We can describe this process through a discrete-time discrete-state space Markov chain. Since the space state is infinite, there's no finite matrix representation of the process, however we can at least write the transition rates $p_{(ij),(i',j')}$, i.e. the probability that the walker will transition from (i, j) to (i', j') in Δt time units. Imposing the bias condition,

$$p_{(ij),(i',j')} = \begin{cases} \frac{1}{2}p & \text{if } (i', j') = (i + 1, j \pm 1) \\ \frac{1}{2}(1 - p) & \text{if } (i', j') = (i - 1, j \pm 1) \end{cases} \quad (27)$$

Because at each step the horizontal and vertical move are statistically independent, we can consider this motion as two separate one-dimensional random walkers. When the number of steps N grows to infinity (or when $\Delta r, \Delta t \rightarrow 0$), we can invoke the central limit theorem and claim that:

$$\begin{aligned} p(x) &\approx \frac{1}{\sigma_x \sqrt{2\pi N}} \exp\left(-\frac{(x - \mu_x)^2}{2N\sigma_x^2}\right) \\ p(y) &\approx \frac{1}{\sigma_y \sqrt{2\pi N}} \exp\left(-\frac{(y - \mu_y)^2}{2N\sigma_y^2}\right) \end{aligned} \quad (28)$$

where x, y are integer value. The statistical quantities $\mu_x, \mu_y, \sigma_x, \sigma_y$ are related to a single stochastic step in the x or y direction:

$$\begin{aligned} \mu_y &= \langle y \rangle = \frac{1}{2}(\Delta r) + \frac{1}{2}(-\Delta r) = 0 \\ \sigma_y^2 &= \langle y^2 \rangle - \langle y \rangle^2 = \frac{1}{2}(\Delta r)^2 + \frac{1}{2}(-\Delta r)^2 = \Delta r^2 \\ \mu_x &= \langle x \rangle = p(\Delta r) + q(-\Delta r) = (p - q) = (2p - 1)\Delta r \\ \sigma_x^2 &= \langle x^2 \rangle - \langle x \rangle^2 = p(\Delta r)^2 + q(-\Delta r)^2 - (p - q)^2 \Delta r^2 = (1 - (2p - 1)^2)\Delta r^2 \end{aligned} \quad (29)$$

Since the state-space is infinite, there's no proper stationary distribution other than the infinitesimal distribution across all \mathbb{Z} . To have a sensible invariant distribution, let's imagine that the random walker is constrained by a square centered in the origin, i.e. $p_{ij} \neq 0$ if $-L/2 < x < L/2, -L/2 < y < L/2$, ($N_x, N_y \in \mathbb{N}$) and is 0 elsewhere. We can't use the central limit theorem anymore because now the steps may be correlated (if the walker is close to the border, there are

some forbidden moves). Fortunately, in the continuum limit, this is a classical diffusion problem described by the Fokker-Planck:

$$\frac{\partial p(x, y, t)}{\partial t} = -v \frac{\partial p}{\partial x} + D \left(\frac{\partial^2 p(x, y, t)}{\partial x^2} + \frac{\partial^2 p(x, y, t)}{\partial y^2} \right) \quad (30)$$

With the proper reflective boundary conditions. The stationary (i.e. invariant) distribution satisfies:

$$0 = -v \frac{\partial p}{\partial x} + D \left(\frac{\partial^2 p(x, y, t)}{\partial x^2} + \frac{\partial^2 p(x, y, t)}{\partial y^2} \right) \quad (31)$$

This is solved by:

$$p(x, y)^* = A \exp\left(\frac{v}{D}x\right) \quad (x, y) \in [-L/2, L/2]^2 \quad (32)$$

As expected, there is no dependence on y . Along the horizontal direction, the probability distribution (in the continuous limit) is exponential and peaks around $x = L/2$, as it should since the drift term will bias towards the border the walker. If $v = 0$, i.e. $p = \frac{1}{2}$, then we obtain back the uniform distribution.

4 MC simulation of a 2D Ising model by the Metropolis algorithm

Write a program that simulates a 2D Ising model with periodic boundary conditions by using the Metropolis acceptance matrix a_{ij} and the matrix Γ_{ij} based on the local spin-flip (Glauber) proposed move

Resolution

The code developed to run the MCMC simulation can be found in [this repository](#). A snapshot of the C++ function that specifically performs a Monte Carlo step is shown here for the sake of exemplification:

```

1   void MCMCEngine::MetropolisHastings() {
2       std::uniform_int_distribution<int> dist(0, m_N - 1);
3       int index = dist(m_rng);
4       int L = static_cast<int>(std::sqrt(m_N));
5       int row = index / L;
6       int col = index % L;
7
8       // extract the spin value
9       int spin = m_spinConfig->getSpins()[index] ? 1 : -1;
10      int neighbourSum = 0;
11      std::array<int, 4> neighbours = m_spinConfig->
12          getNeighbourhoodSpins(row, col);
13      for (int i = 0; i < 4; ++i) {
14          neighbourSum += neighbours[i];
15      }
16      double exponentialFactor = std::exp(- 2* double(spin) *
17          static_cast<double>(neighbourSum) / m_spinConfig->
18          getTemperature() + 2 * m_spinConfig->getMagneticField() *
19          double(spin) / m_spinConfig->getTemperature());
20      // Metropolis-Hastings criterion: accept if ...
21      if (exponentialFactor > 1.0 || std::uniform_real_distribution<
22          double>(0.0, 1.0)(m_rng) < exponentialFactor) {
23          m_spinConfig->getSpins()[index] = !m_spinConfig->getSpins()
24              [index];
25      }
26  }
```

4.1 Ising model at various temperatures: qualitative analysis

By assuming $k_B = 1$ and $J = 1$, we simulate the 2D Ising model for different values of temperature and three different values of L (the linear size of the lattice). In this specific case, we will choose: $L = 100, 200, 500$ and $T = 1, 2.26, 4$ (assuming $k_B = 1$ and $J = 1$, so temperatures are dimensionless). A snapshot for each combination of size and temperature is shown in Fig. 12, Fig. 13, and Fig. 14.

When $T = 4$, the temperature is well above the critical value $T_c = \frac{2}{\ln(1+\sqrt{2})} \approx 2.26$ (Onsager), and the system should always relax to a disordered phase where the average magnetization is zero (at least in the thermodynamic limit), regardless of the initial configuration. This is precisely what we observe running our Metropolis algorithm: when $T = 4$, the system looks fuzzy (left panels of Fig. 12, Fig. 13, Fig. 14) and the spins are oriented up or down in a seemingly equiprobable fashion. Macroscopically, the system lies in a disordered phase, where the entropic contribution prevails over the energy term.

When $T = 1$, the temperature is well below the critical point, and the system should display an ordered ferromagnetic phase according to our understanding of the Ising model. In the thermodynamic limit, two equilibrium states appear with opposite average magnetization, $m_1(T) = -m_2(T)$. In this specific case, the initial condition will determine to which of these equilibrium ordered phases³ the system will converge, depending on the initial percentage of up spins versus down

³This is not strictly true, especially when the size of the system is not large enough, the algorithm will explore

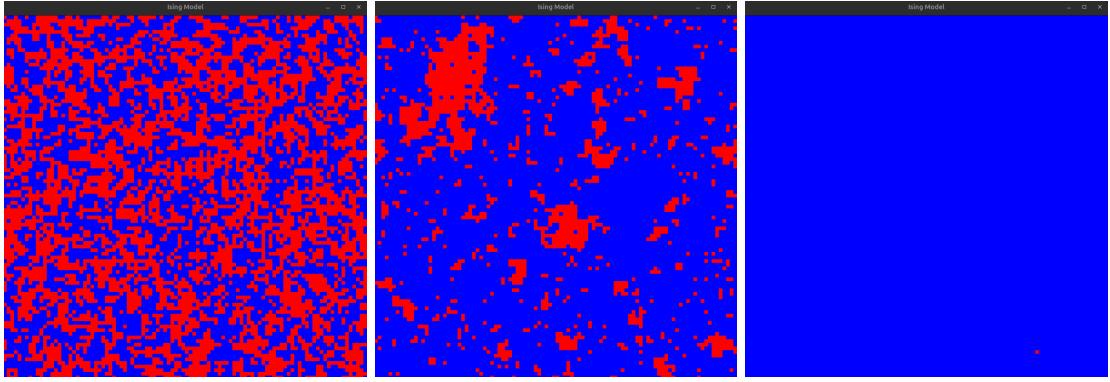


Figure 12: Snapshots from the simulations when $L = 100$ (thus $N = 100^2$) for three different values of (starting from the left) $T = 4, T = 2.26, T = 1$

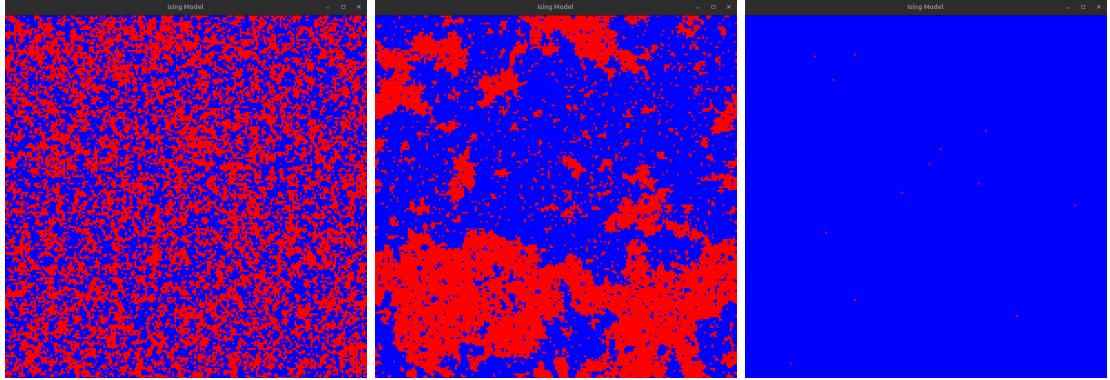


Figure 13: Snapshots from the simulations when $L = 200$ (thus $N = 200^2$) for three different values of (starting from the left) $T = 4, T = 2.26, T = 1$

spins (spontaneous breaking of symmetry). In the simulations I ran, I selected the initial configuration such that approximately 60% of the spins pointed up (blue), and thus we observe a symmetry breaking towards a macroscopic ordered phase where most of the spins are up (in agreement with the right panels of Fig. 12, Fig. 13, Fig. 14).

Finally, when $T = 2.26 \approx T_c$, the system exhibits peculiar properties. Around criticality, we know that the model takes a long time to thermalize, and consequently, the MC Markov chain requires a huge number of steps to reach the desired invariant distribution (the Boltzmann one). Therefore, one cannot be completely sure that the snapshots in Fig. 12, Fig. 13, and Fig. 14 (central panels) truly represent samples drawn from the equilibrium Boltzmann distribution of the Ising model. From the theory of phase transitions, we also know that, at criticality, the correlation length diverges⁴. This means that spins will tend to create *correlated clusters* whose size is of the same order as the correlation length (diverging at T_c), i.e., the system will appear as a mix of mesoscopic ordered phases that take a long time to thermalize with each other. These clusters can be easily seen in the central panels of Fig. 12, Fig. 13, and Fig. 14.

4.2 Time behaviour of the order parameter

By showing the time behaviour (in MC steps) of both the order parameter (magnetization per spin) and the energy density discuss what is happening above, below and close to the transitions. How can we establish that the Markov chain is sampling according to the desired probability distribution independent on the initial conditions?

Resolution

Let us fix the lattice size $L = 500$ (thus total number of spins $N = 500^2$). Fig. 15 shows the trace plot (that is, the random walk of the MC chain) and the final histogram for both the average

both equilibrium macrostates. A more detailed discussion will follow up later

⁴This is strictly true only in the thermodynamic limit, i.e., for an infinite system. Our simulations are, of course, finite, so the correlation length will simply reach a maximum and then decrease again.

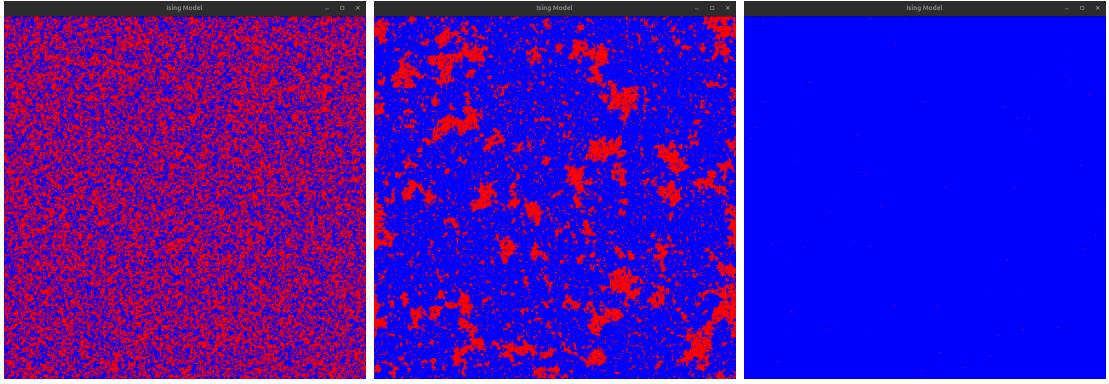


Figure 14: Snapshots from the simulations when $L = 500$ (thus $N = 500^2$) for three different values of (starting from the left) $T = 4, T = 2.26, T = 1$

magnetization per spin m and the energy per spin ϵ when $T = 4$ and when $T = 1$, i.e., far from the critical point. As one can easily observe, both equilibrium distributions for m and ϵ resemble a normal distribution whose average value represents the thermodynamic value at that temperature (and the variance should decrease as $N \rightarrow \infty$). How can we establish whether the Markov chain has actually converged to the target distribution, i.e., the Boltzmann distribution? There are many ways to ensure that the Markov chain has reached stationarity. The most important property of an invariant distribution is that it is asymptotically reached independently of the initial condition. This means that one could repeat the same MCMC simulation with a different initial starting configuration and compare the results. If the final distribution looks the same independently of the specific choice of the initial configuration, then we have probably succeeded in sampling from the correct Boltzmann distribution. This was implemented in the code by launching a multi-threaded set of chains where each thread receives different initial configuration and by comparing the final distributions.

Another way to get a sense of the stationarity of the Markov chain is by closely examining the trace plot of the chain. Fig. 15 shows desirable and clear trace plots for both m and ϵ . The dependence on the initial condition is quickly damped (the abrupt drop observed in the magnetization trace plots), and the algorithm remains in a well-defined region of the m (or ϵ) space, exploring, however, nicely the distribution space.

A different scenario appears when T is close to the critical value ($T = 2.26$, Fig. 16). Now the magnetization trace plot clearly looks like it hasn't yet reached a plateau, and the algorithm was probably still exploring the space when it reached the inputted maximum number of MC steps. This is to be expected: around criticality, MCMC algorithms that use local moves suffer from *critical slowing down*, meaning that they take a huge number of steps to finally reach the invariant distribution. Computing statistical averages on those histograms is problematic, since the system has not reached equilibrium and any thermodynamic average would make no physical sense. Why does this happen? The critical slowing down of the MCMC algorithm is related, once again, to the diverging property of the correlation length ϵ around the critical point. As we have already claimed before, around T_c bubbles of correlated spins will start to appear; out of criticality, those correlated clusters are usually small, but when approaching $T \rightarrow T_c$ their size diverges according to a power law. Take again the example of a 2D spin system where one spin has four nearest neighbors when $T \approx T_c$. If this spin is surrounded by aligned spins, its contribution to the energy is $E = -4$ and after the reversal of this spin, this becomes $E = 4$, thus a ΔE associated to the spin flip of 8. Right at T_c , the acceptance probability is low for the Metropolis algorithm: $A(s \rightarrow s') = \exp(-8/T_c) = 0.0294$: most of the flipping attempts are rejected. Only spin flips at the edge of a cluster have a significant effect over a longer time; but their fraction becomes vanishingly small when the critical temperature is approached and the cluster size diverges. Hence, the critical divergence of ϵ is directly tied to the inefficiency of our sampling algorithm at criticality⁵

An interesting phenomena that was observed is that, in general, the average energy per spin $\langle \epsilon \rangle$ has much nicer convergence properties than the magnetization per spin $\langle m \rangle$. This is particularly evident in Fig. 16, where it's easy to see how the MC chain associated to m is still struggling to find the desired invariant distribution after 2000 MC steps, whereas the chain associated to $\langle \epsilon \rangle$ has nicely found its equilibrium distribution after just ≈ 500 MC steps.

⁵To sample efficiently when $T \approx T_c$, one should use MC proposal moves that flip entire clusters of spin (Wolff algorithm)

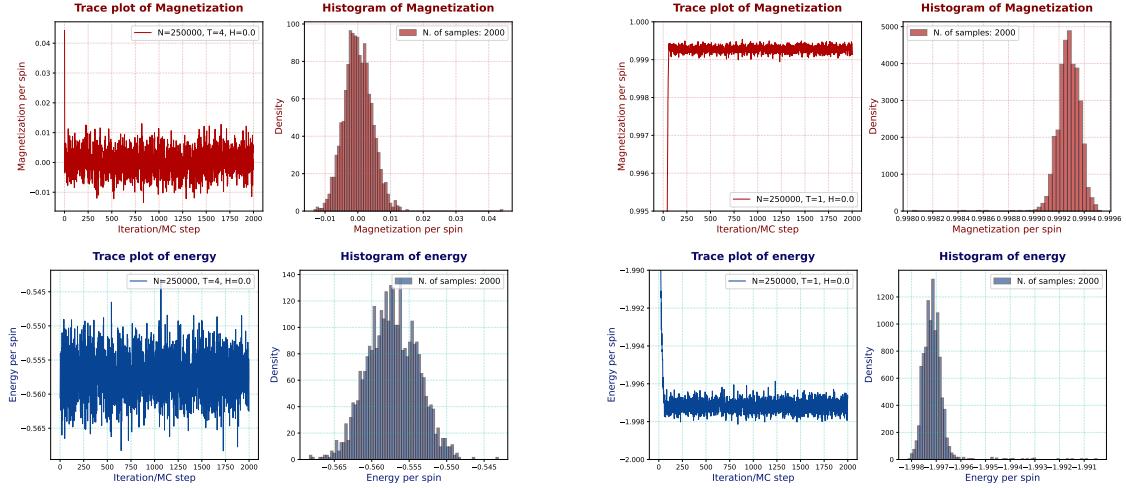


Figure 15: (Top Left) Histograms and trace plots, i.e. collected samples of the MCMC over time, for the average magnetization m (top sector) and for ϵ (bottom sector) when $N = 500^2, T = 4$ (left sector) and $T = 1$ (right sector). The simulation was run until 2000 samples were collected. The initial configuration was quickly forgotten, as one can see by examining the abrupt drop during the first MC steps. The trace plots correctly moves in the phase space and efficiently explores it, while the histograms closely resemble a Gaussian distribution peaked around the thermodynamic value $\langle m \rangle, \langle \epsilon \rangle$.

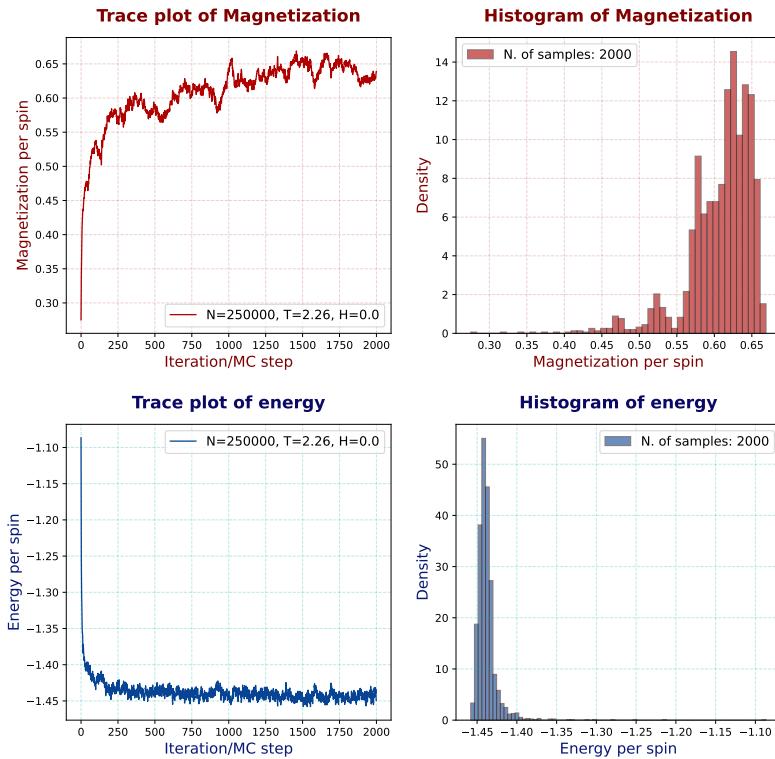


Figure 16: (Top Left) Trace plot for the average magnetization m when $N = 500^2, T = 2.26$, and $H = 0$. The simulation was run until 2000 samples were collected. Because we are close to criticality, the system takes a long time to reach equilibrium (critical slowing down). This can be assessed by looking at the trace plot of the MCMC, whose evolving trend clearly suggests that the algorithm has not yet reached the stationary state, i.e., the Boltzmann distribution.(Top Right) Histogram of the m values (over 2000 samples), using the same parameters as in the trace plot. These results are not representative, since we are not yet sampling from the correct equilibrium distribution.(Bottom Left) Trace plot for the average energy per spin ϵ when $N = 500^2, T = 1$, and $H = 0$. (Bottom Right) Histogram of the collected ϵ samples.

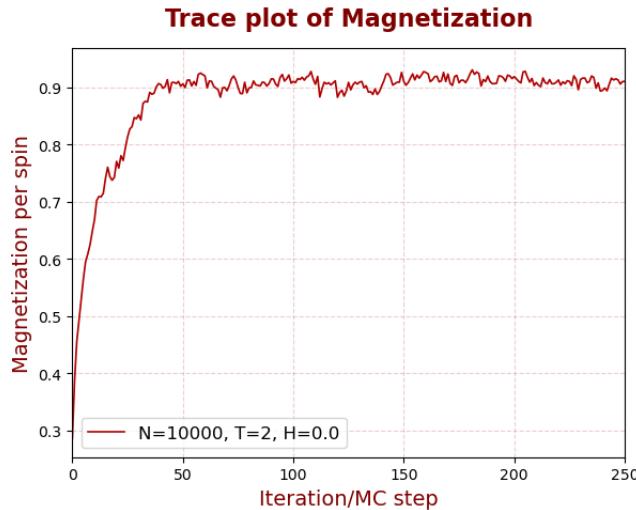


Figure 17: Trace plot for m when $T = 2$ and $N = 100 \times 100$. The Metropolis algorithm takes a while to reach stationarity (approximately $\tau_{eq} = 50$ MC steps). After τ_{eq} , the algorithm is finally sampling from the desired distribution and we can start computing the thermal averages we are interested in.

4.3 Equilibration time

After having determined the equilibrium time τ and disregarding the samples for $t < \tau$ estimate the ensemble averages of the magnetization per spin, the energy per spin, and the corresponding fluctuations (specific heat per site and magnetic susceptibility per site). Plot their estimates as function of the chosen temperatures and discuss the behaviour of the curves as the linear size of the system L increases.

Resolution

The procedure of disregarding samples for $t < \tau$ with τ equilibration time is called *burn-in*; τ is nothing but the characteristic time it takes for the algorithm to converge to the target Boltzmann distribution. Fig. 17 represents the trace plot for a MCMC run with $T = 2$ (under-critical regime) and $N = 100 \times 100$: it's quite easy to see that the Markov algorithm takes approximately $\tau_{eq} = 50$ MC steps to reach equilibrium. We can now proceed to estimate the thermal averages over samples satisfying $t > \tau$, in particular the order parameter *magnetization per spin* estimated by:

$$\langle m \rangle \approx \bar{m} = \frac{1}{T_{max}} \sum_{i=\tau}^{T_{max}} m_i$$

and the energy per spin:

$$\langle \epsilon \rangle \approx \bar{\epsilon} = \frac{1}{T_{max}} \sum_{i=\tau}^{T_{max}} \epsilon_i$$

where the summation is performed on all MC steps starting from τ to T_{max} , the maximum MC steps. By extracting the variance, we can also estimate the specific heat per spin and the magnetic susceptibility per spin:

$$\begin{aligned} \frac{\chi}{N} &= N \frac{Var(m)}{T} = N \frac{\langle m^2 \rangle - \langle m \rangle^2}{T} = N \frac{\frac{1}{T_{max}} \sum_i m_i^2 - \bar{m}^2}{T} \\ \frac{C}{N} &= N \frac{Var(\epsilon)}{T^2} = N \frac{\langle \epsilon^2 \rangle - \langle \epsilon \rangle^2}{T^2} = N \frac{\frac{1}{T_{max}} \sum_i \epsilon_i^2 - \bar{\epsilon}^2}{T^2} \end{aligned} \quad (33)$$

Before showing the results, we need to make a small modification to the definition of the order parameter itself. In fact, one should always remember that, as long as the system size is finite,

$$\langle m \rangle = 0, \quad \forall T$$

This is because the Ising model enjoys a discrete \mathbb{Z}_2 symmetry: the configuration $\sigma_1, \sigma_2, \dots, \sigma_N$ with average magnetization m has the same energy as the flipped configuration $-\sigma_1, -\sigma_2, \dots, -\sigma_N$

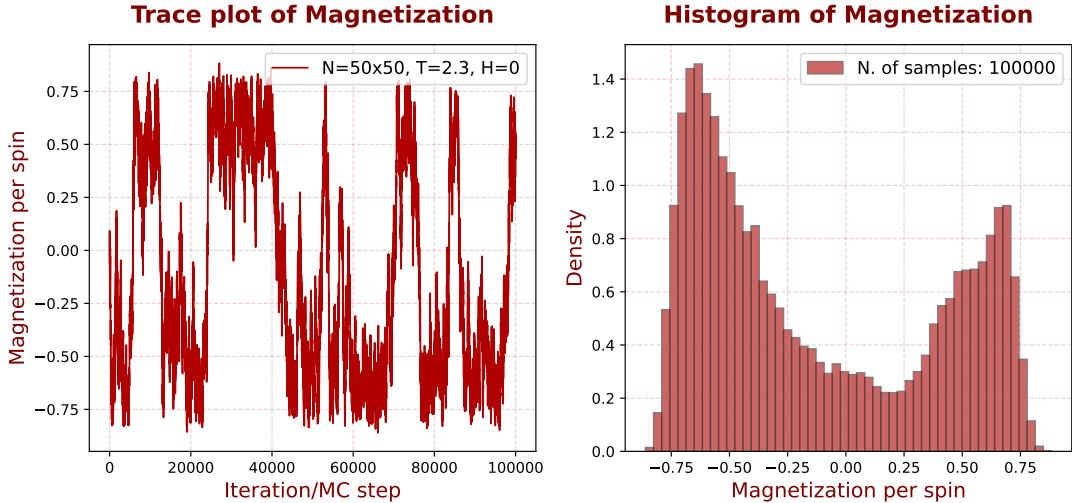


Figure 18: *Trace plot and histogram of m , average magnetization per spin ($T = 2.3, N = 50 \times 50$)*. Metropolis algorithm is ergodic, hence it will explore all of the parameter space. When N is finite, the amount of energy needed to jump from one equilibrium state to the other is finite and decreases when N decreases. Thus, especially for small N , our algorithm will explore both equilibria, resulting in a bimodal distribution and in a zero average magnetization.

with average magnetization $-m$ (discarding borders effect). Therefore, they are equiprobable, since the Boltzmann equilibrium distribution is a function only of the Hamiltonian:

$$P(\vec{\sigma}) = P(-\vec{\sigma})$$

This necessarily implies that averaging over the complete Boltzmann distribution yields $\langle m \rangle = 0$ for any temperature. Indeed, if one inspects the trace plot at $T = 2.3$ for a relatively small system size $N = 50 \times 50$, a peculiar behaviour appears (Fig. 18). The random Glauber dynamics seems to have found two symmetric stable points where the chain spends most of its time ($\langle m \rangle \approx \pm 0.6$). We must stress that what is shown in Fig. 18 is not due to a convergence problem: the MCMC is effectively sampling from the correct Boltzmann distribution. Since the Metropolis algorithm is ergodic by design, it explores the entire parameter space; and as long as N is finite, there will always be two stable wells (minima of the free energy) with $m_1 = -m_2$. The algorithm will eventually jump from one equilibrium to the other, although the probability of such transitions decreases significantly with system size.

How is it thus possible to observe a non-zero $\langle m \rangle$, especially in the ferromagnetic phase? The point is that, for any finite N , no true phase transition occurs and $\langle m \rangle$ is indeed zero for all temperatures (because the Metropolis algorithm is ergodic and, given enough time, it will explore both wells). When one wants to understand what happens in the thermodynamic limit, it is crucial to choose the correct order of the limits:

$$\langle m \rangle = \lim_{H \rightarrow 0} \lim_{N \rightarrow \infty} m(T, H) \neq \lim_{N \rightarrow \infty} \lim_{H \rightarrow 0} m(T, H) = 0 \quad (34)$$

In the (correct) thermodynamic limit, the system undergoes a *spontaneous symmetry breaking*: even though the model is symmetric under spin reversal, it selects a preferred spin orientation, and consequently $\langle m \rangle$ can become different from zero. The Boltzmann distribution remains symmetric under $\vec{\sigma} \rightarrow -\vec{\sigma}$, but when $N \rightarrow \infty$ the thermal energy required to jump from one ordered phase to its opposite diverges, and thus the system will remain permanently in one of the two regions of the m -space, depending on the initial condition (the system is no longer ergodic). However, our simulations are clearly finite and the thermodynamic limit is unreachable. To avoid averaging over chains such as the one in Fig. 18 (which would yield an incorrect value of m), we can redefine our order parameter as $\langle |m| \rangle$, so that the sign of m becomes irrelevant⁶. Results are shown in Fig. 19. As one would expect, the plot of the average magnetization per spin $\langle m \rangle$ displays what appears to be a second-order phase transition somewhere in the interval $2 < T < 2.5$ (and indeed, we know the exact critical temperature to be $T_c \approx 2.26$ thanks to Onsager).

⁶A different approach would have been to introduce a small magnetic field H .

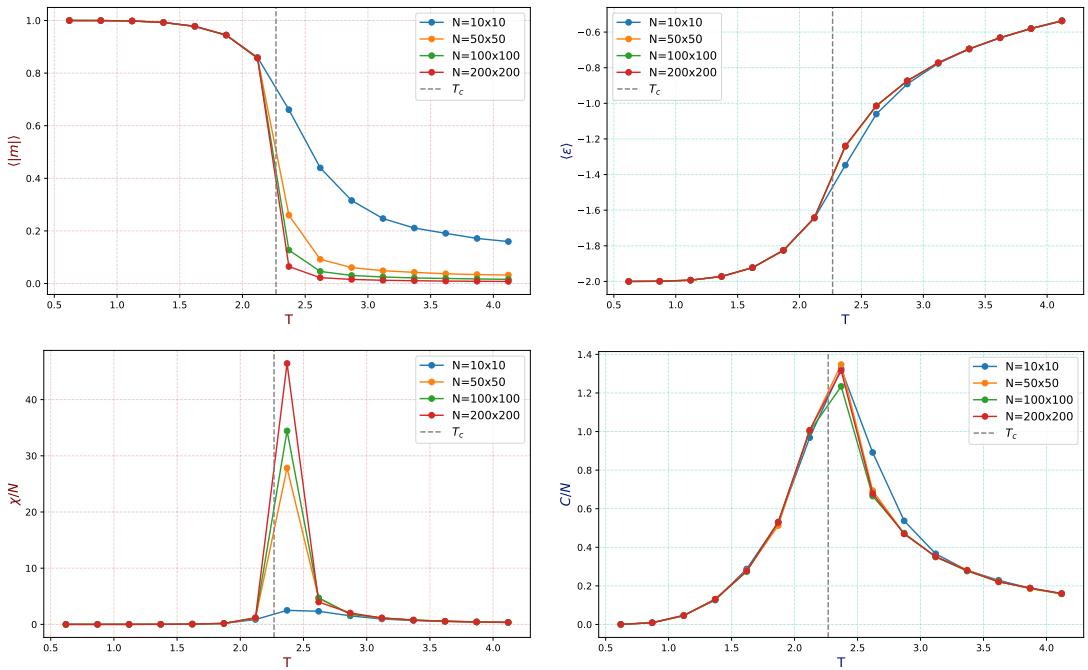


Figure 19: (Top left) Plot of the average magnetization per spin $\langle m \rangle$ sampled at evenly spaced temperatures ($\Delta T = 0.5$) for a 2D lattice of size $N = 10^2, 50^2, 100^2, 200^2$. The phase transition is evident: when $T < T_c \approx 2.26$, the magnetization m is not zero and tends to decrease continuously, whereas when $T > T_c$, the system relaxes to a totally disordered phase and, consequently, the magnetization per spin is 0. The abrupt drop is an artifact, since in this plot we jumped from $T = 2$ to $T = 2.5$; a closer look to the critical region is shown in Fig. 20 (Top Right) Same as in the previous plot, but regarding the average energy per spin ϵ (Bottom Right) Plot of the estimated magnetic susceptibility per spin. As expected, it displays a sharp peak around the critical point. This also means that the variance of the corresponding distribution (here, m) increases dramatically (Bottom Left) Same as in the previous panel, but for the heat capacity per spin

In the subcritical regime, the system relaxes to an ordered ferromagnetic phase where $\langle m \rangle \neq 0$, whereas in the supercritical regime the system behaves paramagnetically with $\langle m \rangle = 0$. In the true thermodynamic limit ($N \rightarrow \infty$), the magnetization curve would exhibit a continuous but non-differentiable point at $T = T_c$; however, since our simulations necessarily involve a finite number of spins, we should instead observe a smoother crossover that gets steeper as N grows. The energy plot is also reasonable: when T is low, the energy term will prevail over the entropic contribution in the free energy and the system will relax to a minimum energy state. This is equivalent to configurations where all spins are oriented in the same direction, hence the energy per spin should be $\epsilon = -2$, precisely what we observe. The plots regarding the magnetic susceptibility and specific heat per spin displays a maximum around T_c , which again is in perfect agreement with our theoretical understanding of the model where χ and C should diverge following a power law at criticality. As the dimension of our simulation grows, the maximum peak of χ/N or C/N gets larger, in agreement with finite size scaling

Let's have a closer look at the critical region (Fig. 20) using the same lattice sizes. As before, the $\langle |m| \rangle$ curves show the onset of a phase transition around $T \approx T_c$; the larger the lattice, the more accurate the behaviour becomes. As a reference, the expected functional form of $\langle |m| \rangle$ in the thermodynamic limit is

$$\langle |m| \rangle = \left(1 - \sinh^{-4}\left(\frac{2}{T}\right)\right)^{1/8} \quad (35)$$

which scales with the critical exponent $\beta = 1/8$.

At criticality, the numerical points become noticeably noisier: this is a direct consequence of the divergence of the correlation length, which causes the fluctuations of the order parameter to increase dramatically. This behaviour is clearly reflected in the plots of χ/N and C/N : as $T \rightarrow T_c$, both quantities grow and would diverge if we had an infinite system. Their scaling behaviour in the vicinity of the critical point is given by

$$C \sim -\ln |t|$$

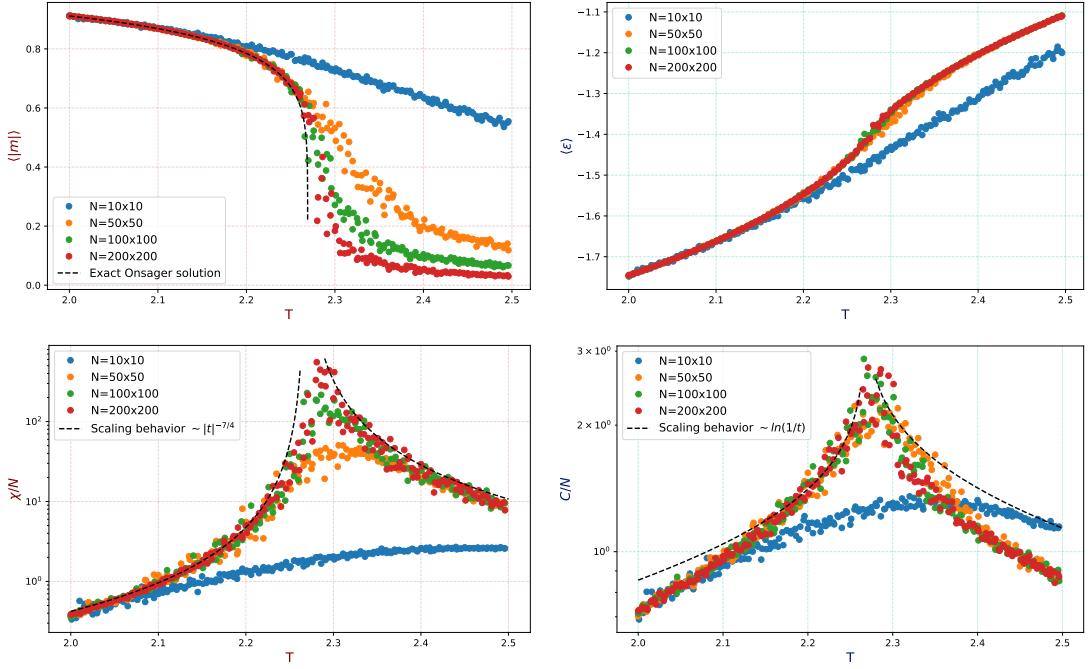


Figure 20: (Top left) Plot of the average magnetization per spin $\langle |m| \rangle$ for a 2D lattice of size $N = 10^2, 50^2, 100^2, 200^2$ sweeping through temperature with $\Delta T = 0.0025$ (thus one point for each temperature value. We could have run the simulation multiple times for each T value, but that would have required an insane compute time). (Top Right) Same as in the previous plot, but regarding the average energy per spin ϵ (Bottom Left) Plot of the magnetic susceptibility per spin, displaying a divergence-like behaviour around criticality (Bottom right) Plot of the specific heat per spin

$$\chi \sim |t|^{-\gamma} = |t|^{-\frac{7}{4}}$$

with $t = \frac{T-T_c}{T_c}$ and is reasonably well reproduced by our Metropolis simulations.

4.4 Integrated correlation time and critical slowing-down

Estimate the autocorrelation time of the magnetization and the energy for the MC simulations proposed above and estimate the errors accordingly

Resolution

The normalized autocorrelation function for m reads:

$$\rho(l) = \frac{\langle (m(t) - \langle m \rangle)(m(t+l) - \langle m \rangle) \rangle}{Var(m)} \quad (36)$$

which can be estimated using Metropolis sampling via:

$$\rho(l) = \frac{1}{(N-l)Var(m)} \sum_{t=1}^{N-l} (m_t - \bar{m})(m_{t+l} - \bar{m}) \quad (37)$$

where the variance can be estimated as usual as $Var(m) \approx \frac{1}{N-1} \sum_t (m_t - \bar{m})^2$, and \bar{m} is the estimate of the thermal average, $\bar{m} = \frac{1}{N} \sum_t m_t$. Similar results hold for ϵ , the average energy per spin. Results for the autocorrelation function when $N = 100 \times 100$ at different temperatures are shown in Fig.21. As expected, in both cases the autocorrelation function decreases to 0 as the lag increases. However, when $T = 2.5 \approx T_c$, the autocorrelation curve takes significantly longer to settle near 0, especially in the case of the magnetization, where it exhibits strong oscillations. This is again due to the fact that, around criticality, the system becomes strongly correlated, and it requires many MC steps to eliminate the internal correlation among the samples. To get an estimate of the average number of MC steps needed for our algorithm to sample truly statistically independent configurations, we can extract from the curve in Fig.21 the *integrated correlation time*:

$$\tau = 1 + 2 \sum_{l=1}^{\infty} \rho(l)$$

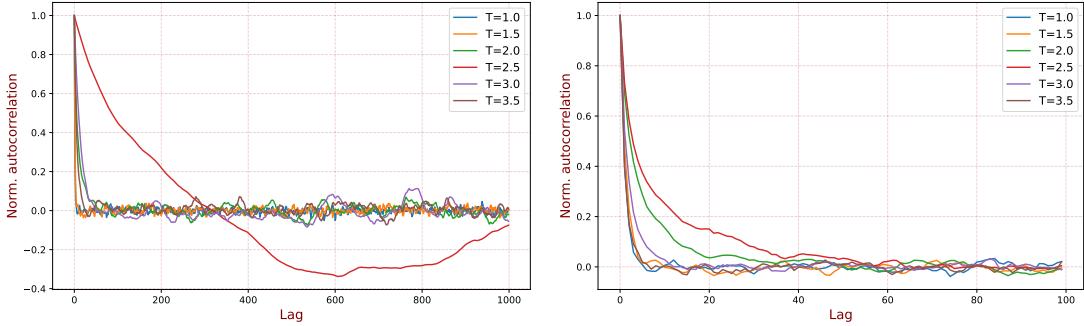


Figure 21: Normalized autocorrelation function for the magnetization per spin (right panel) and the energy per spin (left panel). Lattice size set to $N = 100 \times 100$, various temperatures.

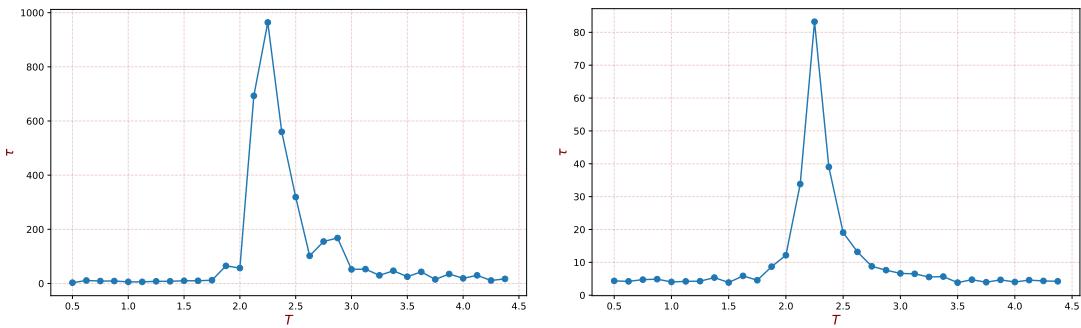


Figure 22: Plot of the integrated correlation time τ , i.e. average number of MC steps needed to achieve independency on the algorithm's samples, for magnetization (right panel) and energy (left panel). Lattice size set to $N = 100 \times 100$.

Assuming that the decay in Fig.21 is exponential, one can determine the characteristic time either by integrating $\rho(l)$ in the continuous case or by summing it in the discrete case (which motivates the formula above, up to the +1 factor). In Fig.22, we illustrate the behaviour of τ as a function of the temperature (fixing a lattice of $N = 100 \times 100$). As expected, τ diverges around the critical point, showing that at criticality the system is strongly correlated. This implies that, if we want our algorithm to sample correctly at criticality, we need to run it for a number of MC steps significantly larger than τ to ensure statistically independent samples. But there is more: if τ is very large, the errors associated with the mean values we obtained in the previous sections (i.e., the thermodynamic variables of interest) are significantly underestimated. The reason is straightforward: when we estimate the thermal average of a distribution using the arithmetic mean of a finite number of samples x_i drawn from that distribution (precisely what Metropolis is useful for), we are implicitly relying on the central limit theorem, which states that the arithmetic mean

$$\bar{x} = \frac{1}{N} \sum_i x_i$$

is, for large N , distributed according to a Gaussian centered at $\langle x \rangle$ and with variance $\sigma_x^2 = \frac{\sigma_x^2}{N}$. As N increases, the error in estimating $\langle x \rangle$ with \bar{x} decreases as $N^{-1/2}$. However, this argument only works if the samples x_i are independently and identically distributed, a necessary condition for the central limit theorem to hold. The problem is that our Metropolis algorithm does not generate independent samples by design, since it is a Markov process at its core.

By using τ as the average number of MC steps required to obtain an effectively independent sample, we can define an *effective number of steps* for a MCMC of total length N as:⁷

$$N_{eff} = \frac{N}{\tau}$$

This represents the effective number of statistically independent samples in the Markov chain that can actually be used to estimate averages and variances. Accordingly, the error on the estimate of

⁷Here N will indicate the number of MC steps, not the number of spins in the simulation

a thermal average is

$$\sigma^2 = \tau \sigma_{naive}^2$$

where σ_{naive}^2 is the variance estimated by naively treating all MC samples as independent, even though they are not (what we did in the previous sections). When the system is far from the critical point, this correction is negligible since τ is small: however, around criticality things change and the correction can be quite large! In the specific case of the system analyzed in Fig.22, if we build estimate on the magnetization ($\bar{m} \approx \langle m \rangle$) considering all samples in the MCMC output chain, we would underestimate the true error at a fixed number of spins $L^2 = 100 \times 100$ by a factor of $\tau \approx 1000$, quite a lot!

5 Active Ising Model

In this section, we simulate the implementation of the active Ising model as first described in the work of A. P. Solon and J. Tailleur. This model differs in several ways from the standard Ising model. In the latter, the (canonical) phase-space distribution is known *a priori*, and the main difficulty lies in computing averages over that distribution, since the normalization constant Z is usually inaccessible. The Metropolis algorithm allows us to reformulate the problem as a Markov chain whose invariant distribution is precisely the desired Boltzmann distribution, enabling us to compute the relevant statistical quantities.

Here, the perspective is fundamentally different. We still deal with a Markov chain, but the Markov process now describes a dynamical (yet stochastic) system. We define the state space and transition rates according to the model, and we interpret the evolution of the simulation as the actual temporal evolution of the system. In the standard Ising case with Metropolis dynamics, time played no role and the distribution was stationary; the number of steps required for convergence was merely an artifact of the algorithm. Here, by contrast, the number of Monte Carlo steps is directly tied to the physical time flow.

The transition rates used in this simulation are those proposed by Solon and Tailleur, namely

$$W(s \rightarrow s') = \exp\left(\frac{s\beta m_i}{\rho_i}\right), \quad (38)$$

where m_i and ρ_i denote, respectively, the total magnetization and total number of particles in cell i . The transition rates for the “hop move” in two dimensions are

$$W = \begin{cases} D, & \text{if moving up or down,} \\ D - s\varepsilon, & \text{if moving right,} \\ D + s\varepsilon, & \text{if moving left.} \end{cases} \quad (39)$$

Thus, in a sufficiently small time interval Δt , a spin s in cell i flips with probability $P = W(s \rightarrow s')\Delta t$, moves in one of the four directions with total probability $P = 4D\Delta t$, and with probability $P = (4D + W(s \rightarrow s'))\Delta t$ simply remains in place. It should now be clear that in this model we only specify the transition rates as functions of the parameters; we do not impose any invariant distribution. Indeed, it is straightforward to show that the Markov process defined by Eqs. (38) and (39) is not even ergodic (this follows, for instance, from Kolmogorov’s cycle criterion). The process has no limiting stationary distribution, making it a prototypical example of a NESS (*non-equilibrium stationary state*), i.e. a stationary state sustained by persistent internal currents.

In the following, we set $D = 1$ and $\Delta t = (4D + \exp(\beta))^{-1}$. The parameters of the model are therefore β , ε , and ρ , the spin density (number of particles in the system). Since our simulations necessarily have finite size, we denote by N_x and N_y the number of cells in the horizontal and vertical directions.

5.1 Keeping ε fixed

The parameter ε , called the *activity*, represents the bias in the diffusive motion of the particles (indeed, when $\varepsilon = 0$, motion becomes purely diffusive). For now, we fix $\varepsilon = 0.9$ and study the behaviour in the ρ - β plane. To keep track of the different phases explored by the system, we monitor order parameters such as the *magnetization profile* $m(x)$ (the average magnetization per column) and the *density profile* $\rho(x)$.

In particular, we can clearly distinguish three macroscopic phases:

- **Phase 1 (gas).** The system rapidly evolves toward a stationary state characterized by uniform density and magnetization across the rectangular lattice (Fig. 23, showing a single snapshot). The density profile fluctuates around a uniform value $\rho(x, t) \approx \rho \neq 0$, while the magnetization profile satisfies $m(x, t) \approx 0$. This regime is referred to as the *gas phase*, as the system forms a low-density, hot and spatially uniform state. Beyond the magnetization and density profiles, we also compute the *horizontal flux* $j(x)$, defined as the number of particles that, within a single time interval Δt , move from column x to column $x + 1$ minus those moving from $x + 1$ to x . As shown in Fig. 23, the flux heat map is nearly uniform, and the average flux satisfies $\langle j(x) \rangle \approx 0$. This suggests that, at least in this phase, the system effectively behaves as if an equilibrium distribution exists, fulfilling detailed balance (no internal currents⁸).

⁸A small clarification. For the general Markov chain to respect detailed balance, there should be no current in

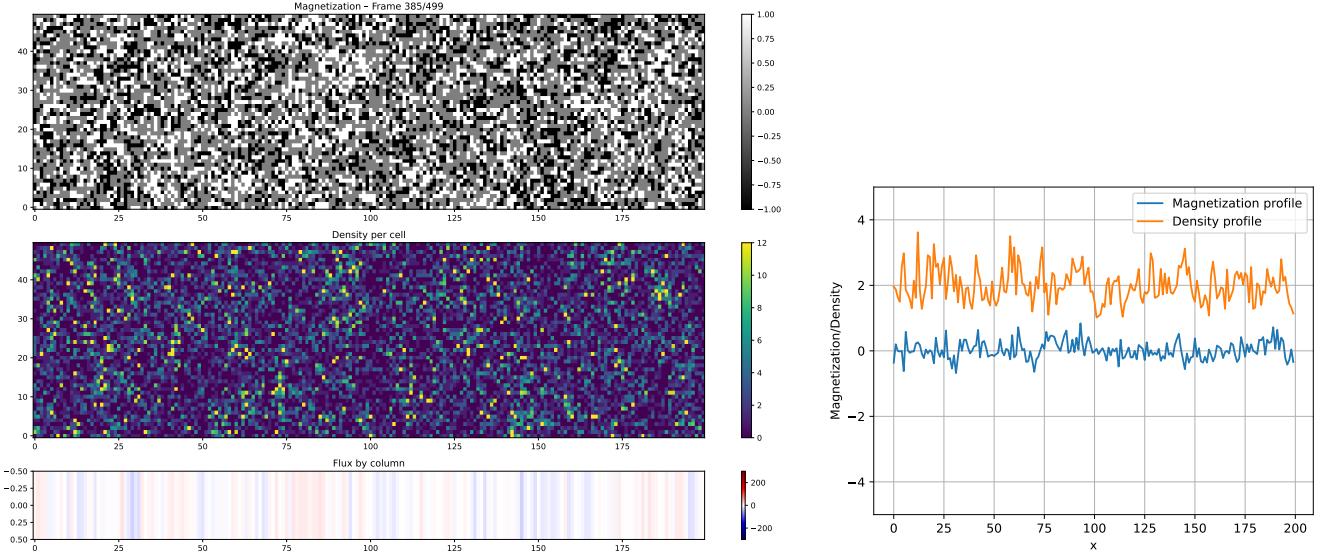


Figure 23: A snapshot of the simulation for $\beta = 1.4$ and $\rho = 2$. The system rapidly evolves into a gas-like phase, characterized by a uniform magnetization that remains close to 0 throughout the lattice. The particle flux is likewise homogeneous and nearly vanishing, suggesting that the resulting stationary state likely satisfies detailed balance.

- **Phase 2 (liquid).** The system evolves toward a high-density stationary state where both $m(x)$ and $\rho(x)$ remain spatially uniform (Fig. 24). However, unlike the gas phase, the average magnetization is now nonzero. The flux heatmap also changes dramatically: the profile horizontal flux $j(x)$ is homogeneous but satisfies $\langle j(x) \rangle \neq 0$! This indicates that the stationary state is not an equilibrium one, due to the presence of a strong internal current (e.g. in Fig. 24, biasing motion to the right). Such a state is a genuine NESS and resembles a liquid phase of ordinary matter, like a fluid sustaining a persistent current.
- **Phase 3 (coexistence).** For certain combinations of β and ρ , the system enters a special region of the parameter space where the stationary state consists of two coexisting phases: a dense, magnetized “liquid” droplet moving through a surrounding cloud of in-equilibrium gas (Fig. 25). The profile magnetization and density closely reflect this phase separation, showing a highly-dense plateau that moves through the system indefinitely⁹

5.2 Phase portrait

The diagrams for the magnetization and density profile in Fig.25, Fig.25 and Fig.25 were obtained by analyzing the column-wise averages for a given time t of the relevant order parameter $(m(x,t), \rho(x,t))$. In Fig.?? we illustrate the same diagrams for the three phases when averaging the temporal dimension, i.e.:

$$m(x) = \sum_{t=\tau}^{T_{max}} m(x, t) \quad (40)$$

$$\rho(x) = \sum_{t=\tau}^{T_{max}} \rho(x, t)$$

where τ was chosen such that the system had already converged to the stationary distribution (a burn-in). The plots confirm our understanding of the process and provide us with a quantitative way of defining the phase of the active system just by analyzing the magnetization and density profiles. In fact:

- When the system lies in the gaseous phase, the average¹⁰ density profile $\langle \rho(x) \rangle = \rho \neq 0$

⁹The simulation was performed using periodic boundary condition
¹⁰where here we are averaging both through columns and time

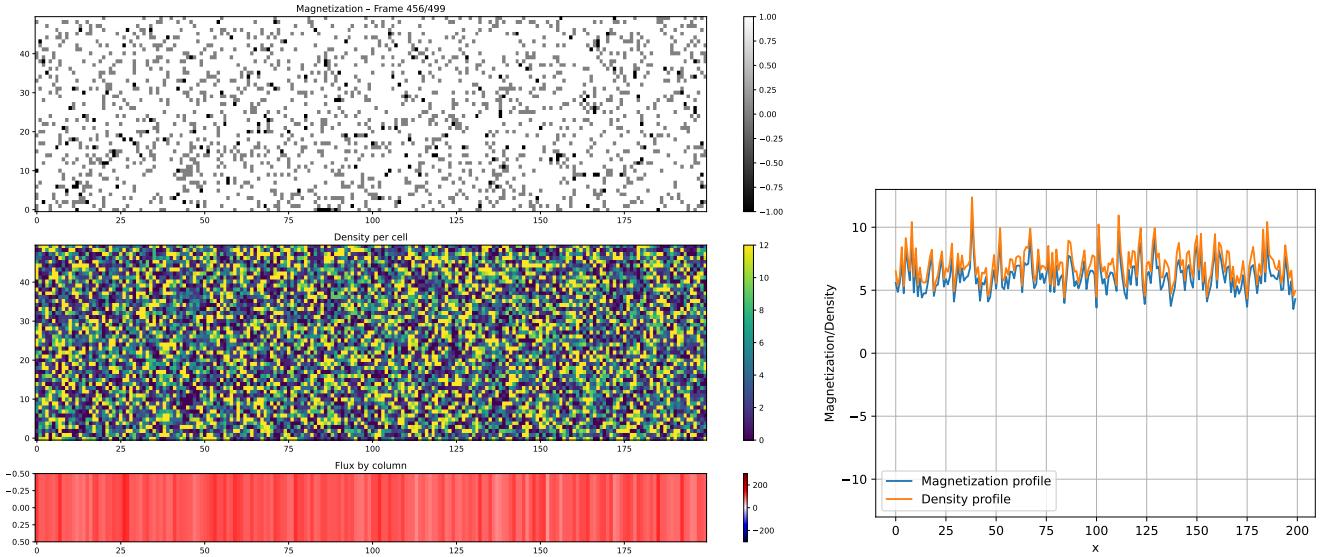


Figure 24: A snapshot of the simulation for $\beta = 2$ and $\rho = 7$. The system evolves into a dense low-temperature homogeneous state with a consistent horizontal flux $j(x) \neq 0$. The magnetization is different from zero as well. Put in other words, the dense and cold "fluid" moves through the rectangular lattice with a fixed velocity, under the effect of an internal flux (a NESS)

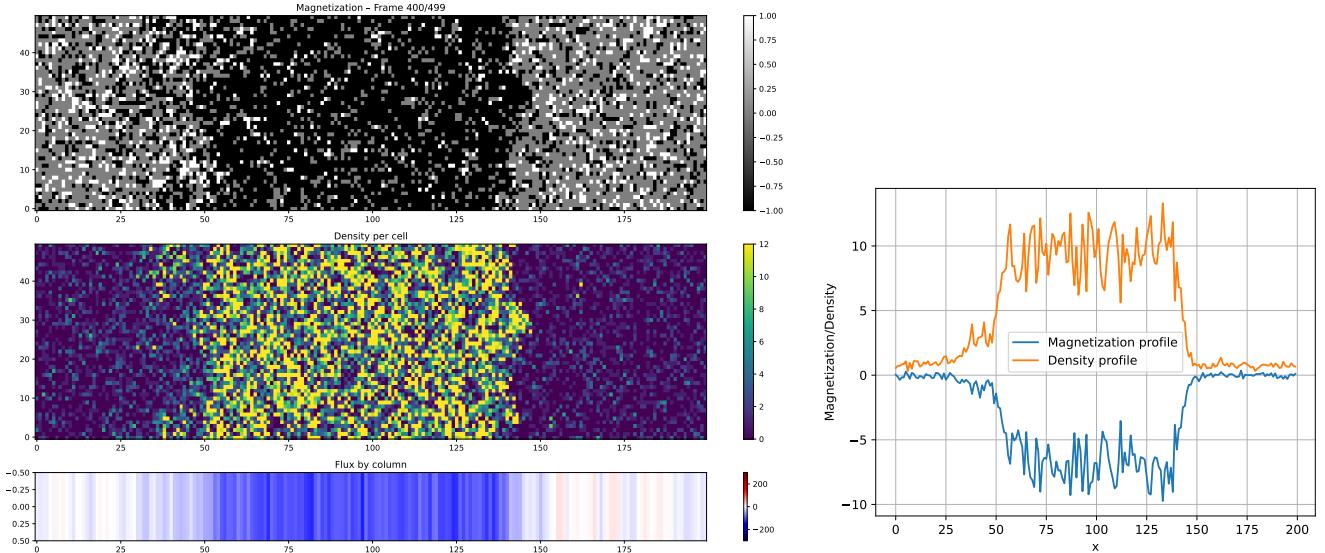


Figure 25: A snapshot of the simulation for $\beta = 1.6$ and $\rho = 5$. The system relaxes into a stationary state in which a "liquid" component (dense and strongly magnetized) travels through a thermalized "gas" phase, where no internal currents are present. Both the heatmaps and the corresponding profiles clearly illustrate this separation between a liquid region and a gaseous one.

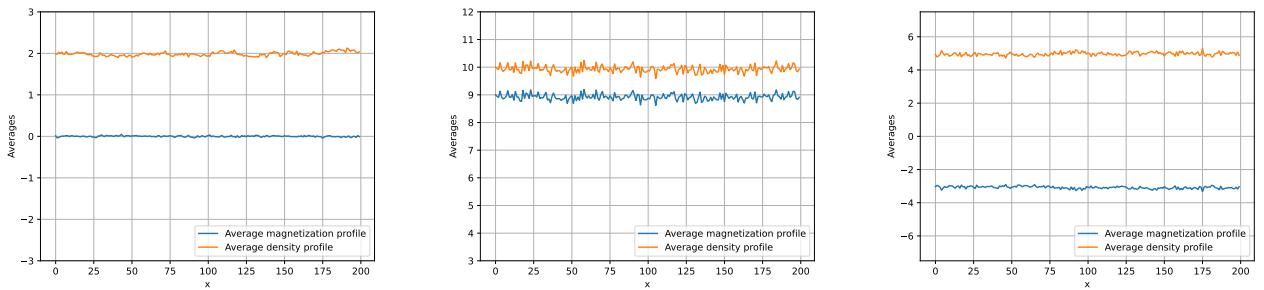


Figure 26: Magnetization and density profile averaged through time for gas phase (left), liquid phase (center), coexistence phase (right)

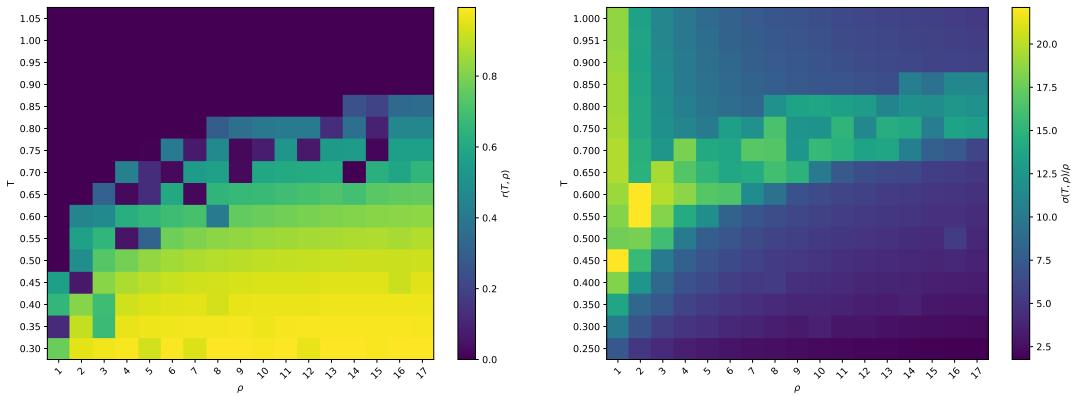


Figure 27: (Left) Phase diagram in the $T - \rho$ plane, using the order parameter r . (Right) Phase diagram in the $T - \rho$ plane, using the order parameter σ normalized by the average occupation of a column.

whereas the average magnetization should be $\langle m(x) \rangle = 0$. Defining $r = \frac{\langle m(x) \rangle}{\langle \rho(x) \rangle}$, then in the gas phase we should observe $r \approx 0$.

- In the liquid phase, the average profile magnetization $\langle m \rangle$ is somewhat close to $\rho = \langle \rho(x) \rangle$, thus we should observe $r \approx 1$
- When the system separates in the two stable phases, then r should be larger than 0 (the liquid phase increases the average magnetization) but less than 1 (the gas phase decreases it). The larger the liquid phase, the closer r should get to 1 (and viceversa for the gas phase).

Hence, r is a reliable indicator of the system's phase. A heat map of r for a discrete set of values of T and ρ is shown in the left panel of Fig. 27. In the high-temperature, low-density region of the diagram, $r \approx 0$ indicating that the system behaves effectively as a gas. In the low-temperature, high-density region, r approaches 1, corresponding to the fluid phase. Between these two regimes lies a transition band where r gradually increases from $r = 0$ (gas) to $r = 1$ (liquid), signaling that the system splits into two coexisting phases. The shape of the phase portrait is in agreement with the findings in the seminal paper of Solon and Tailleur. Another approach used in this paper stems from considering the spatial profile of the flux $j(x)$. When the system is in either the gaseous or the liquid phase, the flux profile is expected to be homogeneous (either equal to 0 or to some nonzero constant). In the multiphase region, however, the flux becomes inhomogeneous. Therefore, by computing the standard deviation of the flux profile (averaged over time), we should obtain higher values of σ in the multiphase region and lower values in the ordered phases (gas or liquid). The phase diagram $\sigma(T, \rho)$ is illustrated in the right panel of Fig. 27. To normalize this value, we can divide the raw standard deviation by the average number of particles in a column, thus the new order parameter can be written as:

$$s = \frac{\sigma(T, \rho)}{\rho \cdot N_y}$$

As expected, the heat map displays a bright tail in almost perfect correspondence with the multiphase region, reflecting the spatial heterogeneity of the flux profile. In addition, a rather bright appears when $\rho \approx 1$, but I can't figure out why that happens.