Exercises NMSM 2024-2025

Name Surname
matr. 0123456

November 23, 2024

Lecture 1: Crude Monte Carlo and inversion sampling

Sampling random points within D-dimensional domains by hit and miss

We want to evaluate the volume of an ellipsoid of given semi-axes $a = 3,\ b = 2,\ c = 2$. In order to reduce the error on our estimate from the get go, we make the following consideration. We can approach the problem in 2 ways : we can consider an integration box containing the whole ellipsoid and proceed with our calculation, or use to our advantage the problem's symmetry, and use as integration box the one containing one octant of the ellipsoid, and then multiply our estimate by 8.
The latter choice has the effect to reduce the variance in our sampling, and by consequence the error on our estimate.
This happens because , given that each hit or miss procedure is in fact a Bernoulli trial, the error on the volume estimate is evaluated by :

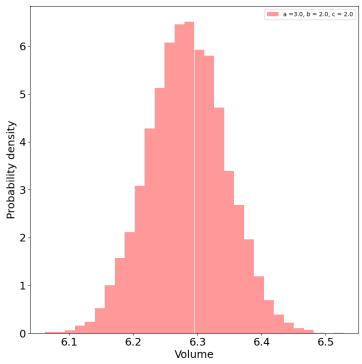


Figure 1.1: Volume distribution of the Monte Carlo estimate by hit and miss

$$\sigma_V = \sqrt{\frac{p(1-p)}{N}} V_{\text{integration box}}$$

Where p is our success probability, i.e. the probability to hit inside the integration box, and N is the number of trials, i.e. the number of iterations.
So, as we seen $\sigma_V \propto V_{\text{integration box}}$, so the smaller it is, better is our estimate.
Given this consideration we report in figure 1.1 the result obtained using the procedure with the ellipsoid octant for an ellipsoid of given semi-axes $a = 3,\ b = 1,\ c = 1$:

$$\langle V \rangle_1 \simeq 6.283 \pm 0.060$$

Now we propose to evaluate the volume of an ellipsoid of given semi-axes $a = 3,\ b = 1,\ c = 1$.
From the previous considerations we'll expect to have on this evaluation a smaller error, specifically, considering a situation in which p and N are the same, we can expect an error that is $\frac{V_{\text{box1}}}{V_{\text{box2}}} = 4$ times smaller, even before running any simulation.
We obtain for this ellipsoid:

$$\langle V \rangle_2 \simeq 1.571 \pm 0.015$$

Which confirms our prediction on the error. One can see also graphically this by comparing the 2 volume distributions and see in figure 1.2 that the one with the smaller integration box is narrower.
Moreover, given that the same volume can be evaluate analitically, we can study the deviation of the estimate from the analytical value.
In figure 1.3 we report for each of the 2 simulations the deviation from the analytical value, i.e $\Delta f = \mid f_{th} - \langle f \rangle \mid$ as a function of the number of iterations.

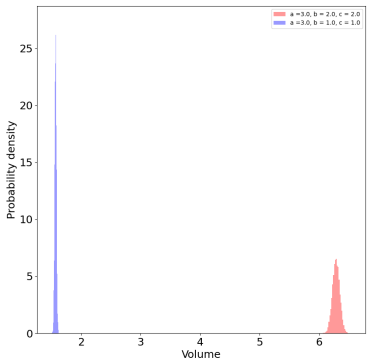


Figure 1.2: Volume distribution comparison between the 2 ellipsoids

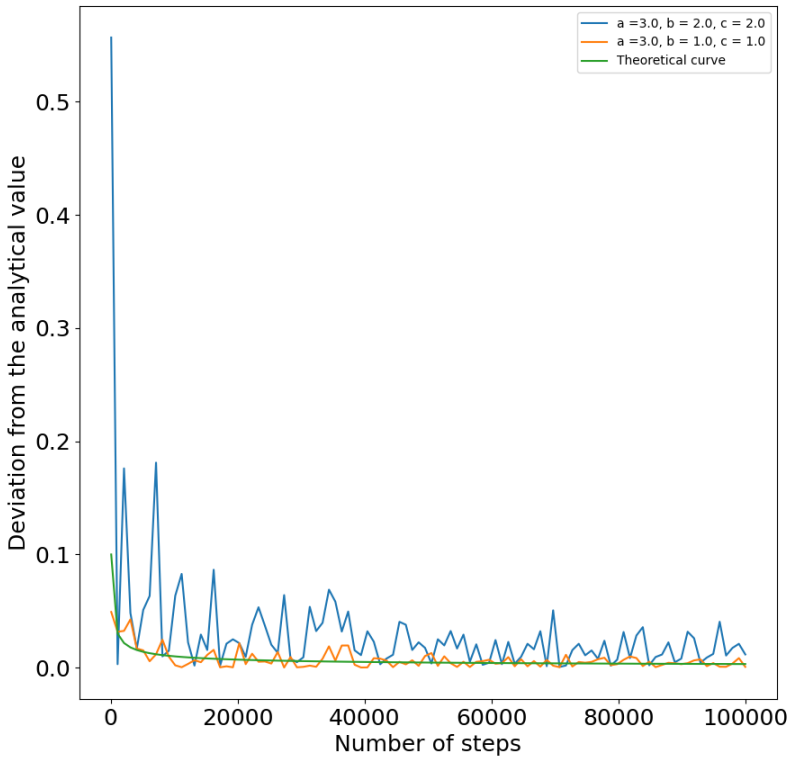


Figure 1.3: Comparison between the relative deviations for the 2 ellipsoids and the expected theroretical trend

Notice that, as discussed before, even if having a bigger number of iterations has a net effect of diminishing the error, the bigger variance in the case of the first ellipsoid makes the deviations bigger respect to the second ellipsoid.
We plotted for reference the general trend $\frac{1}{\sqrt{N}}$ which is the one to be expected in this kind of curves.

Sampling random numbers from a given distribution: inversion method

We start from the following PDF's:

$$\rho_1(x) = cxe^{-x^2}\ , x \in \mathbb{R}^+ \qquad \rho_2(x) = bx^4\ , x \in [0,3]$$

We first proceed by doing the normalization to find the coefficients c and b:

$$\int_{\mathbb{R}^+} \rho_1(x)dx = 1 \iff c = 2 \qquad \int_0^3 \rho_2(x)dx = 1 \iff b = \frac{5}{243}$$

Then one can evaluate the CDF's by definition:

$$F_1(t) = \int_0^t \rho_1(x)dx = 1 - e^{-t^2},\ t \in \mathbb{R}^+ \qquad F_2(t) = \int_0^t \rho_2(x)dx = \frac{t^5}{243},\ t \in [0,3]$$

Finally by inversion:

$$F_1^{-1}(y) = \sqrt{\ln\left(\frac{1}{1-y}\right)} \qquad F_2^{-1}(y) = \sqrt[5]{243y}$$

For both $y \in (0,1]$.
In order to implement this we use algorithm 2. We report in the following the obtained graphs:

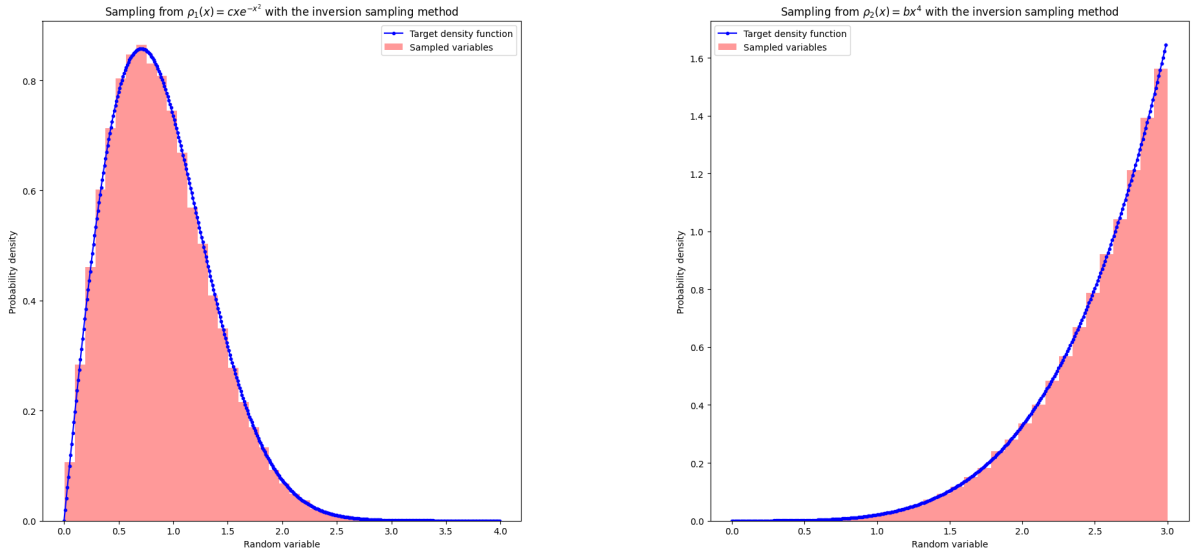


Figure 1.4: Sampling from distribution ρ_1 and ρ_2 using the inversion method.

Lecture 2: Rejection method

The rejection sampling method is useful when our pdf is not easily invertible, (or not invertible analytically) and so the inversion method cannot be applied.
Suppose to have as pdf:

$$\rho(x) = A_f \exp\left(-8\left(\frac{x^2}{2} + \frac{x^4}{4}\right)\right)$$

Imposing the normalization condition we obtain:

$$\int_{\mathbb{R}} \rho(x)dx = 1 \qquad \Longleftrightarrow \qquad \frac{eA_f}{\sqrt{2}} K_{1/4}(1) = 1$$

Where we denote with $K_{1/4}(x)$ the Bessel function of the second kind. We find that $A_f \simeq 1.21$. Now we want to find an invertible pdf $g(x)$ (that we'll call candidate density) with the property:

$$\rho(x) \leq cg(x) \qquad \forall x \in S$$

Where S is the sample space of $\rho(x)$. The closer the function $cg(x)$ is to $\rho(x)$ the more efficient the sampling is.

There can be different ways to choose the candidate density. We explore different possibilities in 1D and higher dimensions.
First we must start with a guess for the candidate density and the c parameter. Then we see if this guess respects the proposed inequality: if it does we are done, if not we do another guess and try again.
In 1D the easiest way to to test our guess is the graphical one: we plot $\rho(x)$ and $cg(x)$ in the same graph and evaluate if the inequality holds. Another way can be to evaluate point-wise the inequality. In dimensions where no graphical rapresentation is available, the second method can be easily extended and in order to reduce the computational complexity instead of starting with a proper point-wise evaluation, we can roughly grid the hyper-domain to check if our guess of the candidate density is sensible. If it is, we refine the grid and iterate the process until we are satisfied.

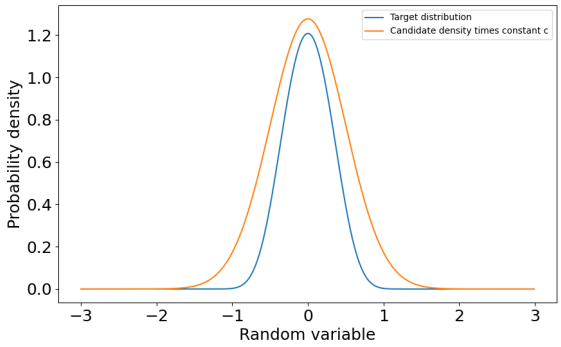


Figure 1.5: Graphical method for evaluating the inequality $\rho(x) < cg(x)$ in the rejection method

For our exercise, we can choose as candidate density $\mathcal{N}(0, \frac{1}{2})$ with c = 1.6, as shown in the picture 1.5

Then the sampling is done with algorithm 3. One can show graphically that the sample is good by plotted the properly binned random variables obtained by the process together with the target density, as showed in figure 1.6:

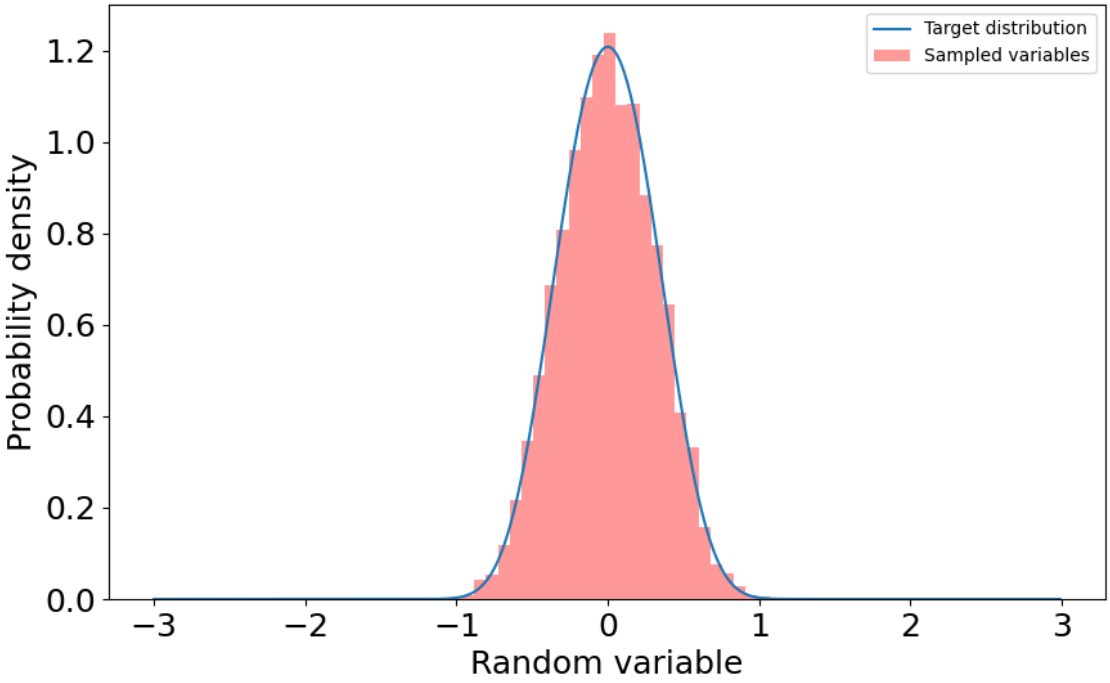


Figure 1.6: Sampled variables with target density

Lecture 3: Importance sampling

Part 1

We want to estimate the following integral:

$$\int_0^{\pi/2} \sin(x)dx$$

First we observe that the integral is easily evaluable analitically and is equal to 1.
We then proceed in doing a crude Monte Carlo estimate, obtaining the result in figure 1.7, leading to the evaluation:

$$\langle f \rangle_{\text{Crude MC}} \simeq 1.0010 \pm 0.0153$$

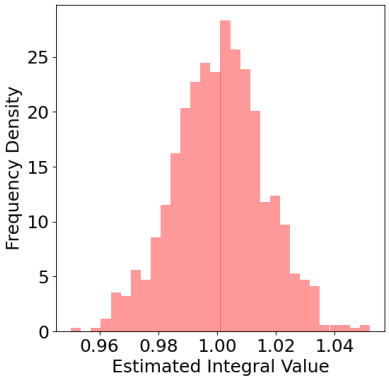


Figure 1.7: Crude Monte Carlo distribution (1000 iterations per point)

In order to reduce the error on the Monte Carlo evaluation one common approach is to use the importance sampling method.
The main algorithm is described in 4.
We propose as the new sampling function the family of functions defined such that:

$$g_{a,b}(x) = a + bx^2 \qquad a, b \in \mathbb{R}$$

We first make some consideration on the family of functions $g_{a,b}(x)$. First we impose, as they must be PDF's, the normalization condition on $[0, \pi/2]$, so that we obtain the constraint:

$$b = \frac{24}{\pi^3} - \frac{12a}{\pi^2}$$

In this way $g_{a,b}(x)$ collapses to a one parameter family:

$$g_a(x) = a + \left(\frac{24}{\pi^3} - \frac{12a}{\pi^2}\right) x^2$$

Moreover if we impose the condition $g_a(x) > 0$ we obtain for a the restriction $0 < a < 1$

There are now many recipes for choosing the function $g_a(x)$. We prescribe to the one that choose $g(x)$ if satisfies the condition $\rho(x) < g(x)$ when the product $f^2(x)\rho(x)$ is "large" and $\rho(x) >$

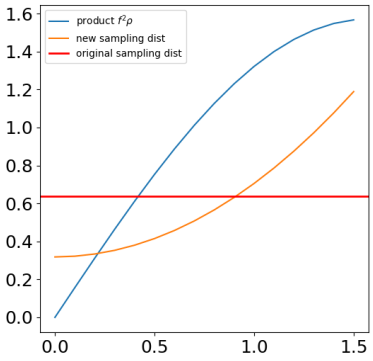


Figure 1.8: Graphical criterion for choosing a in the $g_a(x)$ family

$g(x)$ when the product $f^2(x)\rho(x)$ is "small" ,
where $\rho(x)$ in our case is the original sampling distribution, which is $U(0, \frac{\pi}{2})$.
We show in fig 1.8 that for $a = \frac{1}{\pi}$ these conditions are satisfied.

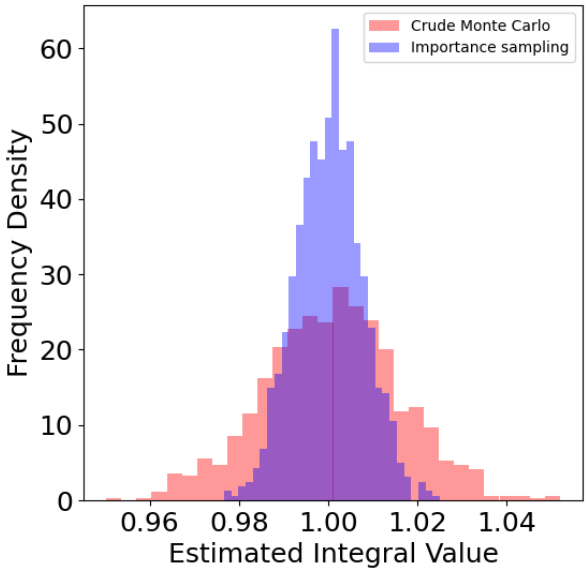


Figure 1.9: Distribution comparison between crude Monte Carlo and importance sampling with the same number of iterations

To sample from $g(x)$ we will use a rejection method algorithm using as a candidate distribution the truncated normal $\mathcal{N}(\frac{\pi}{2}, 1)_{|[0,\frac{\pi}{2}]}$. Note that this specific sampling presents some interesting technical details in Python discussed in the relative algorithm section
Now we perform the importance sampling for the same number of iterations as the crude Monte Carlo, and compare the results, seen in figure 1.9.
As expected the variance is reduced. Specifically we have for this Monte Carlo evaluation:

$$\langle f \rangle_{\text{Importance sampling}} \simeq 1.0001 \pm 0.0078$$

Which is an error reduction of $\sim 51\%$.

Part 2

We want now to evaluate the integral

$$\int_{\mathbb{R}} \left[e^{-(x-3)^2/2} + e^{-(x-6)^2/2} \right] \mathcal{N}(0,1) dx = \int_{\mathbb{R}} f(x) \mathcal{N}(0,1)$$

Where $\mathcal{N}(0,1)$ is the normal distribution with mean 0 and variance 1. One can prove that the integral has the analytical value easily by rewriting it as:

$$\int_{\mathbb{R}} \left[e^{-(x-3)^2/2} + e^{-(x-6)^2/2} \right] \mathcal{N}(0,1) dx = \frac{1}{\sqrt{2\pi}} \left[e^{-9/2} \int_{\mathbb{R}} e^{-x^2+3x} dx + e^{-18} \int_{\mathbb{R}} e^{-x^2+6x} dx \right]$$

And then by using the translation formula $\int_{\mathbb{R}} e^{-ax^2+bx}dx = e^{\frac{b^2}{4a}}\sqrt{\frac{\pi}{a}}$. In this way one obtains:

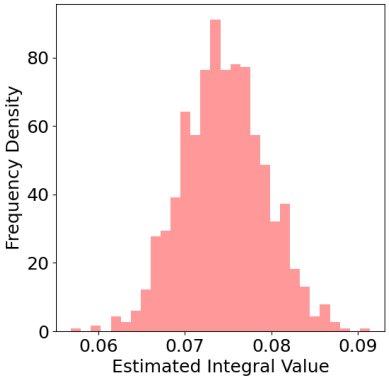
$$\int_{\mathbb{R}} f(x)\mathcal{N}(0,1) = \frac{1}{\sqrt{2\pi}}\sqrt{\pi}\left(e^{-9/2}e^{9/4} + e^{-18}e^{36/4}\right) = e^{-9}\frac{(1+e^{27/4})}{\sqrt{2}} \simeq 0.075$$

Instead of going through the analytical route, one can think to evaluate it by using a crude Monte Carlo method, by sampling respect the normal distribution, i.e. by evaluating $\langle f(x)\rangle_{\mathcal{N}(0,1)}$.
We obtain:

$$\langle f(x)\rangle_{\mathcal{N}(0,1)} = 0.0743 \pm 0.0048$$

As also shown in the picture 1.10.

If, as proposed by the exercise, we now try to evaluate the same integral by the importance sampling method using as sampling function the uniform distribution $U(-8,-1)$ for 1000 iterations, we will see nothing.
It's easy to see way if we show the involved functions together in a graph 1.11
The uniform distribution samples in that interval values for which the integrand function assumes values between 10^{-41} to 10^{-5}, so is not possible with this iteration number to reach convergence.



distribution for the eval-

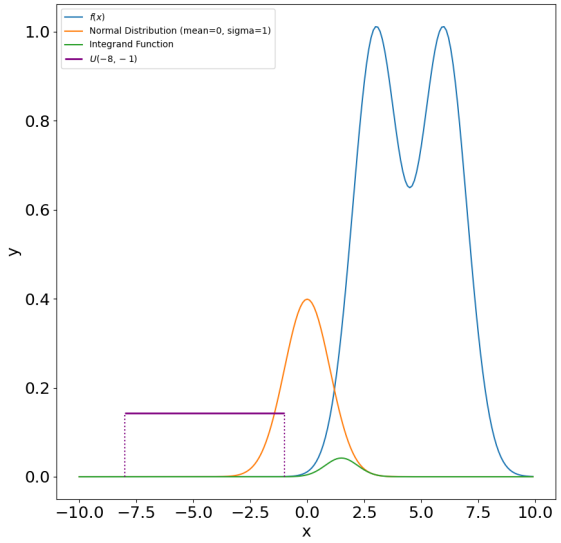


Figure 1.11: Graphical analysis for the bad choice of $U(-8,-1)$ as a sampling function for the integral $\langle f(x)\rangle_{\mathcal{N}(0,1)}$

With this number of iterations one can do way better if proposes for the importance sampling the distribution $U(-1,4.5)$, in this case one obtain the better estimate:

$$\langle f(x)\mathcal{N}(0,1)\rangle_{U(-1,4.5)} = 0.074518 \pm 0.000003$$

This result is further presented in figure 1.12.

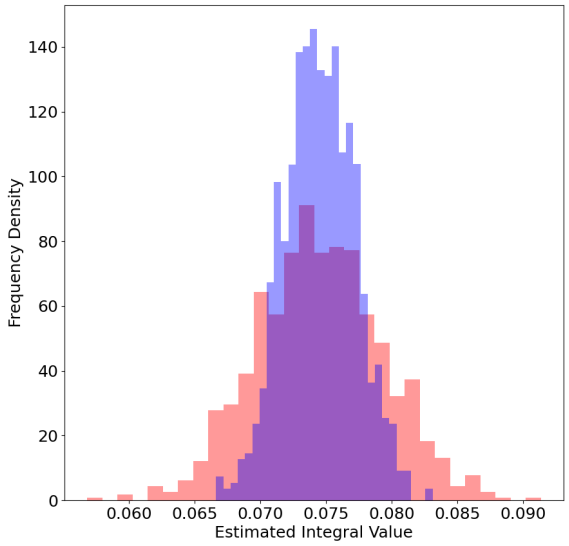


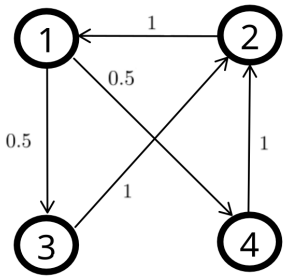
Figure 1.12: Comparison between the integral estimate distributions

Lecture 4: Markov chains

Part 1

We report the stochastic matrixes with the relative digraphs in figure 1.13.
One can prove from the digraph representation that chain A is irreducible, while chain B is not (for example in chain B from state 2 is not possible to reach any other state).

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



$$(B) \quad \mathcal{P} = \begin{pmatrix} 0.3 & 0.4 & 0 & 0 & 0.3 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.6 & 0.4 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

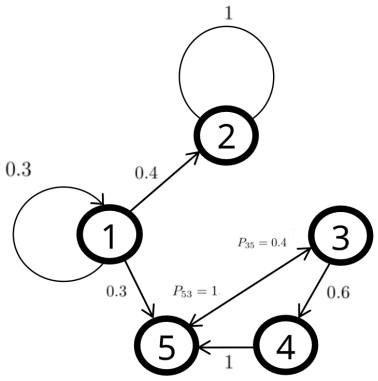


Figure 1.13: Stochastic matrixes with relative digraph representation

Moreover on chain A each state is periodic of period 3. In chain B state 1 is aperiodic and transient, state 2 is absorbing and states 3,4,5 are periodic of period 3.

Part 2

Suppose to have the Markov chain with the following stochastic matrix:

$$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{6} \\ \frac{3}{4} & 0 & \frac{1}{4} \\ 0 & 1 & 0 \end{pmatrix}$$

One can show with the corresponding digraph in figure 1.14 that is irreducible.

$$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{6} \\ \frac{3}{4} & 0 & \frac{1}{4} \\ 0 & 1 & 0 \end{pmatrix}$$

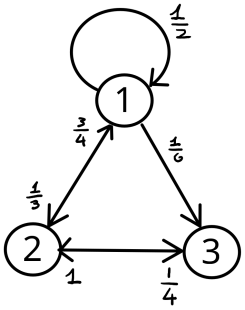


Figure 1.14: Stochastic matrixes with relative digraph representation

Moreover, given that state 1 is aperiodic, we can imply that the whole chain is aperiodic.

Now suppose to start the process in state 1, and that we want to evaluate the probability that it is in state 3 after two steps.
This can be easily accomplished by evaluating P^2 and then selecting from this matrix the element $P^2_{13} = \frac{1}{6}$.

Now we evaluate the stochastic matrix in the limit $n \to +\infty$ in 2 ways, first by using the analytical approach and the by computation trying to estimate it. First we evaluate the invariant distribution:

$$\bar{\pi}P = \bar{\pi} \iff P^t\bar{\pi}^t = \bar{\pi}^t$$

In the end one as to solve this linear system with the normalization constraint $\sum_i \pi_i = 1$. We then obtain the following linear system:

$$\begin{cases} \frac{1}{2}\pi_1 + \frac{3}{4}\pi_2 = \pi_1 \\ \frac{1}{3}\pi_1 + \pi_3 = \pi_2 \\ \frac{1}{6}\pi_1 + \frac{1}{4}\pi_2 = \pi_3 \\ \pi_1 + \pi_2 + \pi_3 = 1 \end{cases} \iff \bar{\pi} = \left(\frac{1}{2}, \frac{1}{3}, \frac{1}{6}\right)$$

The limiting stochastic matrix will have for each row the invariant distribution:

$$P^\infty = \begin{pmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{6} \end{pmatrix}$$

We obtain the same result in the submitted program.

Balls and boxes

We can solve this exercise using a Markov chain approach. Called a_i the state in which there are i red balls in box A there can be just 3 possible states (0,1,2 balls). So we need to evaluate the transition probabilities. In general, for a given state i:

- $a_i \to a_{i+1}$ = $\mathbb{P}$(red extraction in B) and $\mathbb{P}$(white extraction in A)
- $a_i \to a_{i-1}$ = $\mathbb{P}$(red extraction in A) and $\mathbb{P}$(white extraction in B)

- $a_i \to a_i$ = ($\mathbb{P}$(red extraction in B) and $\mathbb{P}$(red extraction in A)) or ($\mathbb{P}$(white extraction in B) and $\mathbb{P}$(white extraction in A))

Now we can proceed to evaluate these probabilities and then evaluate the transition matrix elements. In general one has, for a given state i:

$$\mathbb{P}(\text{red extraction in A at state i}) = \frac{i}{2} \qquad \mathbb{P}(\text{white extraction in A at state i}) = 1 - \frac{i}{2} = \frac{2-i}{2}$$

$$\mathbb{P}(\text{red extraction in B at state i}) = 1 - \frac{i}{3} = \frac{3-i}{3} \qquad \mathbb{P}(\text{white extraction in B at state i}) = \frac{i}{3}$$

Given this considerations we obtain the stochastic matrix:

$$P = \begin{pmatrix} 0 & 1 & 0 \\ \frac{1}{6} & \frac{1}{2} & \frac{1}{3} \\ 0 & \frac{2}{3} & \frac{1}{3} \end{pmatrix}$$

Which is a stochastic matrix because all rows sums up to 1.
Now the ansatz is the same as before. To evaluate the occurrence probability of state a_2 after 3 events we consider the element P^3_{13} (note that our indexes are shifted by 1).
We obtain $P^3_{13} \simeq 0.28$, while in the limit of many events $P^\infty_{13} = 0.3$

Lecture 5: Ising 2D

The Ising model is often regarded as the undisputed benchmark for Monte Carlo simulations. Its straightforward nature, paired with its complex and diverse behavior, makes it an excellent platform for testing and developing critical computational techniques like the Metropolis algorithm, which we will explore shortly.

In this exercise, we will simulate the 2D Ising model on a square lattice using Glauber dynamics (local spin flips) and the Metropolis algorithm to evolve the system configuration. We will then calculate key system observables at equilibrium across various temperatures and system sizes.

1.0.1 Introduction

We recall briefly some fundamental theoretical facts about the Ising model that we'll use through the simulation. We begin considering a generic lattice where we place N spins of the type $\sigma_i = \pm 1$. If one fixes an origin on the lattice, and consider from this the set of positions $\{\bar{r}_0, \bar{r}_1, \dots \bar{r}_{N-1}\}$ of the other lattice points, then is possible to define a configuration $\mathcal{C} = \{\sigma_{\bar{r}_0}, \dots \sigma_{\bar{r}_{N-1}}\}$ as the set of spin values assumed on each lattice point.
We define the energy of configuration $\mathcal{C}$ as:

$$\mathcal{H}(C) = -J \sum_{\langle \bar{r}_i \bar{r}_j \rangle} \sigma_{\bar{r}_i} \sigma_{\bar{r}_i}$$

Where the symbol $\sum_{\langle \bar{r}_i \bar{r}_j \rangle}$ implies that the sum is done only over the nearest neighbors. In our simulation we'll consider for simplicity $J = 1$.

In the case of the square lattice, this model can be solved exactly in the limit of an infinite lattice [3].
The result for the critical temperature and the expected magnetisation in function of the temperature are reported in the following. We have that the magnetization has the following function:

$$M = \begin{cases} (1 - \sinh^{-4}(2\beta J))^{1/8} & T < T_c \\ 0 & T \geq T_c \end{cases}$$

Where holds for the critical temperature the relation:

$$\frac{k_B T_c}{J} = \frac{2}{\ln\left(1 + \sqrt{2}\right)} \simeq 2.269$$

Again, to simplify our simulation, we will consider $k_B = 1$ (natural units).

Now we can begin discussing the simulation details .
We begin by generating a random configuration at the start of the simulation such that the probability to have a spin up or a spin down is $\frac{1}{2}$. With this choice, if the system evolves with $T < T_C$ can either

reach an equilibrium configuration where all spins are up or down , according to the phase diagram. Notice that if we, for whatever reason, would like to reach an equilibrium configuration respect to the other we can do that by skewing the picking probability in favor of what we desire.
Then we go from this configuration $\mathcal{C}$ to a new configuration $\mathcal{C}'$ performing N spin local moves, where the algorithm to perform a single spin flip and decide to accept it or not, is described in the following:

Algorithm 1 Metropolis Spin-Flip Dynamics

$(i, j) \sim \mathrm{U}(0, \mathrm{length})$
neighbor_sum $\leftarrow \sum$ (old_configuration[neighbor] for each neighbor in neighbors_list[(i, j)])
$\Delta E \leftarrow 2J \times$ old_configuration$[i, j] \times$ neighbor_sum
new_configuration $\leftarrow$ old_configuration
new_configuration$[i, j] \leftarrow -$old_configuration$[i, j]$ ▷ Flip the spin at (i, j)
if $\Delta E \leq 0$ **then**
    Return new_configuration
else
    $u \sim U(0, 1)$
    if $u \leq e^{-\beta \Delta E}$ **then**
        Return new_configuration
    else
        Return old_configuration
    end if
end if

Notice that in order to reduce the computation time a neighbor list for each atom, considering the periodic boundary condition, is evaluated using an hash table at the start of the program, so that the algorithm can use it as an input. This can be implemented easily in Python using a dictionary.
In our simulation we'll define 1 Monte Carlo timestep as the number of steps in which we perform N local spin flip moves.

1.0.2 Warm-up run

Given that we're starting from a random configuration we'll need some time before reaching the thermal equilibrium, i.e. the system we'll need to evolve for a certain number of Monte Carlo timesteps that we'll call τ_{eq} before we start to evaluate any mean observable.
This process is called thermalization and is a fundamental part in any simulation of this kind.
There are different recipes to find τ_{eq} for a given system, either qualitative or quantitative. One possible way can be to plot the observables of interest against time and see qualitatively when they tend to stabilize. One can, on top of this, introduce a quantitative criterion, say that the variance per observable must be less than a predetermined value α. This approach, even if simple, has a clear caveat : it establish a lower bound for the observables time series but not an upper bound.
Another approach is to evaluate at this stage the autocorrelation time for each observable. If the process is stationary or weakly stationary this number is well defined through the time series ($C_O(s, s+t) = C_O = (0, t)\ \forall s, \forall O$ for these kind of processes). Then we can use this number as a discard criterion (the common recipe is $\tau_{eq} \simeq 20\tau$, see [4]) but moreover we can use it in establishing an upper bound for the simulation.
We'll discuss in detail in the autocorrelation paragraph that errors for correlated measures are of the order $\sim (\frac{\tau}{N})^{1/2}$, so in order to achieve, for example, an error of the order ~ 0.01 for our measurements or better, $N > 10000\tau$.

A point of interest is that, technically, there is no need to discard any data. The initial data (biased estimation) leads to a systematic error $\sim \frac{\tau}{N}$, while the statistical errors will be of the order $\sim (\frac{\tau}{N})^{1/2}$, which is larger. In practice, given that the systematic error can be big, and we can let it be 0 with no cost, we discard the initial transient.

We then start the thermalization process with a warmup run where the maximum number of iterations is not the final one, and plot for each temperature and for each square lattice the 2 observables of interest (energy and magnetisation per spin) against the simulation time.

Here we have 2 possible roads: the quantitative one, i.e. if an observable stabilize to a constant value (apart from the statistical noise) we assume that from there on the process is stationary, or the quantitative one, i.e. to effectively prove that the time series is, from some τ_{eq} on , stationary or weakly stationary.
In order to prove this we can naively see numerically if the time series has constant mean, constant variance, and constant autocorrelation time in a certain tolerance range. While the implementation of this procedure is straightforward, is extremely susceptible to numerical instability, so in the code just a proof of concept version is presented, because its optimization is behind the scope of this analysis.
If one want to use a more robust and reliable way to prove this, then an Augmented Dickey-Fuller (ADF) test should be performed on the time series.
Due to time limitation, we opted this time for the qualitative approach.

Once the time series are evaluated for each lattice, we discard the initial transient and evaluate for each observable the autocorrelation time, and then take the maximum value between the 2. These τ, that are of the same order of the final τ values reported in the tables 1.1, 1.2, 1.3 where used to estimate how much the final simulation could run, i.e. the final upper bound and to tune for each temperature the minimum τ_{eq} possible

1.0.3 Autocorrelation

In Monte Carlo simulations, successive samples for each observable are often correlated due to the nature of the sampling algorithm.
How much they are correlated is described by the autocorrelation function which can be showed that for large t goes to 0 as an exponential:

$$C_O(T) \sim e^{-t/\tau^O_{exp}}$$

Where τ^O_{exp} is called autocorrelation time.
In literature [6] one can also find the definition of the so called integrated autocorrelation correlation time:

$$\tau^O_{int} \equiv \frac{1}{2} + \sum_{t=1}^{N} \frac{C_O(t)}{C_O(0)}$$

This definition, i.e. the factor $\frac{1}{2}$, is a matter of convention [4] that makes $\tau^O_{exp} \sim \tau^O_{int}$, and moreover one can prove [6] that one is the upper bound of the other, specifically $\tau^O_{int} \leq \tau^O_{exp}$.
In the case of correlated data one should correct the variance by the number of uncorrelated data [6] :

$$\sigma^2(O)_{corr} = \frac{\sigma^2(O)}{\frac{N}{2\tau^O_{int}}}$$

In our analysis, even if we recognize that these 2 times technically play different roles in the analysis, we just evaluate τ^O_{exp}, and take this as an overestimate of τ^O_{int}. So, from now on $\tau^O_{int} \sim \tau^O_{exp} \equiv \tau_O$

1.0.4 Simulation

In our simulation, we found correlation times of different orders of magnitude, as seen in tables 1.1, 1.2, 1.3, where we have an higher correlation time when we are near the critical temperature for each dimension of the square lattice, due to the critical slowing down phenomenon.
In order to find a tradeoff between computation time and accuracy, we choose, $t_{max} = 10^5$. This is a good choice in general but leads in the worst case scenario, i.e. $\tau \sim 400$, to a $t_{max} = 250\tau$ and subsequently to an error $\sim (\frac{\tau}{t_{max}})^{1/2}$ of the order of 0.2, which for us is terrible giving that we're estimating quantities of the same order.
In order to mitigate this one should propose a simulation of the order $t_{max} = 10^6 - 10^9$, to be safe.

We now comment the results obtained in tables 1.1, 1.2, 1.3. Note that all observables are evaluated per spin and the heat capacity C_V and the magnetic susceptibility χ_M are evaluated via block averaging (5) to be able to estimate their errors.

- **Magnetisation per spin** : We see for each dimension a transition from a completly magnetized state to a non-magnetized one.
  In the neighborhood of the critical temperature for each dimension the lower bound ($0.95\ T_c$) is consistent with the theoretical prediction and the upper bound ($1.05T_c$) can be considered consistent given the premises on the large error made at the start of the paragraph.

- **Energy per spin**: We have that the lowest energy state for the square lattice Ising model is $E = -2JN$ (all spins up) , while if all the spins are random ($T \to +\infty$), we have $E \simeq 0$. We see clearly this trend respected for all dimensions.

- **Heat capacity per spin**: We see that we obtain a result that is almost invariant by the choice of the dimension. This may seems an error at first but we recall that we are analyzing a finite size Ising model, and this has the effect to shift the T_c from the theoretical results when the dimensions changes.
  To have a proper description of the system further measure at different temperatures should be taken.

- **Magnetic susceptibility per spin**: The same consideration discussed for the heat capacity applies also to this observable.

Lecture 6: Polymer MMC

We want in this exercise simulate the absorption of a grafted polymer due to an absorbing hard wall. We will use in a first istance a regular Monte Carlo based on the Metropolis filter and secondly we will use the parallel tempering method, also known as multiple Markov chain (MMC) method.

Introduction

We want to recreate on a 2D plane the condition typical of a SAW (self avoiding walk) model, without the restriction that the monomers have to occupy a lattice point.
This can be achieved easily if in the displacement function the conditions of non-compenetration, bond length preservation and the hard wall constraint are respected.

A technical note here: the main way here in which these conditions can be implemented is via some if statement, which are notorious for slowing the routine (they create branches in the assembly instructions). For this reason the displacement function scales badly with number of monomers N, and for this reason we choose $N = 20$, which is a relatively small number when considering polymeric chains. An optimization improvement that can be proposed is to use a technique called branchless programming, rewriting these statements using logical operators. Note that this is not always faster because, if the compiler optimization are turned on , the compiler can sometimes optimize the assembly instruction better than the programmer. So one should check the assembly instruction to verify if a branch has been create or removed. This optimization was not performed in this code because outside of the scope of the analysis.

The choice of the possible trial displacements is at discretion of the algorithm designer, the only constraint is that the underlying Markov chain must be ergodic. One possible way to achieve this is to choose a trial displacement that is symmetric so that after a trial move that makes the system goes from $\mathcal{C} \to \mathcal{C}'$ there is a non zero probability that the system can go from $\mathcal{C}' \to \mathcal{C}$.
Specifically we choose as a move to perform N local gaussian shift of each monomers choosen from the distribution $\mathcal{N}(0, \sigma^2_{max})$ and one pivot rotation around a randomly choosen monomer (excluding the last one) of an angle choosen from a uniform symmetric distribution of the type $U(-\vartheta_{max}, \vartheta_{max})$.
The displacement function is extremely dependent the parameters σ_{max} and ϑ_{max} of these 2 distribution, specifically its performance can vary of orders of magnitude if the parameters are selected incorrectly. We choosed to have small parameters ($\sigma_{max} = r_{monomer}/8$ and $\vartheta_{max} = \pi/6$) and to investigate longer periods of time. The rationale is that one does small moves and doesn't reach the equilibrium as fast as in the situation of more sudden moves, but when the equilibrium is reached the small moves leads to less rejections in the geometry constraint part, and in total to a way faster move.

Monte Carlo evolution with Metropolis filter

Having decided how a trial move is defined for our system, we select a set of inverse temperature β_k and for each of them we evaluate for $t = 100000$ timesteps we evaluate the energy, the end to end distance, the end to height distance and the gyration ratio, with the relative error corrected by the autocorrelation time τ. All consideration about this last quantity are the same made on the last exercise.
In order to correctly evaluate τ for each observable we begin by discard $\tau_{equilibrium}$ and once the τ's are found we redefine our discard time.
After this warmup run is performed to estimate τ and to prime our intuition on the system, we then perform a final run with $t_{max} = 10^6$ and $\tau_{equilibrium} = 300000$. We choose such an high discard time because moves for high β , i.e. in the absorption phase, are extremely correlated. We report our data in table 1.4 and the evolution of the observables just for selected values of the inverse temperatures in figure 1.15.

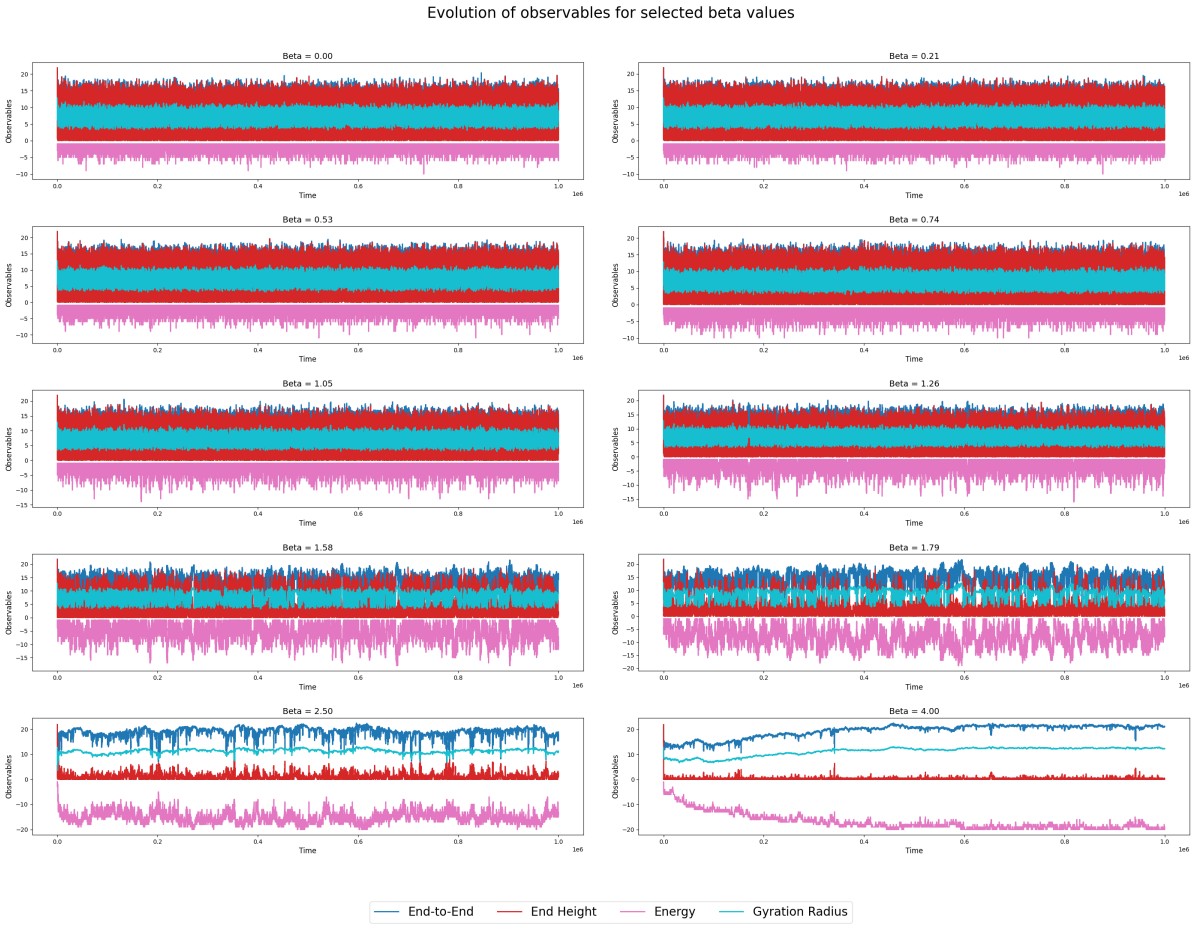


Figure 1.15: Observables evolution in time for selected values of β

First we can clearly see the existence of 2 regimes from figure 1.15: one of low β (high temperature), in which the grafted polymer floats freely and is not absorbed, and one of high β (low temperature), in which the polymer is completely absorbed.
The β_c is the one in which the energy exhibits the highest variance, which is in our case $\beta_c \simeq 1.79$.
We observe that the absorption mechanism, in the region $\beta \geq \beta_c$ leads by its nature to an high

correlation time.
This can be explained intuitively by the fact that if a move make the last monomer "attaches" to the absorbing wall, then the system goes in a metastable state where it tries to "detach" and "reattaches" better. Quantitatively this can be seen by the gyration radius that in this temperature regime stays for long periods in the "clamped" configuration and then suddently "relaxes".
Also this implies that for this system is very difficult, with this evolution method (the Metropolis filter), to establish a poor/good solvent limit in terms of temperatures, contrarely as what is seen typically in a SAW simulation [5], because of the existence of these metastable states that increases the autocorrelation time, and thus the error.

1.0.5 Parallel tempering method

Now we explain why we choose this specific set of β_k in the first place. We want to perform a parallel tempering simulation, so is extremely important that the energy histograms of neighbouring beta present a non negligible overlap [2] between them.
We choose this overlap to be $\gtrapprox 20\%$. Note that in the low β region we achieve this easily, while in the high β region this is difficult to achieve for the existence of the discussed metastable states. This is why just choosed less β in this regime and more spaced out than the ones in the low beta regime, to achieve our overlap constraint.
We plot in figure 1.16 the energy histogram overlap.

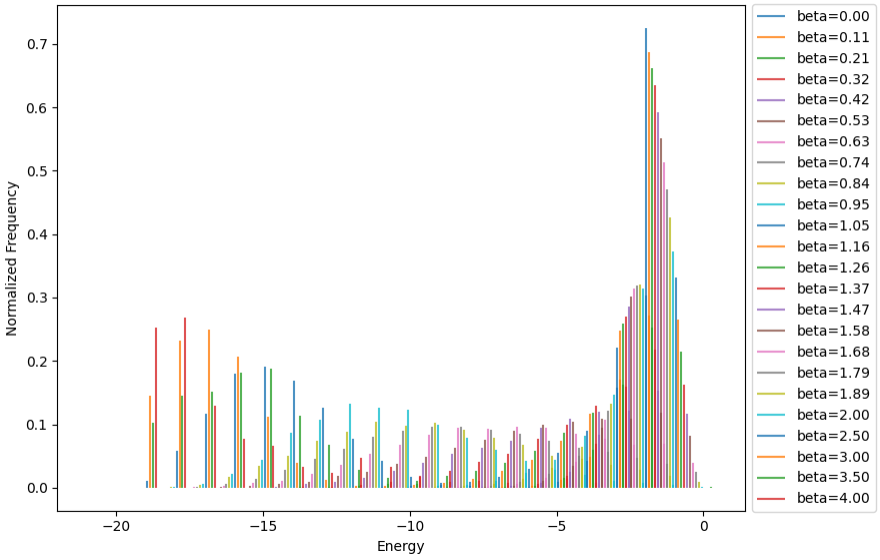


Figure 1.16: Energy histograms for different values of β

In the parallel tempering algorithm one must choose a decorrelation time $\tau_{decorrelation}$ after which the chain swap is attempted. Ideally one should choose the maximum of the set of of all correlation times for all the observables, in our case $\tau_{decorrelation} \simeq 100000$, to ensure that every observable is decorrelated. Instead we opted for $\tau_{dec} \simeq 25000$, given that just 1 chain has $\tau \sim \tau_{decorrelation}$, while the 3 chains have $\tau \sim \tau_{decorrelation}/2$ Given that the chain to swap is choosen uniformly, the probability that we have 2 runs without encountering the high τ chains is $\left(1 - \frac{4}{24}\right)^2 \sim 70\%$ so that there is enough time on average for the system to decorrelate. We preferred a smaller decorrelation time to attempt more swaps through the run.

The run is performed with $t = 10^6$ total steps, the result are reported in table 1.5, and the plot for selected β for all the observables is in figure 1.17.

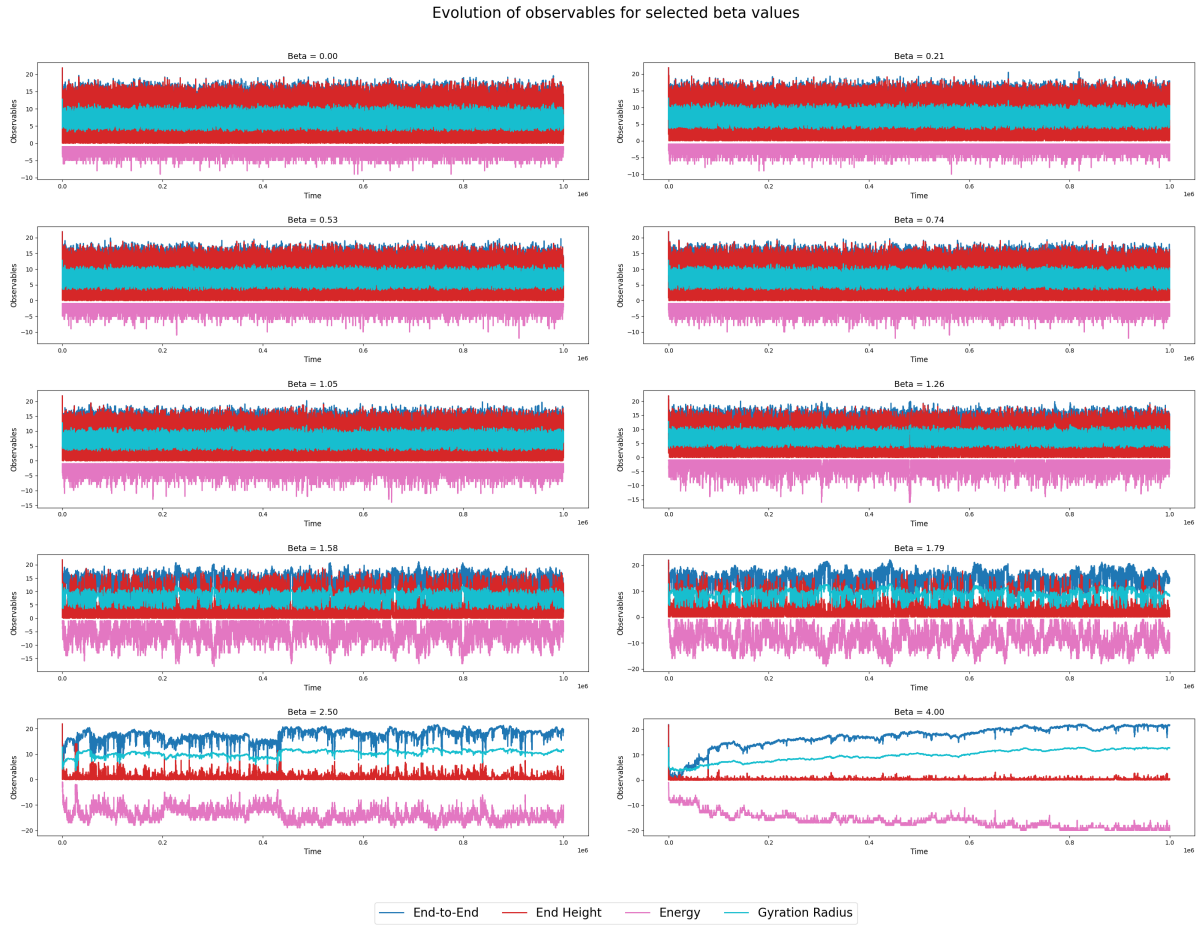


Figure 1.17: Observables evolution in time for selected values of β for the parallel tempering run

We have a reduction of the autocorrelation time for values of β where we expect with high probability the presence of metastable states, as expected [5]. The only outlier is $\beta = 4.00$ where we find an higher τ than the previous attempts. This can be due to the low decorrelation time we setted in order to achieve a tradeoff between number of swaps and total length of the simulation. In an ideal simulation we would propose $\tau_{decorrelation} \simeq 100000$ and make simulations of the order of $10^8 - 10^9$ total steps to see if this problem persists, but this is outside the scope of this report.
Finally we can plot each observable against the beta values. We add to the already discussed observables the evaluation of the heat capacity done via block averaging, in order to address the problem with the high correlation time in the $\beta > \beta_c$ region. The algorithm for block averaging is reported in 5. We can observe the plot in figures 1.18, 1.19, 1.20, 1.21 and 1.22

In all plots, we observe the signature of the critical temperature at approximately $\beta_c \simeq 1.79$.

- **Energy (Fig. 1.20)**: The plot clearly shows a transition from the floating state, characterized by a near-zero energy, to the totally absorbed state, where the energy approaches -20 for our specific polymer. The critical temperature β_c corresponds to the inflection point of this curve, marking the transition between these two phases.

- **End height (Fig. 1.18)**: In the floating state, the polymer maintains a non-zero height, roughly at half its maximum height. In contrast, the totally absorbed state is marked by a mean height close to zero. The critical temperature β_c is again identified as the point where the concavity of the curve changes.
- **End to end distance (Fig. 1.19)**: In the floating state, the polymer due to its constant movement maintains an end to end distance again roughly close to at half its maximum height. In contrast, in the totally absorbed state the polymer is completely stretched on the absorbing wall, leading to an end to end distance roughly close to the polymer length. Again the critical temperature shows the same properties of the other graphs.
- **Gyration radius (Fig. 1.21)**: In the floating state, the polymer due to its constant movement looks like it's "clamped", so its gyration radius its lower than the absorbed phase, where is totally stretched. Again the critical temperature shows the same properties.
- **Heat capacity (Fig. 1.22)**: In this graph, given that we are implictly plotting the energy variance, we should find that the β_c presents the maximum heat capacity. The observation is consistent with all the other graphs.

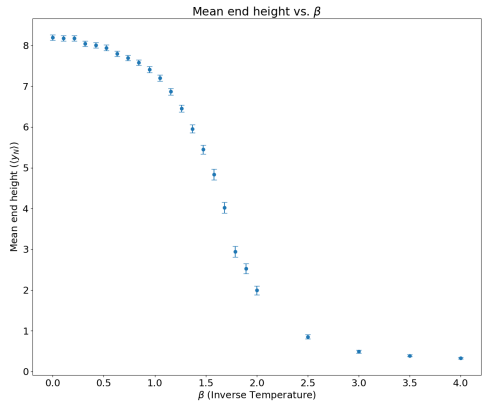


Figure 1.18: End height as a function of β for a polymer of length $N = 20$

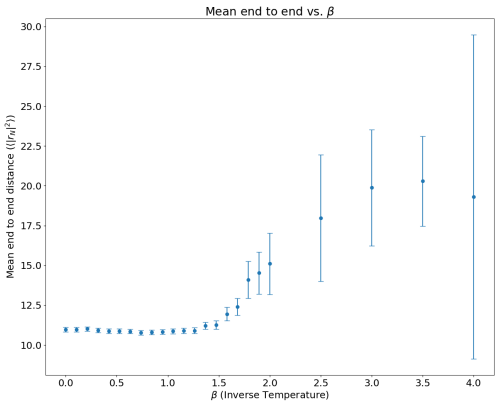


Figure 1.19: End to end distance as a function of β for a polymer of length $N = 20$

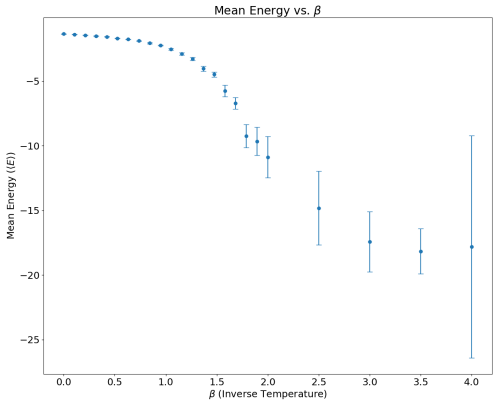


Figure 1.20: Energy as a function of β for a polymer of length $N = 20$

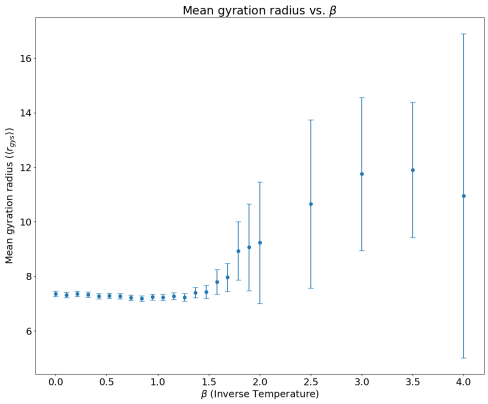


Figure 1.21: Gyration radius as a function of β for a polymer of length $N = 20$

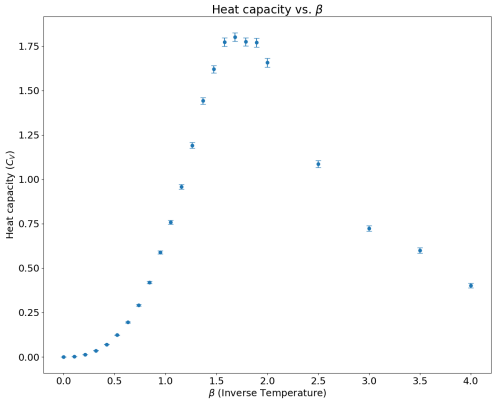


Figure 1.22: Heat capacity as a function of β for a polymer of length $N = 20$

Lecture 7: Brussellator

The Brussellator is a model that describes an autocatalytic and oscillatory chemical reaction. In literature is given as follows:

$$\begin{cases} A \xrightarrow{K_1} X, & \text{(1a)} \\ B + X \xrightarrow{K_2} Y + D, & \text{(2a)} \\ 2X + Y \xrightarrow{K_3} 3X, & \text{(3a)} \\ X \xrightarrow{K_4} E. & \text{(4a)} \end{cases} \tag{1.1}$$

Notice that for the reaction dynamics, only the third equation is important, i.e. the autocatalysis reaction $2X + Y \rightarrow 3X$, while the other equations describe ,in order, the X production (1a), the $X \rightarrow Y$ conversion (2a), the Y depletion (4a).
So, in a context in which we are interest in simulating just the autocatalysis dynamics and not the system details, we can reduce the set of reactions to the following one:

$$\begin{cases} \emptyset \xrightarrow{k_1} A_X \\ A_X \xrightarrow{k_2} \emptyset \\ 2A_X + A_Y \xrightarrow{k_3} 3A_X \\ A_X \xrightarrow{k_4} A_Y \end{cases}$$

Where with A_X, A_Y we denote molecules of two different species.
The only hypothesis one has to make is that the molecules A and B are in high concentration so that the reactions (1a) and (2a) can happen with no problem.
This implies automatically that our system will perform limit cycles and the stationary state will be unstable and oscillatory.
This because, from the stability analysis of 1.1, we have that the system is unstable if the following condition is satisfied:

$$K_2[B] - K_4 - K_3[X_{eq}]^2 > 0$$

Which is satisfied for an high B concentration.

Instead of going through the deterministic route, we can consider the stochastic version of these reactions, transforming in this way the deterministic dynamics into a Markov process.

We denote from now with $\mathcal{C} = (X, Y)$ a state of the system where X and Y represents the number molecules of the two species A_X and A_Y respectively. The transition rates for a state $\mathcal{C} = (X, Y)$ in this process are given by the equations:

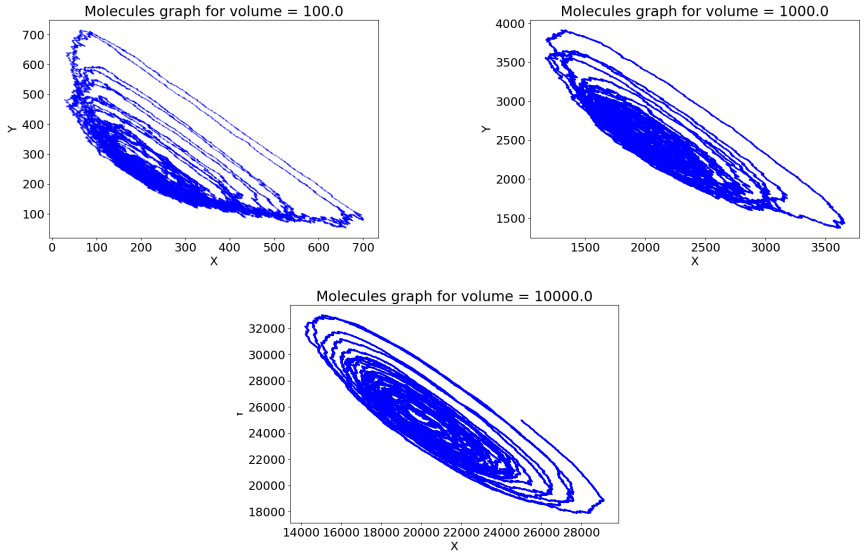


Figure 1.23: Comparison between the molecules graphs for different volumes

$$\begin{cases} w_1 = a\Omega & \text{for} \quad (X,Y) \to (X+1,Y) \\ w_2 = X & \text{for} \quad (X,Y) \to (X-1,Y) \\ w_3 = \frac{1}{\Omega^2} X(X-1)Y & \text{for} \quad (X,Y) \to (X+1,Y-1) \\ w_4 = bX & \text{for} \quad (X,Y) \to (X-1,Y+1) \end{cases}$$

Notice that these equations introduces the finite size of the system via a volume Ω. We study in the following the effects on evolution for different volumes $(\Omega = 10^2, 10^3, 10^4)$, given $a = 2$, $b = 5$. We report in figure 1.23 the molecules orbits for the different volumes.

As expected we can see we have a limit cycle for each volume, and moreover we expect an oscillatory, described by picture 1.24.

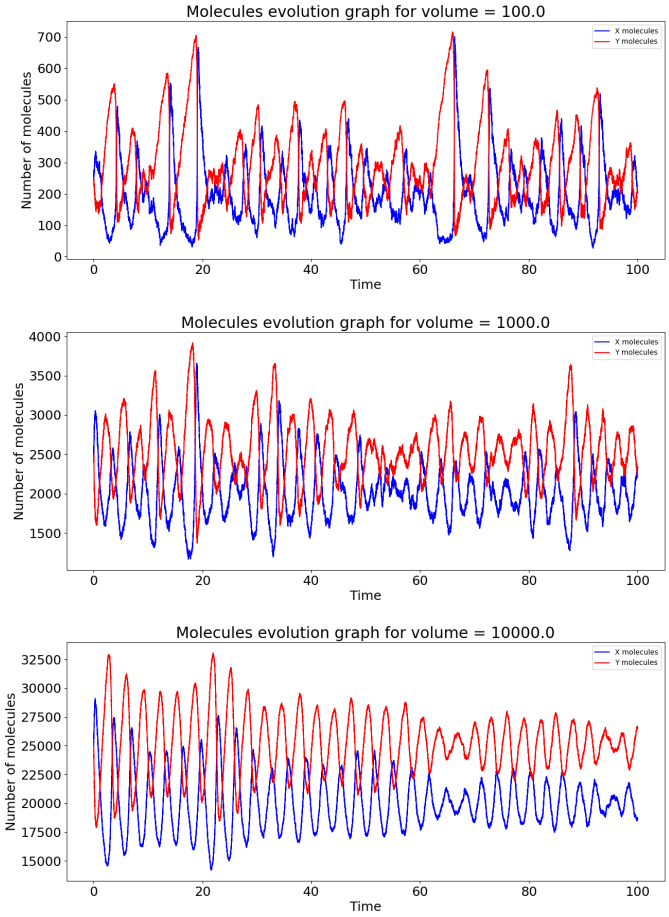


Figure 1.24: Comparison between the molecules graphs for different volumes

Notice that the volume has a regularizing effect on the oscillations.

Data tables

Temperature	τ_{energy}	$\tau_{\text{magnetisation}}$	Energy per spin	Magnetisation per spin	C_V per spin	χ_M per spin
$0.50T_c$	1.140 ± 0.006	1.166 ± 0.006	-1.992 ± 0.007	0.998 ± 0.004	0.0492 ± 0.0004	0.00395 ± 0.00003
$0.95T_c$	9.681 ± 0.040	52.468 ± 0.200	-1.604 ± 0.022	0.831 ± 0.027	1.146 ± 0.025	2.031 ± 0.360
$1.05T_c$	36.246 ± 0.151	413.969 ± 1.696	-1.251 ± 0.034	0.041 ± 0.045	1.519 ± 0.029	57.797 ± 2.635
$2.50T_c$	0.561 ± 0.004	1.657 ± 0.008	-0.3716 ± 0.0005	0.0002 ± 0.0004	0.0723 ± 0.0004	0.427 ± 0.002

Table 1.1: Autocorrelation time and selected observables for the square lattice with side $L = 25$

Temperature	τ_{energy}	$\tau_{\text{magnetisation}}$	Energy per spin	Magnetisation per spin	C_V per spin	χ_M per spin
$0.50T_c$	1.152 ± 0.006	1.173 ± 0.006	-1.992 ± 0.007	0.998 ± 0.004	0.04952 ± 0.00031	0.00396 ± 0.00003
$0.95T_c$	11.396 ± 0.051	78.375 ± 0.295	-1.602 ± 0.024	0.829 ± 0.033	1.197 ± 0.031	2.693 ± 0.525
$1.05T_c$	23.328 ± 0.107	401.713 ± 2.027	-1.227 ± 0.027	0.029 ± 0.027	1.254 ± 0.033	78.340 ± 6.934
$2.50T_c$	0.578 ± 0.004	1.689 ± 0.008	-0.3715 ± 0.00048	0.00001 ± 0.00018	0.07278 ± 0.00027	0.431 ± 0.003

Table 1.2: Autocorrelation time and selected observables for the square lattice with side $L = 50$

Temperature	τ_{energy}	$\tau_{\text{magnetisation}}$	Energy per spin	Magnetisation per spin	C_V per spin	χ_M per spin
$0.50T_c$	1.140 ± 0.006	1.158 ± 0.006	-1.992 ± 0.007	-0.998 ± 0.004	0.04986 ± 0.00028	0.00399 ± 0.00002
$0.95T_c$	10.472 ± 0.045	50.320 ± 0.194	-1.603 ± 0.023	-0.830 ± 0.026	1.180 ± 0.017	2.055 ± 0.160
$1.05T_c$	18.292 ± 0.078	392.670 ± 1.778	-1.224 ± 0.023	-0.019 ± 0.012	1.186 ± 0.028	76.825 ± 6.088
$2.50T_c$	0.567 ± 0.004	1.675 ± 0.008	-0.3716 ± 0.00044	0.0001 ± 0.00009	0.0731 ± 0.00042	0.430 ± 0.003

Table 1.3: Autocorrelation time and selected observables for the square lattice with side $L = 100$

β	τ_{Energy}	τ_{ee2}	$\tau_{\text{End Height}}$	$\tau_{\text{Gyr Rad}}$	Energy	ee2	End Height	Gyr Rad
0.00	26.13 ± 0.04	51.57 ± 0.08	20.49 ± 0.03	58.07 ± 0.09	-1.34 ± 0.01	10.91 ± 0.14	8.20 ± 0.07	7.35 ± 0.10
0.11	26.97 ± 0.04	50.08 ± 0.08	20.31 ± 0.03	58.53 ± 0.09	-1.40 ± 0.01	10.99 ± 0.14	8.18 ± 0.07	7.36 ± 0.10
0.21	30.79 ± 0.05	50.65 ± 0.08	19.43 ± 0.03	59.69 ± 0.10	-1.44 ± 0.02	10.92 ± 0.14	8.10 ± 0.07	7.30 ± 0.10
0.32	34.26 ± 0.05	51.94 ± 0.08	20.62 ± 0.03	60.87 ± 0.10	-1.50 ± 0.02	10.86 ± 0.14	8.04 ± 0.07	7.29 ± 0.10
0.42	36.54 ± 0.06	51.02 ± 0.08	20.71 ± 0.03	61.19 ± 0.10	-1.58 ± 0.02	10.88 ± 0.14	7.98 ± 0.07	7.29 ± 0.10
0.53	44.43 ± 0.07	54.33 ± 0.09	21.13 ± 0.03	62.63 ± 0.10	-1.68 ± 0.02	10.90 ± 0.14	7.92 ± 0.07	7.27 ± 0.10
0.63	54.40 ± 0.09	55.76 ± 0.09	21.46 ± 0.04	62.83 ± 0.10	-1.75 ± 0.03	10.87 ± 0.14	7.84 ± 0.07	7.25 ± 0.10
0.74	59.33 ± 0.09	56.49 ± 0.09	23.10 ± 0.04	66.89 ± 0.11	-1.89 ± 0.03	10.86 ± 0.14	7.69 ± 0.07	7.26 ± 0.10
0.84	78.72 ± 0.13	53.75 ± 0.09	23.55 ± 0.04	65.54 ± 0.11	-2.02 ± 0.04	10.78 ± 0.14	7.60 ± 0.07	7.19 ± 0.10
0.95	97.89 ± 0.16	59.11 ± 0.10	27.38 ± 0.04	72.62 ± 0.12	-2.28 ± 0.05	10.80 ± 0.15	7.37 ± 0.07	7.20 ± 0.11
1.05	145.08 ± 0.23	59.53 ± 0.10	28.80 ± 0.05	79.86 ± 0.13	-2.49 ± 0.06	10.88 ± 0.15	7.20 ± 0.07	7.25 ± 0.11
1.16	219.78 ± 0.36	71.95 ± 0.12	36.57 ± 0.06	100.24 ± 0.16	-2.93 ± 0.09	10.81 ± 0.16	6.82 ± 0.08	7.24 ± 0.12
1.26	344.06 ± 0.57	90.97 ± 0.16	47.56 ± 0.08	139.87 ± 0.24	-3.36 ± 0.13	10.94 ± 0.18	6.47 ± 0.09	7.28 ± 0.15
1.37	396.47 ± 0.61	97.42 ± 0.16	56.69 ± 0.09	160.31 ± 0.25	-3.84 ± 0.15	11.09 ± 0.19	6.08 ± 0.09	7.38 ± 0.16
1.47	685.02 ± 0.99	195.12 ± 0.30	106.52 ± 0.17	389.16 ± 0.62	-4.88 ± 0.25	11.48 ± 0.28	5.31 ± 0.11	7.54 ± 0.26
1.58	1122.03 ± 1.55	450.79 ± 0.73	170.34 ± 0.28	1040.00 ± 1.51	-5.85 ± 0.38	11.99 ± 0.45	4.66 ± 0.13	7.81 ± 0.44
1.68	1666.58 ± 2.40	936.06 ± 1.50	236.11 ± 0.38	1982.23 ± 2.83	-7.13 ± 0.55	12.69 ± 0.68	3.91 ± 0.14	8.17 ± 0.63
1.79	2141.79 ± 3.06	1439.93 ± 1.98	345.93 ± 0.53	2831.62 ± 3.84	-8.88 ± 0.75	13.90 ± 0.92	3.10 ± 0.14	8.80 ± 0.81
1.89	2910.75 ± 5.04	2292.31 ± 3.48	344.38 ± 0.58	3555.72 ± 5.88	-10.13 ± 0.99	14.88 ± 1.24	2.53 ± 0.12	9.33 ± 0.96
2.00	4303.85 ± 6.59	3367.79 ± 5.53	469.52 ± 0.75	10337.49 ± 14.23	-11.22 ± 1.29	15.56 ± 1.55	1.87 ± 0.11	9.55 ± 1.66
2.50	5800.59 ± 7.09	4750.94 ± 5.82	613.59 ± 1.02	20320.74 ± 20.25	-15.83 ± 2.05	18.93 ± 2.22	0.91 ± 0.06	11.43 ± 2.76
3.00	8707.33 ± 13.97	8760.96 ± 13.12	655.00 ± 1.20	24802.97 ± 36.72	-17.87 ± 2.83	20.39 ± 3.23	0.55 ± 0.04	12.15 ± 3.24
3.50	70782.23 ± 85.95	91320.27 ± 136.40	703.07 ± 1.06	97780.70 ± 161.14	-17.55 ± 7.93	19.37 ± 9.94	0.38 ± 0.03	11.08 ± 5.90
4.00	19172.11 ± 24.64	31138.53 ± 36.00	685.41 ± 0.96	50974.24 ± 60.96	-19.01 ± 4.46	20.86 ± 6.23	0.33 ± 0.02	12.26 ± 4.68

Table 1.4: Values of different observables at various β values without parallel tempering for a run of 10^6 steps

β	τ_{Energy}	τ_{ee2}	$\tau_{\text{End Height}}$	$\tau_{\text{Gyr Rad}}$	Energy	ee2	End Height	Gyr Rad
0.00	25.06 ± 0.04	51.58 ± 0.08	19.99 ± 0.03	59.38 ± 0.10	-1.35 ± 0.01	10.97 ± 0.14	8.19 ± 0.07	7.36 ± 0.10
0.11	28.79 ± 0.04	50.88 ± 0.08	19.90 ± 0.03	62.46 ± 0.10	-1.39 ± 0.01	10.97 ± 0.14	8.18 ± 0.07	7.32 ± 0.10
0.21	30.95 ± 0.05	52.57 ± 0.09	20.41 ± 0.03	61.57 ± 0.10	-1.44 ± 0.02	11.01 ± 0.14	8.17 ± 0.07	7.36 ± 0.10
0.32	34.70 ± 0.05	50.34 ± 0.09	19.44 ± 0.03	61.26 ± 0.10	-1.51 ± 0.02	10.93 ± 0.14	8.04 ± 0.07	7.33 ± 0.10
0.42	36.78 ± 0.06	53.85 ± 0.09	20.48 ± 0.03	61.70 ± 0.10	-1.57 ± 0.02	10.87 ± 0.14	8.01 ± 0.07	7.28 ± 0.10
0.53	43.46 ± 0.07	53.06 ± 0.09	20.89 ± 0.03	62.71 ± 0.10	-1.68 ± 0.02	10.88 ± 0.14	7.94 ± 0.07	7.29 ± 0.10
0.63	49.52 ± 0.08	55.48 ± 0.09	22.14 ± 0.04	64.99 ± 0.11	-1.76 ± 0.02	10.86 ± 0.14	7.80 ± 0.07	7.28 ± 0.10
0.74	61.27 ± 0.09	53.27 ± 0.09	21.89 ± 0.04	64.16 ± 0.10	-1.89 ± 0.03	10.78 ± 0.14	7.69 ± 0.07	7.22 ± 0.10
0.84	74.68 ± 0.12	55.20 ± 0.09	23.86 ± 0.04	65.09 ± 0.10	-2.06 ± 0.04	10.80 ± 0.14	7.58 ± 0.07	7.19 ± 0.10
0.95	90.87 ± 0.14	56.99 ± 0.09	26.55 ± 0.04	72.88 ± 0.11	-2.23 ± 0.04	10.83 ± 0.14	7.41 ± 0.07	7.24 ± 0.11
1.05	137.19 ± 0.21	62.69 ± 0.10	30.56 ± 0.05	80.42 ± 0.13	-2.53 ± 0.06	10.87 ± 0.15	7.20 ± 0.08	7.24 ± 0.11
1.16	232.68 ± 0.34	72.60 ± 0.12	37.47 ± 0.06	100.09 ± 0.17	-2.88 ± 0.09	10.91 ± 0.16	6.87 ± 0.08	7.27 ± 0.13
1.26	342.32 ± 0.53	87.68 ± 0.14	45.98 ± 0.08	134.19 ± 0.21	-3.27 ± 0.12	10.91 ± 0.18	6.46 ± 0.09	7.24 ± 0.14
1.37	558.96 ± 0.85	128.56 ± 0.22	69.48 ± 0.12	221.59 ± 0.39	-4.03 ± 0.19	11.21 ± 0.22	5.96 ± 0.10	7.41 ± 0.19
1.47	498.01 ± 0.87	177.98 ± 0.30	89.88 ± 0.14	347.32 ± 0.61	-4.48 ± 0.20	11.26 ± 0.27	5.45 ± 0.11	7.44 ± 0.24
1.58	1589.84 ± 2.41	407.43 ± 0.69	156.39 ± 0.28	1134.02 ± 1.55	-5.74 ± 0.45	11.95 ± 0.43	4.83 ± 0.13	7.80 ± 0.45
1.68	1206.40 ± 1.73	583.09 ± 0.92	222.64 ± 0.35	1408.28 ± 2.19	-6.71 ± 0.44	12.41 ± 0.53	4.02 ± 0.13	7.96 ± 0.52
1.79	2846.30 ± 4.31	2241.52 ± 3.31	348.89 ± 0.58	4798.38 ± 6.32	-9.24 ± 0.90	14.10 ± 1.17	2.94 ± 0.13	8.94 ± 1.07
1.89	3890.37 ± 5.65	2744.47 ± 4.34	398.85 ± 0.65	10457.97 ± 13.63	-9.66 ± 1.09	14.52 ± 1.32	2.53 ± 0.13	9.07 ± 1.59
2.00	6833.30 ± 9.58	5439.60 ± 7.68	419.07 ± 0.63	19637.23 ± 21.55	-10.88 ± 1.60	15.10 ± 1.93	1.99 ± 0.11	9.24 ± 2.23
2.50	12592.04 ± 13.41	16879.90 ± 17.14	547.87 ± 0.83	29139.82 ± 28.82	-14.81 ± 2.84	17.97 ± 3.97	0.85 ± 0.05	10.66 ± 3.09
3.00	6187.15 ± 8.46	11717.34 ± 12.28	758.68 ± 1.09	19876.01 ± 28.55	-17.42 ± 2.32	19.89 ± 3.64	0.49 ± 0.04	11.76 ± 2.80
3.50	3217.54 ± 4.82	6798.91 ± 11.37	627.96 ± 0.91	15124.37 ± 21.81	-18.16 ± 1.74	20.29 ± 2.83	0.39 ± 0.02	11.91 ± 2.48
4.00	80743.95 ± 104.44	96385.29 ± 145.09	620.36 ± 1.01	101357.26 ± 164.05	-17.81 ± 8.59	19.31 ± 10.18	0.33 ± 0.02	10.95 ± 5.94

Table 1.5: Values of different observables at various β values with parallel tempering for a run of 10^6 steps

Appendix : algorithms

The notation on this appendix (tries to) follow the pseudo-code convention in algorithms in [1].

Algorithm 2 Inversion Sampling Algorithm

$S \leftarrow \emptyset$
count $\leftarrow 0$
while count $<$ N **do**
    Sample $U \sim \text{Uniform}(0, 1)$
    Compute $X \leftarrow F^{-1}(U)$ ▷ Apply the inverse CDF to transform U
    Append X to S
    count $\leftarrow$ count $+ 1$
end while
Return S

Algorithm 3 Rejection Sampling Algorithm

$S \leftarrow \emptyset$
count $\leftarrow 0$
while count $<$ N **do**
    Sample $X \sim g(x)$ ▷ Generate sample from candidate distribution
    Sample $U \sim \text{Uniform}(0, 1)$
    $A \leftarrow \frac{f(X)}{c \cdot g(X)}$ ▷ Calculate acceptance threshold
    if $U \leq A$ **then**
        Append X to S
        count $\leftarrow$ count $+ 1$
    end if
end while
Return S

Implementation detail : Note that to obtain exactly N samples a while loop must be used in the algorithm.
Moreover is interesting to discuss some technical details regarding the implementation of this algorithm in Python when the candidate distribution is a truncated distribution, or in general a distribution that cannot be generated easily but needs the `scipy.stats` package to be generated.
In this case a naive implementation of algorithm 3 can hinder the perfomance greatly, because even if one has in mind to speed up the function with a `numba` decorator, this will not work, due to known incompatibility issues within the 2 cited packages.
The easiest solution to this problem is to vectorize our algorithm using the `numpy` library, overproducing samples and then cutting what is not necessary. This approach has a clear caveat, we risk an `int` overflow, so we must put attention in not choosing an high value of N for this solution. If a big

number of samples is required, this approach must be revised.

Algorithm 4 Importance Sampling Algorithm

$\mu \leftarrow 0$
count $\leftarrow 0$
while count $<$ N **do**
    Sample $X \sim g(x)$ ▷ Generate sample from proposal distribution $g(x)$
    $\mu \leftarrow \mu + \frac{f(X)}{g(X)}\rho(X)$
    count $\leftarrow$ count $+ 1$
end while
Return μ/N

Algorithm 5 Block Averaging Algorithm

$S \leftarrow \{X_1, X_2, \ldots, X_N\}$ ▷ Set of N samples
Divide S into M blocks, each containing L samples ▷ Where $N = M \times L$
$B \leftarrow \emptyset$ ▷ List to store block averages
for each block i from 1 to M **do**
    Compute the block mean: $\bar{X}_i \leftarrow \frac{1}{L}\sum_{j=1}^{L} X_{(i-1)L+j}$
    Append $\bar{X}_i$ to B
end for
Compute the overall block average: $\bar{X}_{\text{block}} \leftarrow \frac{1}{M}\sum_{i=1}^{M} \bar{X}_i$
Compute the variance of the block means:

$$\hat{\sigma}^2_{\text{block}} \leftarrow \frac{1}{M-1}\sum_{i=1}^{M}(\bar{X}_i - \bar{X}_{\text{block}})^2$$

Return $\bar{X}_{\text{block}}$, $error_{\bar{X}_{\text{block}}} \leftarrow \sqrt{\hat{\sigma}^2_{\text{block}}/M}$

Bibliography

[1] Donald E. Knuth. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms*. Addison-Wesley, Reading, Mass., third edition, 1997.

[2] G Marinari, E; Parisi. Simulated tempering: A new monte carlo scheme. *EPL (Europhysics Letters)*, 19:451–458, 1992.

[3] Lars Onsager. Crystal statistics. i. a two-dimensional model with an order-disorder transition. *Physical Review (Series I)*, 65:117–149, 1944.

[4] A. Sokal. *Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms*, pages 131–192. Springer US, Boston, MA, 1997.

[5] M. C. Tesi, E. J. Janse van Rensburg, E. Orlandini, and S. G. Whittington. Monte Carlo study of the interacting self-avoiding walk model in three dimensions. *Journal of Statistical Physics*, 82(1-2):155–181, January 1996.

[6] Wolfhard Weigel, Martin; Janke. Error estimation and reduction with cross correlations. *Physical Review E*, 81:066701, 2010.