A stochastic root-finding algorithm via a biased random walk

Rodrigo de León Ardón,[a]

[a] *University of life,*
some where with my dogs in Guatemala
rodrigodla.blog

[1]This is not a publication it is (hopefully) a manuscript useful for some people interested on this subject. I use jheppub.sty LaTeX package for style.

Contents

1 Motivation

In this manuscript I want to answer the following question: given a function $f(x)$ does a value x_0 correspond to a root?

For some functions, the answer can be found more or less immediately if there is a known formula. We can also contemplate the Newton–Raphson method for finding roots for more complicated functions. The basic point is that the question entails finding the possible roots and then compare with x_0 to establish if it is a root or not.

Instead of using a deterministic approach, the goal is to use a probabilistic method. For simple functions this approach represents a very complicated way to do the job –a long detour– but it has its advantages. It provides a framework for other types questions such as: how can AI tell from a picture that a cat is a cat?

The fun part of this detour is that involves optimization, statistics and programming. These elements are developed in the manuscript with certain detail in order to provide a transparent solution. In the right perspective it can be thought as a basic introduction to MCMC (Markov chain Monte Carlo) but it should be noticed that my goal is different.

I find it exciting to answer a simple question in a "convoluted" way since it opens a different perspective of thinking. While developing this manuscript, I benefited from conversations with K. Lua and I. Lua, as well as from extensive and practical use of ChatGPT (GPT-5.3). By "extensive", I mean that over the years it has adapted to my constraints regarding mathematical knowledge and references (any hallucinations, if present, were identified and corrected). The core ideas were not suggested or created by ChatGPT; it only

provided information that is already known.[1] By "practical" I mean that the Python scripts were developed in a few seconds by ChatGPT (wonderful!). I wrote the document in LaTeX as usual (not online, I'm not that young or old) but with some help of ChatGPT in the wording.

My suggested references are:

- The video Diffusion and Score-Based Generative Models by Yang Song. Check his website for publications.
- The publication General state space Markov chains and MCMC algorithms by Gareth O. Roberts and Jeffrey S. Rosenthal.
- Nicholas Zabaras lectures. In particular: Lecture 33 - The Metropolis Hastings Algorithm.
- The nice video Markov Chain Monte Carlo Explained in 10 Minutes by Sydney Katz. Check her website for publications.
- The book Information Theory, Inference and Learning Algorithms by David J. C. MacKay. Cambridge University Press.

If the specific content of this manuscript is not useful for you maybe the above references will.

The subject of Markov chain was not covered in order to keep things as simple as possible. No attempt was made to give a physical interpretation (check HMCMC). The temptation was tremendous but I realized that it is not the way, statistical mechanics provides analogies only. Here we are just applying distributions in a fun way. Nevertheless, I do tend to secretly use a physicist interpretation (sorry, it is still embedded).

[1] It was very efficient, clear, and helpful. It significantly reduced research time and emulated a research environment to which I no longer have access.

2 Finding roots as a minimization problem

Let $f(x)$ be a real function with domain denoted by $\mathcal{X} \subseteq \mathbb{R}$. The problem is to find the roots of the function, i.e. the set $\{x_r \in \mathcal{X} : f(x_r) = 0\}$. A root has multiplicity k if

$$f(x_r) = f'(x_r) = \ldots = f^{(k-1)}(x_r), \quad f^{(k)}(x_r) \neq 0.$$

A root with multiplicity $k = 1$ is referred to as a simple root.

Consider $f(x) = x$, the function has a simple root $x_r = 0$ and thus $f'(0) = 1$. For $f(x) = x^2$ we have $x_r = 0$ has multiplicity $k = 2$. Then $f'(0) = 0$ and $f''(0) = 2$. If we instead consider $f(x) = \sin x$ the roots $\{n\pi\}_{n\in\mathbb{Z}}$ are simple thus $f'(n\pi) = (-1)^n$.

Let us now consider the function

$$E(x) = \varepsilon + \frac{1}{2}(f(x))^2.$$

We have that $E \geq \varepsilon$ where lower bound is saturated for all possible roots. Notice that we have transformed the problem of finding the roots of $f(x)$ to the problem of finding the minima of $E(x)$. Recall that in this framework a critical point is the solution of $E'(x_c) = 0$. The second derivative test tells us that

- if $E''(x_c) < 0$, the point $(x_c, E(x_c))$ corresponds to a local maximum,
- if $E''(x_c) > 0$, the point $(x_c, E(x_c))$ corresponds to a local minimum,
- if $E''(x_c) = 0$, the test is inconclusive.

Since in our case

$$E'(x) = f(x)f'(x), \quad E''(x) = (f'(x))^2 + f(x)f''(x),$$

we find for a simple root

$$E'(x_r) = 0, \quad E''(x_r) = (f'(x_r))^2, \quad k = 1.$$

That is, the simple root corresponds to a critical point and since $(f'(x_r))^2 > 0$ the point $(x_r, E(x_r))$ is a local minimum. For a root with multiplicity $k \geq 2$ we find

$$E'(x_r) = 0, \quad E''(x_r) = 0, \quad k \geq 2,$$

and the test is inconclusive. We must understand this last case in detail.

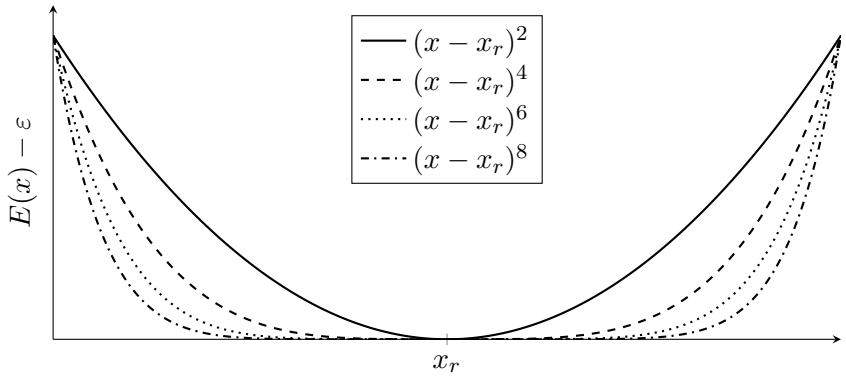
In order to do so, recall that around a simple root we have that $f(x) \approx f'(x_r)(x - x_r)$ and therefore $E(x)$ is approximately quadratic

$$E(x) \approx \varepsilon + \frac{1}{2}(f'(x_r))^2(x - x_r)^2.$$

Since $(f'(x_r))^2 > 0$ the graph of $E(x)$ correspond to a U-shape parabola centered at the root. Around a root with multiplicity $k \geq 2$ we have $f(x) \approx \frac{1}{k!}f^{(k)}(x_r)(x - x_r)^k$ and thus

$$E(x) \approx \varepsilon + \frac{1}{2(k!)^2}(f^{(k)}(x_r))^2(x - x_r)^{2k}.$$

Notice that the factor $\frac{1}{2(k!)^2}(f^{(k)}(x_r))^2 > 0$ and since $k \geq 2$ the graph now corresponds to a flat bottom and steeper walls compared to the quadratic well. We find that the critical point gives also a local minimum. This is depicted in the following figure:



3 Statistical approach to the minimization problem

Let us interpret the domain $\mathcal{X}$ of the function as a state space. A state corresponds to a point in $\mathcal{X}$. We can also consider the E-space defined as $\mathbb{R}_{\geq 0}$. An E-state corresponds to a point in $\mathbb{R}_{\geq 0}$. Root states are associated with the lowest E-states.

For a given E-state, multiple x-states may correspond to it. For example consider $f(x) = x$. Since $E(x) = \varepsilon + \frac{1}{2}x^2$ we see that $x = \pm 1$ have the same E-state. Hence, E-state can be degenerate and we denote $g(E)$ the possible number of x-states associated to the E-state. That is, $g(E)$ is the *degeneracy* of the E-state.

For a single simple root, like $f(x) = x$, we must have that the lowest E-state is not degenerate: $g(\varepsilon) = 1$. For a root with multiplicity $k = 2$, like $f(x) = x^2$, we also have a non degenerate lowest E-state but the form around the this state is flatter with respect to the simple root. For $f(x) = \sin x$, the lowest E-state is degenerate since we have an infinite set of simple roots for $\mathcal{X} = \mathbb{R}$.

The task is clear: given the state space, find the root states. This is equivalent to define a root state and this is straightforward: consider a state x_0, if $E(x_0) > \epsilon$ the state is not a root state and if $E(x_0) = \varepsilon$ the state is a root state. But how do we scan the whole state space? This is not trivial since the state space is continuous.

3.1 Probability density

Consider the following idea. Let X be a continuous random variable $X : \Omega \to \mathcal{X}$. Since the sample space $\Omega = \mathcal{X}$, X corresponds to an identity random variable $X(x) = x$. Let $\pi_\theta(x)$ be an unnormalized probability density function (pdf) where θ are a collection of parameters. Then $X \sim \pi_\theta$ and if A is a proper interval in $\mathcal{X}$, we have

$$\mathbb{P}(X \in A) = \frac{\int_{A \subset \mathcal{X}} \mathrm{d}x\, \pi_\theta(x)}{\int_{\mathcal{X}} \mathrm{d}x\, \pi_\theta(x)}, \quad 0 < \int_{\mathcal{X}} \mathrm{d}x\, \pi_\theta(x) < \infty,$$

which corresponds to the probability that the value of X lies in A.

Assuming only the existence of the pdf and roots states, $\mathbb{P}(X \in [x_r - \Delta x, x_r + \Delta x])$ (with Δx positive and small) corresponds to the probability that the value of X lies near a root. Since we know how to identify a root state by means of the energy, then it is natural to introduce the energy dependence via

$$\pi_\theta(x) = \Phi(\theta(E(x) - \varepsilon)).$$

We only have one parameter and the argument of the function Φ is chosen such that for a root state we obtain $\pi_\theta(x_r) = \Phi(0)$ for any fixed value of θ.

For convenience, we define $u_\theta(x) = \theta(E(x) - \varepsilon)$ so that we can write $\pi_\theta(x) = \Phi(u_\theta(x))$. It is straightforward to show that for a root with multiplicity k we have

$$\left.\frac{\mathrm{d}^k \pi_\theta}{\mathrm{d}x^k}\right|_{x=x_r} = \theta \left.\frac{\mathrm{d}\Phi}{\mathrm{d}u_\theta}\right|_{u_\theta=0} \left.\frac{\mathrm{d}^k E}{\mathrm{d}x^k}\right|_{x=x_r},$$

provided that higher derivatives of Φ at $u_\theta = 0$ are finite for any fixed value of θ. Since $\left.\frac{\mathrm{d}^k E}{\mathrm{d}x^k}\right|_{x=x_r} > 0$ we are going to set

$$\theta \left.\frac{\mathrm{d}\Phi}{\mathrm{d}u_\theta}\right|_{u_\theta=0} < 0,$$

for any fixed value of θ so that $(0, \Phi(0))$ corresponds to a local maximum. Then for several root states we see that π_θ is multimodal. Each possible root corresponds to a mode and since all have the same value, all the modes have the same height.

Assume that $\Phi(u_\theta) = \Phi(0)e^{u_\theta}$. Then we have $\frac{\mathrm{d}\Phi}{\mathrm{d}u_\theta} = \Phi$ and $\theta\Phi(0) < 0$. Choosing $\theta = -\beta$ for $\beta > 0$ we find that $\Phi(0) > 0$. Hence we propose

$$\pi_\beta(x) = e^{-\beta(E(x)-\varepsilon)},$$

which corresponds to multimodal unnormalized pdf with modes of unit height.

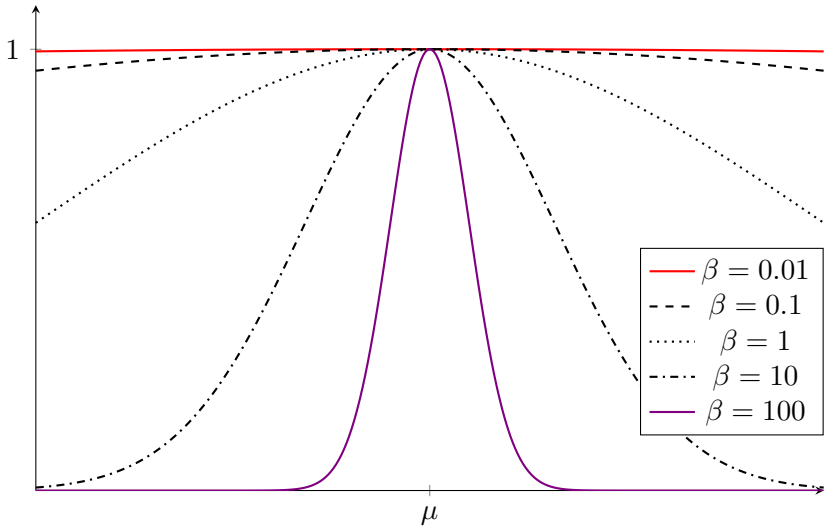
One virtue of this proposal is that from

$$\mathbb{P}(X \in A) = \frac{\int_{A \subset \mathcal{X}} \mathrm{d}x\, e^{-\beta(E(x)-\varepsilon))}}{\int_{\mathcal{X}} \mathrm{d}x\, e^{-\beta(E(x)-\varepsilon)}},$$

the contribution of the lowest E-state cancels out. Another virtue is that around a simple root state we have

$$e^{-\beta(E(x)-\varepsilon)} \approx e^{-\frac{\beta}{2}(f'(x_r))^2(x-x_r)^2},$$

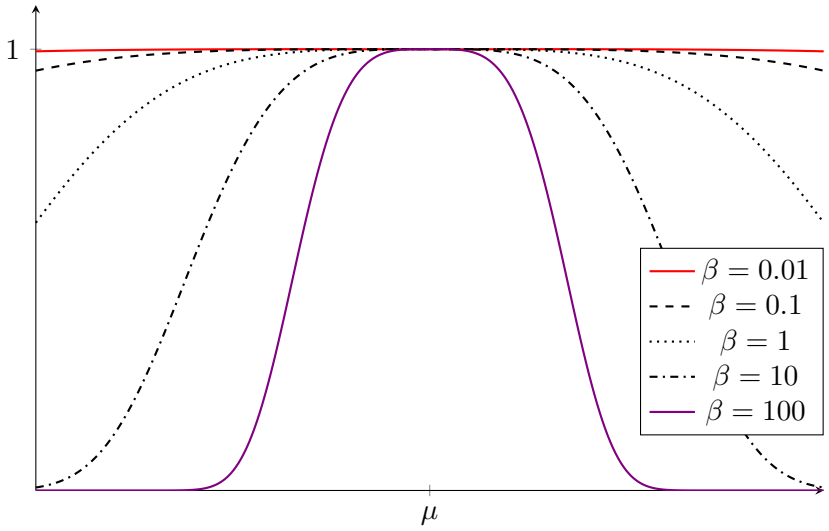
which corresponds to an unnormalized normal distribution with mean $\mu = x_r$ and variance $\sigma^2 = \frac{1}{\beta(f'(x_r))^2}$. In the following figure:



we see how β controls the width of the peak. It is important to notice that for large β the pdf localized around the root state . Around a root state with multiplicity k we have

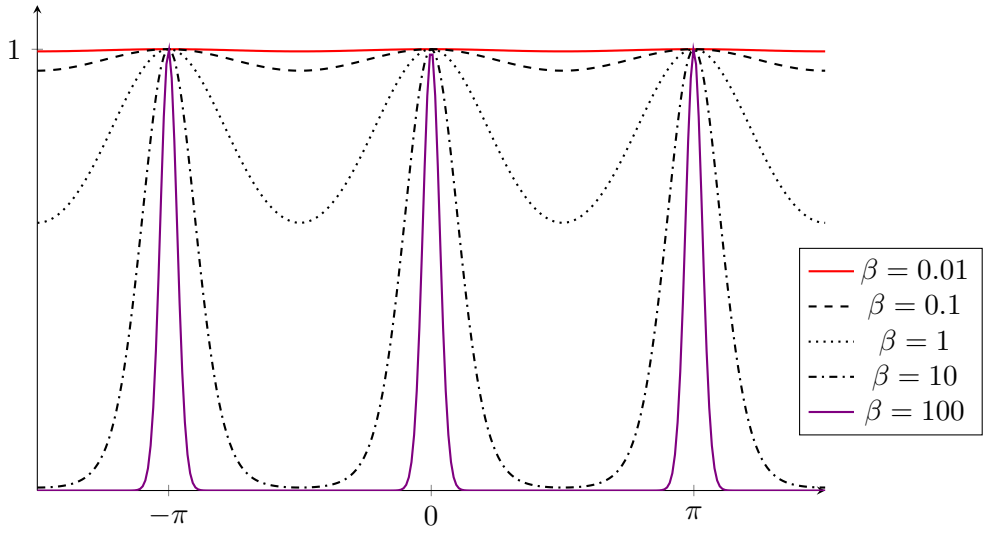
$$e^{-\beta(E(x)-\varepsilon)} \approx e^{-\frac{\beta}{2(k!)^2}(f^{(k)}(x_r))^2(x-x_r)^{2k}},$$

which does not has a normal form. Nevertheless from the following figure for $k = 2$:



we find a similar behavior as in the normal case but the shape of the peak is different.

For a degenerate lowest E-state case consider $f(x) = \sin x$ and $\mathcal{X} = [-3\pi/2, 3\pi/2]$. The behavior is depicted in the following figure:



From the proposal we have learned that regions of *localized* high probability density gives us information about the possible root states and the shape of how they are localized matters.

3.2 Unbiased random walk

In order to find the actual root states we need to develop further the idea. Consider a discrete "time" random walk in order to explore the continuous state space. The basic idea is that for given state x_i we want a rule that gives us another state x_{i+1}. This "jump" rule must take into account if we are in a high probability density region or low probability density region of π_β. That is, we must also have a selection rule which can be realized by analyzing low/high E-state regions[2].

First let us interpret that the exploratory walk is done in discrete "time" intervals $t_i = i\Delta t$ so that $t_{i+1} - t_i = \Delta t$. We assume $\Delta t > 0$ to that we can discuss a "time" evolution scenario and for simplicity we set $\Delta t = 1$. We say that x_i corresponds to a state at "time" i and x_{i+1} a state at "time" $i + 1$. Suppose the simple determinist rule

$$x_{i+1} - x_i = v_i.$$

If $i + 1$ is the current state, it only depends on the previous state x_i in addition to v_i. In order to incorporate a probabilistic rule for the walk we consider

$$X_{i+1} = x_i + V_i,$$

where V_i is a continuous identity random variable V_i with value v_i. Since the previous state is fixed, X_{i+1} is just a linear transformation of the random variable V_i. Let V_i follow the properly normalized $\wp_i$ which is defined over $\mathcal{X}$. We have that

$$P(V_i \in B) = \int_{B \subset \mathcal{X}} \mathrm{d}z \, \wp_i(z),$$

[2]The notions of low probability density regions and low/high E-state regions eventually must be formalized.

corresponds to the probability that the value of V_i lies in λ for all i. From the walk perspective, X_{i+1} follows $\wp_i$ and consequently the probability $P(X_{i+1} \in [a,b])$ is equal to

$$P(V_i \in [a-x_i, b-x_i]) = \int_{a-x_i}^{b-x_i} \mathrm{d}z\, \wp_i(z) = \int_a^b \mathrm{d}s\, \wp_i(s-x_i)\,,$$

which corresponds to the probability that the value of X_{i+1} lies in $[a,b]$ given that the previous state is fixed to x_i.

But if the walk is truly random, we must consider x_i as the value of a random variable X_i that does not follow $\wp_i$. A more fundamental reason for this consideration is that we need to be able distinguish between states and increments.

Recall the definition of the sum of two independent random variables X_i and V_i. Let X_i follow the properly normalized pdf q_i defined over $\mathcal{X}$ and we now that V_i follows $\wp_i$. Then, $X_{i+1} = X_i + V_i$ is another random variable whose pdf is the convolution of ρ_i and $\wp_i$, that is X_{i+1} follows the properly normalized pdf $\gamma_{i+1} = \rho_i * \wp_i$ (convolution of ρ_i and $\wp_i$). We have

$$\gamma_{i+1}(\sigma) = \int_{\mathcal{X}} \mathrm{d}\omega\, \rho_i(\sigma-\omega)\wp_i(\omega) = \int_{\mathcal{X}} \mathrm{d}\lambda\, \wp_i(\sigma-\lambda)\rho_i(\lambda),$$

and now the probability that the value of X_{i+1} lies in $[a,b]$is given by

$$\mathcal{P}(X_{i+1} \in [a,b]) = \int_a^b \mathrm{d}\sigma\, (\rho_i * \wp_i)(\sigma).$$

Since

$$\mathcal{P}(X_{i+1} \in [a,b]) = \int_{\mathcal{X}} \mathrm{d}\lambda\, \wp_i(\sigma-\lambda)\rho_i(\lambda) = \int_{\mathcal{X}} \mathrm{d}\lambda\, \left[\int_a^b \mathrm{d}\sigma\, \wp_i(\sigma-\lambda)\right] \rho_i(\lambda).$$

The term in the bracket corresponds to the conditional probability

$$\mathcal{P}(X_{i+1} \in [a,b] | X_i = \lambda) = \int_a^b \mathrm{d}\sigma\, \wp_i(\sigma-\lambda).$$

In summary we have

$$X_{i+1} = X_i + V_i, \quad X_{i+1} \sim \gamma_{i+1}, \quad X_i \sim \rho_i, \quad V_i \sim \wp_i, \quad \gamma_{i+1}(\sigma) = \int_{\mathcal{X}} \mathrm{d}\lambda\, \wp_i(\sigma-\lambda)\rho_i(\lambda).$$

Let us study the rule in detail. A the initial "time" we have the independent random variables X_0 and V_0 that follow ρ_0 and $\wp_0$ respectively. Then the sum $X_0 + V_0$ follows $\rho_0 * \wp_0$, that is $X_1 \sim \gamma_1$. For the next step we have $V_1 \sim \wp_1$ and the sum $X_1 + V_1$. Since X_1 and V_1 are independent by definition we obtain that $X_2 \sim \gamma_2$ with $\gamma_2 = \rho_0 * \wp_0 * \wp_1$.

Notice that this does not imply that V_1 is independent with respect X_0 and V_0. If we carry on, we will find for the n-th step that

$$X_n = X_0 + V_0 + V_1 + \ldots + V_{n-1}, \quad \gamma_n = \rho_0 * \wp_0 * \wp_1 * \cdots * \wp_{n-1},$$

but this does not imply that the collection $\{X_0, V_0, V_1, \ldots, V_{n-1}\}$ is jointly independent. Instead of assuming at the beginning that X_i and V_i are independent, we consider a stronger assumption: the collection $\{X_0, V_0, V_1, \ldots, V_{n-1}\}$ are jointly independent. This automatically imply that X_i and V_i are independent and that $\{V_0, V_1, \ldots, V_{n-1}\}$ are jointly independent. The last implication it will be useful for the exploration of the states.

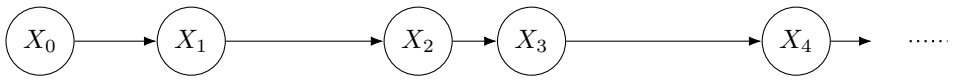
Now we are in position to discuss the basic logic of the exploration:

1. Define $\rho_0, \wp_0$ and sample a state x_0 from ρ_0 and an sample increment v_0 from $\wp_0$.
2. Compute $x_1 = x_0 + v_0$. Define $\wp_1$ and sample an increment v_1.
3. Compute $x_2 = x_1 + v_1 = x_0 + v_0 + v_1$. Define $\wp_2$ and sample an increment v_2.
4. Carry on until you define $\wp_{n-1}$ and sample an increment v_{n-1}.
5. Compute $x_n = x_0 + v_0 + v_1 + v_2 + \cdots + v_{n-1}$.
6. We obtain the sample $\{x_0, x_1, x_2, \ldots, x_n\}$.

The fundamental distributions to define are: ρ_0 and the collection$\{\wp_i\}$. Instead of considering the collection $\{\wp_i\}$ of pdf's we can simple interpret $\wp_i(z) = \wp(z; t_i)$ as a single object. That is, $\wp(t_i, z)$ corresponds to a discrete "time" dependent pdf which is referred as to the kernel. From

$$\mathcal{P}(X_{i+1} \in [a, b] | X_i = \lambda) = \int_a^b \mathrm{d}\sigma \, \wp(\sigma - \lambda; t_i),$$

we see that the kernel corresponds to a conditional probability density which is spatial translational invariant and "time" dependent. The conditional probability describes the probability of a "jump" from X_i to X_{i+1}. These "jump" or transition probabilities vary with the step i since the kernel is "time" dependent. This is depicted in the following figure:



where the arrows represent the "jump" or transition probabilities values.

3.3 Biased random walk

In addition to exploring the state space, we aim to identify clusters of states around root states. This implies that our exploration must be biased and only the structural aspects of the unbiased random walk remain useful. We learned from the unbiased case that the kernel gives us a rule for the "jump" from one state to another. Therefore we need to define a kernel that also incorporate a selection rule in order to identify the clusters.

Several modifications from the unbiased case are in order. The first being

$$X_{i+1} = \begin{cases} X_i + V_i & \text{if it the value of } X_i + V_i \text{ lies in "low } E\text{-states regions"} \\ X_i & \text{if it the value of } X_i + V_i \text{ lies in "high regions } E\text{-states regions"} \end{cases}.$$

The case $X_{i+1} = X_i$ indicates that we are rejecting the proposal $X_i + V_i$. Equivalently we can write

$$X_{i+1} = X_i + \eta_{i+1} V_i,$$

where

$$\eta_{i+1} = \begin{cases} 1 & \text{if it the value of } X_i + V_i \text{ lies in "low } E\text{-states regions"} \\ 0 & \text{if it the value of } X_i + V_i \text{ lies in "high } E\text{-states regions"} \end{cases}, \quad \text{for all } i \geq 0.$$

This destroys spatial translational invariance. Moreover we can no longer assume that X_i, V_i are independent or the stronger condition that $\{X_0, V_0, V_1, \ldots\}$ are jointly independent. In general, the pdf's X_i and V_i are not known. Nevertheless we still can define a kernel via

$$\mathcal{P}(X_{i+1} \in [a, b] | X_i = \lambda) = \int_a^b \mathrm{d}\sigma \, K(\sigma, \lambda; t_i),$$

with

$$\int_{\mathcal{X}} \mathrm{d}\sigma \, K(\sigma, \lambda; t_i) = 1,$$

for all i and fixed λ. Notice that the kernel K is not longer spatial translational invariant.

Even though we are now considering the rejection of proposals, the conditional probability must be well defined. This implies that the kernel should be decompose into an accepted proposal part and rejected proposal parts. For this reason consider the decomposition

$$K(\sigma, \lambda; t_i) = \mathcal{A}(\sigma, \lambda; t_i) + \mathcal{R}(\sigma, \lambda; t_i),$$

where $\mathcal{A}$ is the acceptance contribution and $\mathcal{R}$ the rejection contribution. They must satisfy

$$\int_{\mathcal{X}} \mathrm{d}\sigma \, \mathcal{A}(\sigma, \lambda; t_i) + \int_{\mathcal{X}} \mathrm{d}\sigma \, \mathcal{R}(\sigma, \lambda; t_i) = 1.$$

For an accepted/rejected proposal we can still have $\mathcal{A}$ and $\mathcal{R}$ well defined. This means that for a rejected proposal $\mathcal{A}$ exist but with a penalization and for an accepted proposal $\mathcal{R}$ exist with penalization. The selection rule incorporates the penalization and therefore in general it is not binary. We will adjoint to $\mathcal{A}$ the selection rule and also the distribution associated to the increments.

We see that

$$\int_a^b \mathrm{d}\sigma \, \mathcal{R}(\sigma, \lambda; t_i),$$

corresponds to the conditional probability that the value of X_{i+1} lies in $[a, b]$ and the proposal is rejected ($X_{i+1} = X_i$) given that $X_i = \lambda$. This does not imply that $\mathcal{A}$ is zero even if we have rejected.

Moreover, the rejection contribution must be of the form

$$\mathcal{R}(\sigma, \lambda; t_i) \propto \delta(\sigma - \lambda),$$

since Dirac's delta gives

$$\int_a^b \mathrm{d}\sigma\, \delta(\sigma - \lambda) = \begin{cases} 1 & \lambda \in [a, b] \\ 0 & \lambda \notin [a, b] \end{cases}.$$

This reproduces the case in which we reject the proposal and consider $X_{i+1} = X_i$. Therefore, after using the normalization of the kernel, we find

$$\mathcal{R}(\sigma, \lambda; t_i) = \delta(\sigma - \lambda) \left[1 - \int_{\mathcal{X}} \mathrm{d}\sigma'\, \mathcal{A}(\sigma', \lambda; t_i) \right].$$

On the other hand we have that

$$\int_a^b \mathrm{d}\sigma\, \mathcal{A}(\sigma, \lambda; t_i),$$

corresponds to the conditional probability that the value of X_{i+1} lies in $[a, b]$ and the proposal is accepted ($X_{i+1} = X_i + V_i$) given that $X_i = \lambda$. This does not imply that $\mathcal{R}$ is zero even if we have accepted.

The acceptance contribution must be of the form

$$\mathcal{A}(\sigma, \lambda; t_i) = w_i(\sigma - \lambda)\alpha(\sigma, \lambda).$$

The function α gives the selection rule. It is defined as "time" independent such that it applies equally for every step. Let us assume that $0 \leq \alpha(\sigma, \lambda) \leq 1$. The $w_i(v)$ corresponds to the pdf that V_i follows. Since one the proposal is accepted we have $V_i = X_{i+1} - X_i$ therefore we have $w_i(\sigma - \lambda)$ and therefore w_i is referred as to the proposal distribution.

Then it is natural that the selection rule must be have a functional form

$$\alpha(\sigma, \lambda) = \Psi[E(\sigma) - E(\lambda)].$$

Then if $E(\sigma) - E(\lambda) \leq 0$, α should be greater contrary to the case $E(\sigma) - E(\lambda) > 0$ we must penalized. The form of the distribution π_β is clearly useful for the penalization and thus we consider

$$\alpha_\beta(\sigma, \lambda) = e^{-\beta \times \max(E(\sigma) - E(\lambda), 0)}.$$

Since $\max(a, b) = \frac{1}{2}(a + b + |a - b|)$ we have $\max(a, 0) = \frac{1}{2}(a + |a|)$ and

$$\alpha_\beta(\sigma, \lambda) = \begin{cases} 1 & E(\sigma) - E(\lambda) \leq 0 \\ e^{-\beta(E(\sigma) - E(\lambda))} & E(\sigma) - E(\lambda) > 0 \end{cases}.$$

Alternatively we can also write $\alpha_\beta(\sigma,\lambda) = \min(1, e^{-\beta\times(E(\sigma)-E(\lambda))})$.

Notice the influence of the parameter β:

$$\lim_{\beta\to\infty} \alpha_\beta(\sigma,\lambda) = \eta_{i+1},$$

for finite E-states differences. This is important, even thought the selection rule is universal in the walk, it recovers a "time" dependence in the $\beta \to \infty$ limit and becomes binary. Then we conclude that $0 \leq \alpha_\beta(\sigma,\lambda) \leq 1$.

Let us dwell into this limit. The kernel with α_β gains an overall dependence on this parameter:

$$K_\beta(\sigma,\lambda,t_i) = w_i(\sigma-\lambda)\alpha_\beta(\sigma,\lambda) + \delta(\sigma-\lambda)\left[1 - \int_{\mathcal{X}} \mathrm{d}\sigma'\, w_i(\sigma'-\lambda)\alpha_\beta(\sigma',\lambda)\right].$$

In the $\beta \to \infty$ limit it reduces to

$$K_\infty(\sigma,\lambda,t_i) = w_i(\sigma-\lambda)\eta_{i+1} + \delta(\sigma-\lambda)\left[1 - \int_{\mathcal{X}} \mathrm{d}\sigma'\, w_i(\sigma'-\lambda)\eta_{i+1}\right].$$

This teach us that we can consider a single w instead of the collection $\{w_i\}$ since the "time" dependence in the acceptance contribution is carried by η_{i+1}.

Therefore, we propose the kernel to be

$$\boxed{K_\beta(\sigma,\lambda,t_i) = w(\sigma-\lambda)\alpha_\beta(\sigma,\lambda) + \delta(\sigma-\lambda)\left[1 - \int_{\mathcal{X}} \mathrm{d}\sigma'\, w(\sigma'-\lambda)\alpha_\beta(\sigma',\lambda)\right],}$$

and find

$$\boxed{\mathcal{P}_\beta(X_{i+1}\in[a,b]|X_i=\lambda) = 1 - \left[\int_{\mathcal{X}} \mathrm{d}\sigma'\, w(\sigma'-\lambda)\alpha_\beta(\sigma',\lambda) - \int_a^b \mathrm{d}\sigma\, w(\sigma-\lambda)\alpha_\beta(\sigma,\lambda)\right].}$$

Now we are in position to consider the expectation value

$$\mathbb{E}_\beta[\varphi(X_{i+1})|X_i=\lambda] = \int_{\mathcal{X}} \mathrm{d}\sigma\, \varphi(\sigma)K_\beta(\sigma,\lambda,t_i),$$

which corresponds to the expected value of the test function φ evaluated at the next state X_{i+1} given that $X_i = \lambda$. Notice that it can be written as

$$\mathbb{E}_\beta[\varphi(X_{i+1})|X_i=\lambda] = \varphi(\lambda) + \int_{\mathcal{X}} \mathrm{d}\sigma\, w(\sigma-\lambda)\alpha_\beta(\sigma,\lambda)[\varphi(\sigma)-\varphi(\lambda)].$$

As an example consider $\varphi(X_{i+1}) = E(X_{i+1})$ to find

$$\mathbb{E}_\beta[E(X_{i+1})|X_i=\lambda] = E(\lambda) + \int_{\mathcal{X}} \mathrm{d}\sigma\, w(\sigma-\lambda)\alpha_\beta(\sigma,\lambda)[E(\sigma)-E(\lambda)].$$

The integral contribution splits by organizing the contribution of states in which $E(\sigma) \leq E(\lambda)$ and $E(\sigma) > E(\lambda)$. We define

$$\mathcal{E}_-(\lambda) = \int_{E(\sigma)\leq E(\lambda)} \mathrm{d}\sigma\, w(\sigma-\lambda)|E(\sigma)-E(\lambda)| \geq 0,$$

and

$$\mathcal{E}_+(\beta,\lambda) = \int_{E(\sigma)> E(\lambda)} \mathrm{d}\sigma\, w(\sigma-\lambda)e^{-\beta(E(\sigma)-E(\lambda))}|E(\sigma)-E(\lambda)| \geq 0$$

Then we have

$$\mathbb{E}_\beta[E(X_{i+1})|X_i=\lambda] = E(\lambda) - \mathcal{E}_-(\lambda) + \mathcal{E}_+(\beta,\lambda).$$

For large β we find that

$$\mathbb{E}_\beta[E(X_{i+1})|X_i=\lambda] \lesssim E(\lambda),$$

and therefore the biased random walk toward low E-state regions. This is the clustering phenomenon.

We can be more precise by considering $\varphi(\sigma) = (\sigma - x_r)^{2k}$, $\lambda = x_r + \delta x_r$ to obtain

$$\mathbb{E}_\beta[(X_{i+1}-x_r)^{2k}|X_i = x_r+\delta x_r] = (\delta x_r)^{2k} + \int_{\mathcal{X}} \mathrm{d}\sigma\, w(\sigma-\lambda)\alpha_\beta(\sigma,\lambda)[(\sigma-x_r)^{2k} - (\delta x_r)^{2k}].$$

In a small neighborhood around the root with multiplicity k, that is $|\delta x_r|$ small, the negative integral contribution arises for $|\sigma - x_r| \leq |\delta x_r|$ ($E(\sigma) \leq E(x_r+\delta x_r)$) and the suppressed positive integral for $|\sigma - x_r| > |\delta x_r|$ ($E(\sigma) > E(x_r+\delta x_r)$). For large β we find that

$$\mathbb{E}_\beta[(X_{i+1}-x_r)^{2k}|X_i = x_r+\delta x_r] \lesssim (\delta x_r)^{2k},$$

and thus local clustering around the root is realized since $E(\sigma) \propto (\sigma - x_r)^{2k}$.

Hence, the proposed biased random walk is controlled by the number of total steps n, β and the parameters of ρ and w. We can only infer that for high values of β we expect that n can be low. That is the walk requieres less "time" to cluster around a single root state. For several root states we need n to be large in order to explore correctly the state space and high values of β serves as a noise reduction parameter. That is, cluster sharpness depends on the values of β. It remains to discuss the role of w and its influence on clustering.

For this reason consider $w(\sigma-\lambda) \propto \delta(\sigma-\lambda)$. We obtain

$$\mathcal{P}_\beta(X_{i+1}\in[a,b]|X_i=\lambda) = \begin{cases} 1 & \lambda \in [a,b] \\ 0 & \lambda \notin [a,b] \end{cases},$$

which indicates that no exploration is implemented. It only preserves whatever state was already chosen initially. To actually incorporate exploration we can consider

$$w(\sigma-\lambda) = \frac{e^{-\frac{(\sigma-\lambda)^2}{2\xi}}}{\int_{\mathcal{X}} \mathrm{d}\sigma'\, e^{-\frac{(\sigma'-\lambda)^2}{2\xi}}}, \quad \xi > 0,$$

where the parameter ξ controls the width and furthermore the exploration.

4 Implementation of algorithms for the biased random walk

4.1 Algorithm 1

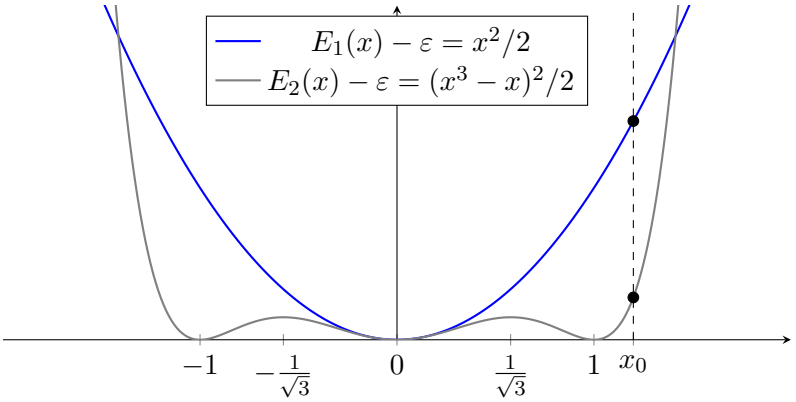
The simplest algorithm that we can apply is of the form

1. Fix n and ξ and ignore β.
2. Sample x_0 from ρ_0 or simply pick by hand x_0.
3. Sample v_0 from $w(\sigma - x_0)$ (truncated Gaussian sampling). Set $x_1 = x_0 + v_0$ if $E(x_1) - E(x_0) \leq 0$ and $x_1 = x_0$ if $E(x_1) - E(x_0) > 0$[3].
4. Sample v_1 from $w(\sigma - x_1)$. Set $x_2 = x_1 + v_1$ if $E(x_2) - E(x_1) \leq 0$ and $x_2 = x_1$ if $E(x_2) - E(x_1) > 0$.
5. Carry on until we reach a set of n states which some of them are the same due to rejection.
6. Repeat for different x_0.
7. Plot the histogram of all the walks. The histogram should be multimodal, with peaks located at the root states.

Let us discuss this algorithm for two cases

$$f_1(x) = x, \quad f_2(x) = x(x-1)(x+1), \quad \mathcal{X} \in [-2, 2].$$

The E-states are depicted for both cases in the following figure:



Suppose that we start at x_0 as shown at the figure. For E_1 all the stochastic jumps will move towards the root state and can change side with respect the y-axis. For E_2 we face a different situation, still stochastic jumps will move towards a root state but it might confined in just one of them. That is, near a root state, the algorithm will favor clustering and jumps to another root state will be not supported (there is a local maximum between

[3]The key point is that for small values of ξ we could find $x_1 \approx x_0$ independently if we accept or reject the proposal and thus the exploration is intentionally reduced.

the root states) unless the distribution w allows it. That is why, in the current algorithm, we need to run several biased random walks.

In figure 1 we concentrate on $f_2(x)$ and the effects of ξ. We see that for larger values of

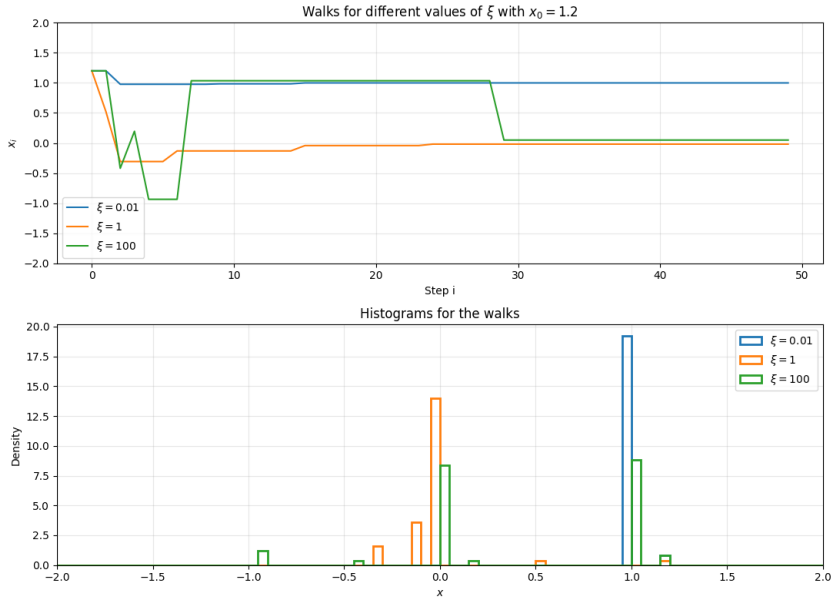


Figure 1. Biased random walks ($\beta \to \infty$) with $n = 50$, $x_0 = 1.2$ and $\xi = 0.01, 1, 100$. Script: A1.py

ξ the jump's are bigger since we are broadening the probability density. Moreover, we see the effect of been near a root for ξ small: we are not able to get out of the root state well.

In figure 2 we increase n and observe the clustering around the root states.

For a single ξ (say $\xi = 1$), it is reasonable to consider several walks with an initial condition x_0 sampled form a uniform distribution and large n. Then, we just plot the overall walks histogram as depicted in figure 3 for 10 walks and figure 4 for 1000 walks. Notice that the height of the $x = \pm 1$ root states is suppressed.

This basic algorithm teaches us the importance of the interplay between n, ξ and the number of walks. Regarding the omission of β and α_β, we can interpret this case as the $\beta \to \infty$ limit. That is why a walk can descend into a well and cluster. Nevertheless, if ξ is sufficiently large enough we can actually "tunnel" the well in spite the low E-states rule.

4.2 Algorithm 2

Now we want to incorporate α_β. The kernel is of the form

$$K_\beta(\sigma, \lambda, t_i) = w(\sigma - \lambda)\alpha_\beta(\sigma, \lambda) + \delta(\sigma - \lambda)\left[1 - \int_{\mathcal{X}} \mathrm{d}\sigma'\, w(\sigma' - \lambda)\alpha_\beta(\sigma', \lambda)\right],$$

where w is a pdf and α_β is not. Since $0 \leq \alpha_\beta \leq 1$ we can interpret α_β also as a conditional probability of acceptance. That is, the probability of accepting given the proposal and current state.

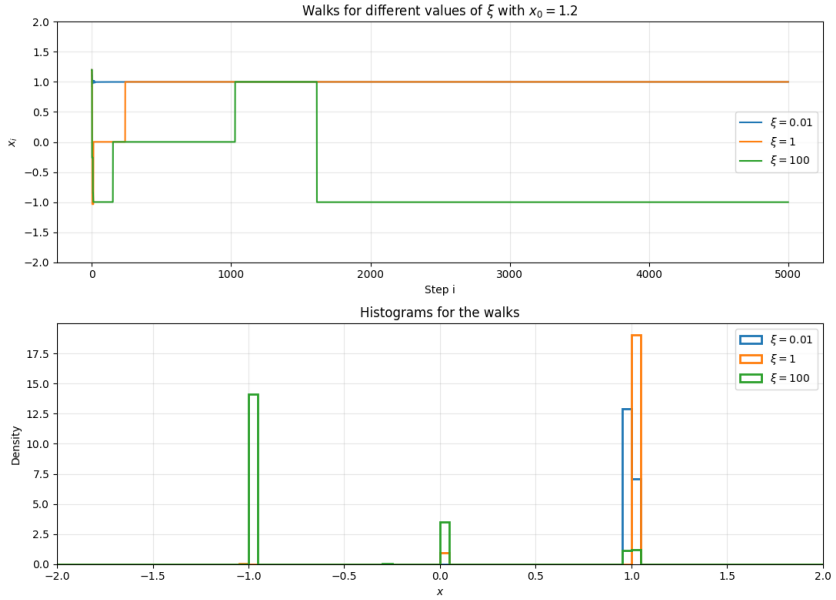


Figure 2. Biased random walks ($\beta \to \infty$) with $n = 5000$, $x_0 = 1.2$ and $\xi = 0.01, 1, 100$. Script: A1.py

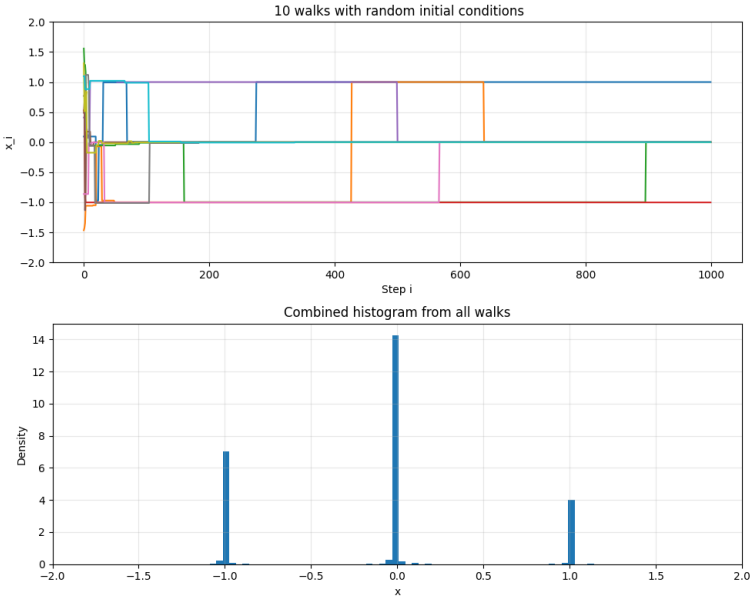


Figure 3. Biased random walks ($\beta \to \infty$) with $n = 1000$, $\xi = 1$ and 10 initial conditions. Script: A2.py

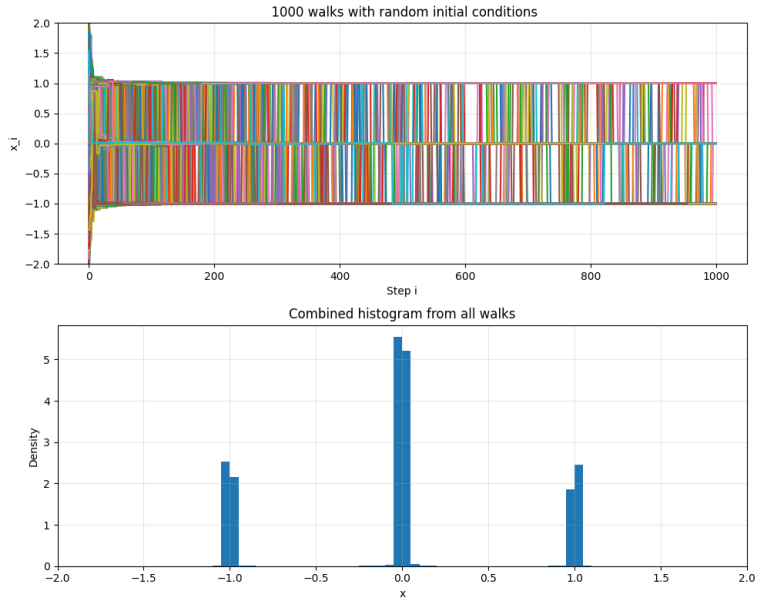


Figure 4. Biased random walks ($\beta \to \infty$) with $n = 1000$, $\xi = 1$ and 1000 initial conditions. Script: A2.py

Suppose that x_0 is given and we sample v_0 from $w(\sigma - x_0)$. Then we would obtain

$$x_1 = \begin{cases} x_0 + v_0 & \text{with probability } \alpha_\beta(x_0 + v_0, x_0) \\ x_0 & \text{with probability } 1 - \alpha_\beta(x_0 + v_0, x_0) \end{cases}.$$

Recall that $\alpha_\beta = 1$ if $\Delta E \leq 0$ and $\alpha_\beta = e^{-\beta \Delta E}$ if $\Delta E > 0$ where $\Delta E = E(x_0 + v_0) - E(x_0)$. For the former case we obtain

$$x_1 = \begin{cases} x_0 + v_0 & \text{with probability } 1 \\ x_0 & \text{with probability } 0 \end{cases}, \quad \Delta E \leq 0.$$

For the latter

$$x_1 = \begin{cases} x_0 + v_0 & \text{with probability } e^{-\beta \Delta E} \\ x_0 & \text{with probability } 1 - e^{-\beta \Delta E} \end{cases}, \quad \Delta E > 0.$$

The first point to notice is that for $\Delta E < 0$, moving towards low E-states regions, the move is determinist. In contrast, moving toward high E-states regions, the move is probabilistic.

So the rule can be summarized as:

$$\Delta E \leq 0 \ \to \ \text{move always accepted}, \quad \Delta E > 0 \ \to \ \text{move sometimes rejected}.$$

The second point is that for $\Delta E > 0$ requires the realization of a binary random experiment. To this end, we define $U_1 \sim \text{Uniform}(0, 1)$ with

$$\int_0^p \mathrm{d}u \ + \int_p^1 \mathrm{d}u \ = 1, \quad p = e^{-\beta \Delta E}.$$

Then we have $\mathtt{P}(U_1 \leq p) = p$. We generate a sample u_1 and consider

$$x_1 = \begin{cases} x_0 + v_0 & u_1 \leq e^{-\beta \Delta E} \\ x_0 & u_1 > e^{-\beta \Delta E} \end{cases}, \quad \Delta E > 0, \quad U_1 \sim \text{Uniform}(0,1).$$

So now the algorithm can be defined incorporating α_β. From figures 5, 6 7, 8 and 9, we see that the final histogram mimics a probability distribution density and β controls the "noise" (compare figure 5 and 6). Thus, due to fact that now sometimes we reject proposals the random walks are different and create more "noise". Consequently we do not need to consider a large number of walks, see figure 8 and 9.

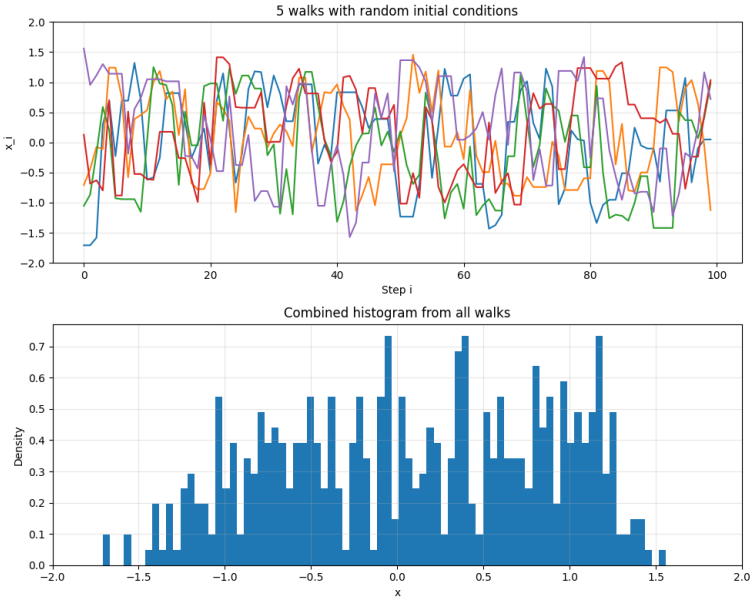


Figure 5. Biased random walks with $\beta = 1$, $n = 100$, $\xi = 1$, 5 initial conditions. Script: A3.py

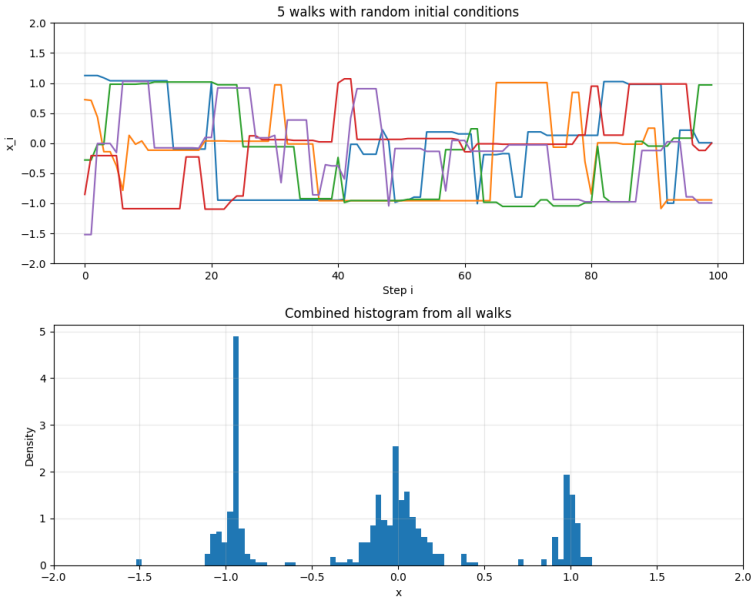


Figure 6. Biased random walks with $\beta = 100$, $n = 100$, $\xi = 1$, 5 initial conditions. Script: A3.py

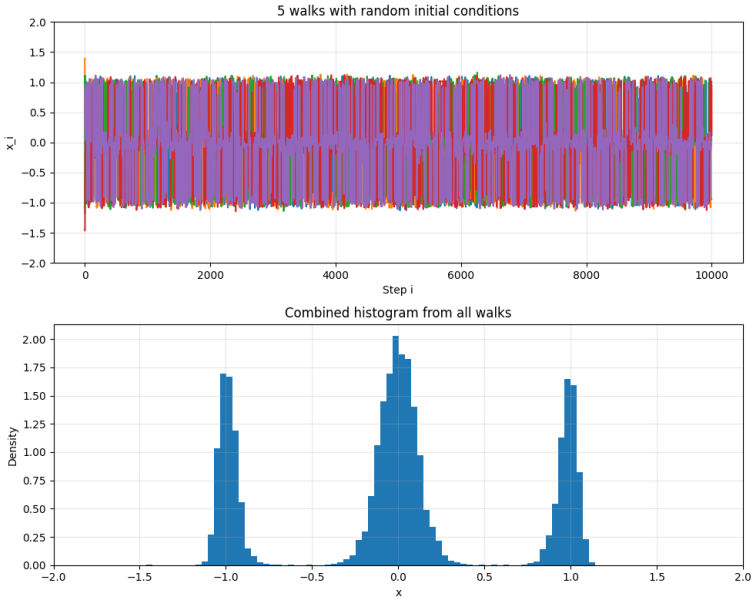


Figure 7. Biased random walks with $\beta = 100$, $n = 10000$, $\xi = 1$, 5 initial conditions. Script: A3.py

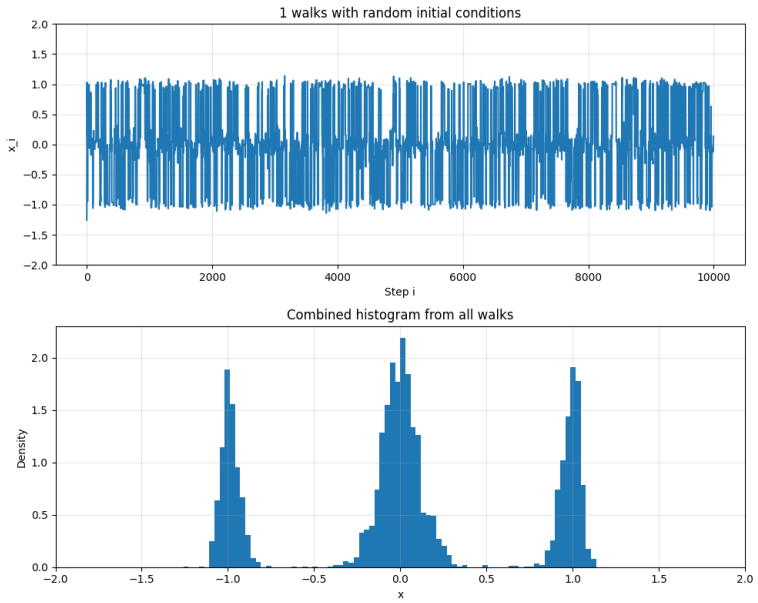


Figure 8. Biased random walks with $\beta = 100$, $n = 10000$, $\xi = 1$, 1 initial condition. Script: A3.py

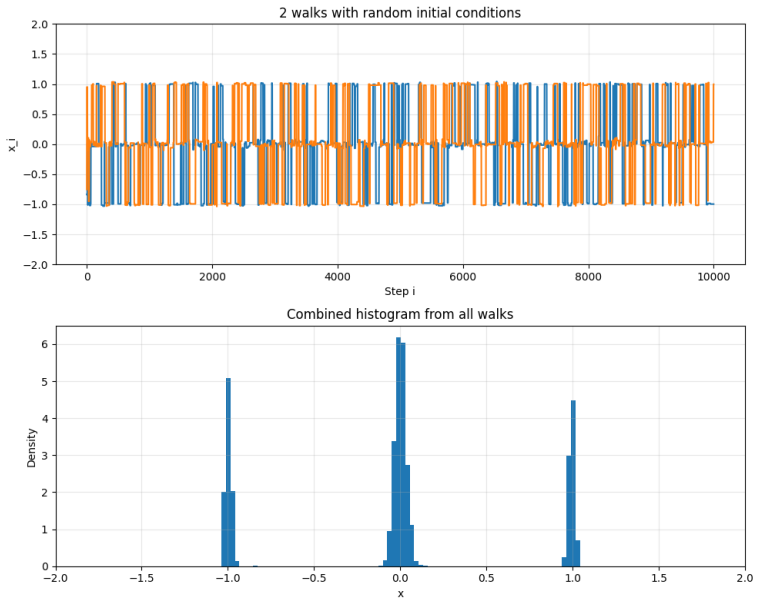


Figure 9. Biased random walks with $\beta = 1000$, $n = 10000$, $\xi = 1$, 2 initial conditions. Script: A3.py

4.3 Algorithm 3

Using algorithm 2, we now incorporate a section in which we define the root state and a region around it. The reason behind this is two fold: we can use the algorithm to find the root states or answer the question: given x_* does it corresponds to the root of $f(x)$? Clearly the question translates into: does x_* belong to one of the possible regions?

We expect for large β that the regions are well define (sharp and do not overlap) and we will use known of algorithms that find the peaks, the sample mean and standard deviation for each of them[4]. So we will define the region as $\mu \pm \sigma$ per each root state. Then, the answer to the question reduces to compare x_* to the regions. This is illustrated in figure 10.

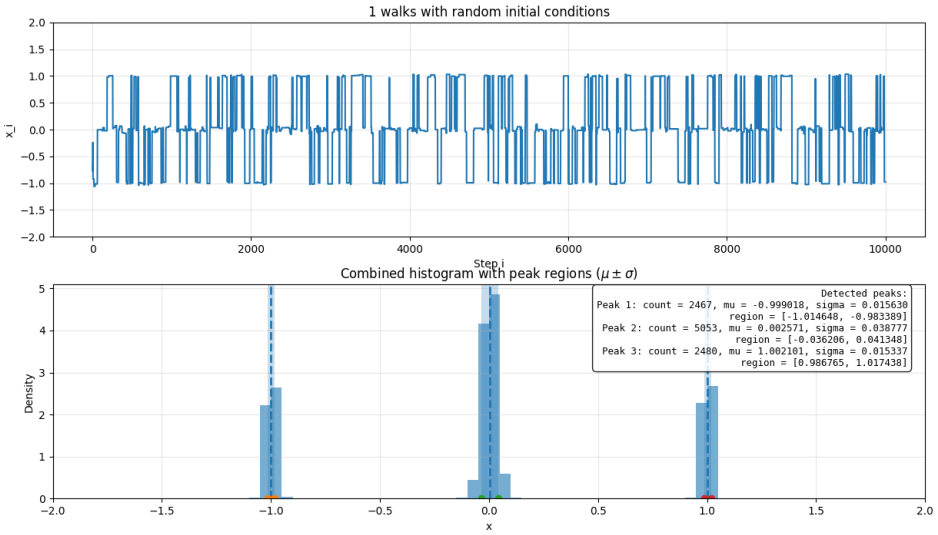


Figure 10. Biased random walks with $\beta = 1000$, $n = 10000$, $\xi = 1$, 1 initial conditions. Script A4.py

Finally we discuss the effect of ξ in figure 11. Recall that for $\beta \to \infty$, ξ large we are allowing the possibility of "tunnel" away from a well. For β large but finite, the effect of α_β comes into play. We notice that also provides a mechanism to "tunnel" away from a well in spite for a small value of ξ .

[4]By itself, this is a discrete optimization problem. Recall the second derivative criteria in calculus.

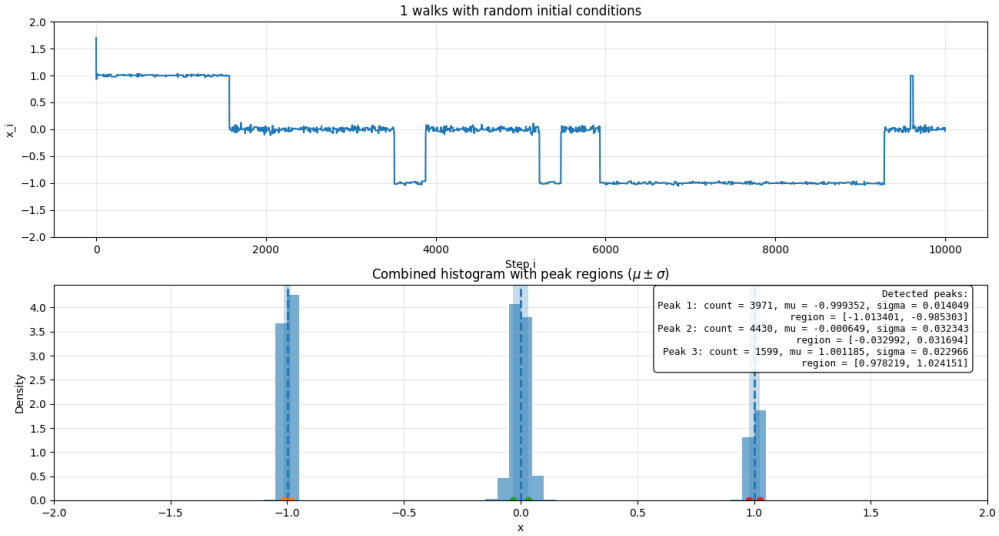


Figure 11. Biased random walks with $\beta = 1000$, $n = 10000$, $\xi = 0.1$, 1 initial conditions. Script A4.py