

Wang-Landau Algorithm: an adapted random walk to boost convergence

Augustin Chevallier, Frédéric Cazals

▶ **To cite this version:**

Augustin Chevallier, Frédéric Cazals. Wang-Landau Algorithm: an adapted random walk to boost convergence. [Research Report] RR-9223, INRIA Sophia Antipolis, France. 2018, 34p. hal-01919860v2

HAL Id: hal-01919860
https://hal.science/hal-01919860v2

Submitted on 14 Dec 2018 (v2), last revised 19 Nov 2019 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Wang-Landau Algorithm: an adapted random walk to boost convergence

Augustin Chevallier and Frédéric Cazals

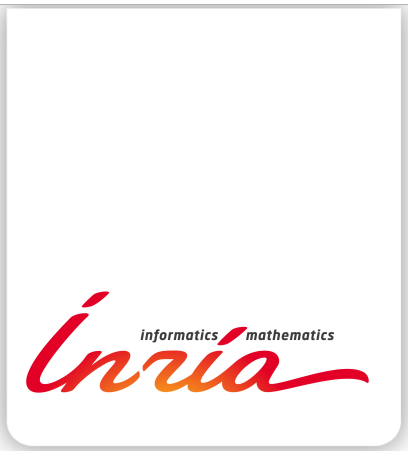
RESEARCH REPORT

N° 9223

November 2018

Project-Team Algorithms-Biology-Structure

ISSN 0249-6399 ISRN INRIA/RR--9223--FR+ENG



Wang-Landau Algorithm: an adapted random walk to boost convergence

Augustin Chevallier and Frédéric Cazals

Project-Team Algorithms-Biology-Structure

Research Report n° 9223 — November 2018 — 34 pages

Abstract: The Wang-Landau (WL) algorithm is a recently developed stochastic algorithm computing densities of states of a physical system. Since its inception, it has been used on a variety of (bio-)physical systems, and in selected cases, its convergence has been proved. The convergence speed of the algorithm is tightly tied to the connectivity properties of the underlying random walk.
As such, we propose an efficient random walk that uses geometrical information to circumvent the following inherent difficulties: avoiding overstepping strata, toning down concentration phenomena in high-dimensional spaces, and accommodating multidimensional distribution.
Experiments on various models stress the importance of these improvements to make WL effective in challenging cases. Altogether, these improvements make it possible to compute density of states for regions of the phase space of small biomolecules.

Key-words: MCMC, Wang-Landau, statistical physics, random walk, high dimension, sampling, importance sampling

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Wang-Landau Algorithm: an adapted random walk to boost convergence

Résumé : L'algorithme de Wang-Landau est un algorithme stochastique récemment développé calculant la densité d'états pour des systèmes physiques. Depuis sa création, il a été utilisé sur des systèmes (bio-)physiques. Dans certain cas, sa convergence a été prouvée. La vitesse de convergence de l'algorithme est intimement liée aux propriétés de connectivité de la marche aléatoire sous-jacente.

Nous proposons ici une marche aléatoire efficace utilisant des informations géométriques pour prévenir les difficultés suivantes: passer par dessus des strates, atténuer les phénomènes de concentration de la mesure en grande dimension, et gérer les distributions multimodales.

Les expériences numériques sur différents modèles démontrent l'importance de ces améliorations pour rendre WL efficace dans des cas complexes. In fine, ces améliorations rendent possible le calcul de densité d'état pour des régions de l'espace des phases de petite bio-molécules.

Mots-clés : MCMC, Wang-Landau, physique statistique, marche aléatoire, grande dimension, échantillonnage, importance sampling

Contents

1 Introduction **5**

2 The Wang-Landau algorithm **6**
2.1 Problem statement . . . 6
2.2 Algorithm . . . 7
2.3 Theoretical convergence . . . 8
2.4 Convergence rate: further insights . . . 8
2.5 MCMC and adaptivity . . . 9

3 Improving convergence speed **9**
3.1 Rationale . . . 9
3.2 Overstepping strata . . . 10
3.2.1 Problem . . . 10
3.2.2 Solution . . . 10
3.3 High dimensionality and concentration . . . 11
3.3.1 Problem . . . 11
3.3.2 Solution . . . 11
3.4 Handling multimodal distributions via darting . . . 12
3.4.1 Problem . . . 12
3.4.2 Solution . . . 12
3.5 Splitting energy bins . . . 13
3.6 The random walk . . . 13

4 Experiments **14**
4.1 Setup . . . 14
4.1.1 Statistics of interest . . . 14
4.1.2 Contenders . . . 14
4.1.3 Models . . . 15
4.2 Results . . . 15
4.2.1 Single well potential . . . 15
4.2.2 Single well potential - no isotropic . . . 15
4.2.3 Dual wells potential: darting . . . 15
4.2.4 Dialanine . . . 16

5 Outlook **16**

6 Artwork **19**
6.1 Method . . . 19
6.2 Results . . . 20

7 Appendix: uniform sampling in a hypercone **27**
7.1 Pre-requisites . . . 27
7.2 Algorithm to uniformly sample a hypercone . . . 28
7.2.1 Overview . . . 28
7.2.2 Sampling from f_{cap} . . . 29
7.2.3 Sampling from f_{cone} . . . 30
7.3 Changing the cone axis . . . 30

8 Appendix: transition probability for darting **32**
8.1 Notations 32
8.2 Assumptions 32
8.3 Derivation of the transition probability 33

1 Introduction

The Wang-Landau algorithm for density of states calculations. The derivation of observable properties of (bio-)molecular systems at thermodynamic equilibrium relies on statistical physics, with the formalism of stochastic ensembles playing a pivotal role[1, 2, 3, 4]. Amidst the various algorithms available, the Wang-Landau (WL) algorithm [5, 6] is now well known and widely used despite its recent inception, in particular due to its simplicity and genericity. The WL algorithm estimates the density of states (DOS) of a system, which is especially useful to compute partition functions in statistical physics, and more generally observables–e.g. the average energy or the heat capacity. Estimating the DOS is especially challenging for standard Markov Chain Monte Carlo techniques.

To review previous work, it is important to recall that the WL algorithm falls in the realm of adaptive MCMC sampling algorithms. In a nutshell, WL returns an estimation of the DOS in terms of histogram. The bins of the histogram correspond to a partitioning of the energy range of the system. The algorithm resorts to importance sampling, using a biasing function derived from the current estimation of the DOS. Since the limit distribution sought is defined by the density of states, the random walk is build from the Metropolis-Hastings algorithm (M-H), using the current DOS estimate in the rejection rate. (We note in passing that since the DOS values used to define transition probability depend on the history, WL is not a Markov process.) Additionally, a so-called flat histogram rule may be used to count the visits in each energy stratum and update the learning rate when all stratum have been evenly visited. These main ingredients recalled, one may observe that numerous improvements were made to the original algorithm [5], both in terms of design and analysis of performances. The first key improvement has been the 1/t algorithm which solved the so-called saturation error problem [7, 8], in which a constant error on DOS estimates was incurred, due to a too fast reduction of the learning rate. Another key initiative has been to tune the random walk and the energy discretization [9], as large bins may hinder convergence by keeping the system trapped. To avoid this pitfall, a dynamic maintenance of bins has been proposed, in order to maintain a proper balance of samples across a stratum. Concomitantly, a random walk defined from a mixture of Gaussians has been introduced, in order to attempt moves of the proper size. In a different vein, it has been proposed to speed up convergence resorting to parallelism via multiple walkers [6]. However, this approach should be taken with care, as problems arise when a large number of walker are used [10].

On the mathematical side, for the WL algorithm variant using the flat histogram, the importance of the analytical form of the DOS update rule was established [11]. For WL with a deterministic adaptation of the learning rate, to which the 1/t variant belongs, the correctness of the DOS estimates was proved, regardless of the particular analytical expression of the update rule [12].

Applications. Application-wise, WL has been used on a variety of physical systems, and more recently to biomolecules. Thermodynamics properties of RNA secondary structures were estimated using the WL algorithm [13]. Properties of clusters and peptides (up to 8 a.a.) were studied in [14]. Likewise, the thermodynamics properties of misfolded (containing a helix structure rather than a β-sheet) proteins, such as those involved in mad cow and Creutzfeldt-Jakob diseases, were studied by feeding a coarse grain protein to the 1/t WL algorithm variant [15]. In a similar spirit, a modified flat rule histogram was used in [16] to study properties of polymers on a lattice, in the HP model. However, processing continuous models of protein of significant size has remained out of reach so far [17].

Contributions. The random walk and the energy discretization influence one another: the average step size of the random walk should be dependent on the size energy bins. For large energy bins, the step size should be large, and small for narrow energy bins. Thus the random walk and bin sizes should not be independent, and the step size of the random walk should depend on local information. Such intricacies have precluded the development of effective WL algorithms to to handle systems as complex as bio-molecules, and the goal of this paper is precisely to improve the convergence speed of the algorithm, especially in high dimensional settings. (NB: our focus is not on asymptotic convergence properties.)

To make a stride towards circumventing these observations, we make three contributions targeting improvements of the convergence speed (section 3.1). First, we design a random walk which takes the bin size and local geometric information into account to avoid overstepping strata (section 3.2). Second, we tackle the so-called measure concentration problem inherent to high dimensional spaces, which is especially pregnant near local minima, and which decreases convergence speed exponentially fast with the dimension (section 3.3). Finally, we introduce a darting move for multimodal distributions (section 3.4). In addition, we provide a generic WL implementation, which allows tuning all key building blocks. The source code is integrated to the SBL Structural Biology Library [18] and `http://sbl.inria.fr`.

2 The Wang-Landau algorithm

2.1 Problem statement

Consider a probability distribution with density $\pi(x)$ defined on a subset $\mathcal{E} \subset \mathbb{R}^D$. Also consider a partition of $\mathcal{E}$ into so-called strata $\{\mathcal{E}_1, \ldots, \mathcal{E}_d\}$. Denoting λ the Lebesgue measure, our problem is to estimate

$$\theta_i^* \stackrel{Def}{=} \int_{\mathcal{E}_i} \pi(x)\lambda(dx). \tag{1}$$

This problem arises in many areas of science and engineering, two of them being of particular interest in the sequel.

Statistical physics. Assume that one is given a potential energy function $U : \mathbb{R}^n \longrightarrow \mathbb{R}$. The measure of interest π is given by Boltzmann's distribution

$$\pi(x) = Z_\beta^{-1} exp(-\beta U(x)), \text{ with } Z_\beta = \int_{\mathcal{E}} exp(-\beta U(x))dx. \tag{2}$$

Note that Z_β is the so-called partition function of the system. Consider the following discretization $U_0 < U_1 < ... < U_d$ of the potential energy space. The strata $\mathcal{E}_i$ are the pre-images of the potential energy, that is

$$\mathcal{E}_i = \{x \in \mathcal{E} | U(x) \in [U_{i-1}, U_i)\}$$

In this context, Eq. (1) reads as $\theta_i^* = Z_\beta^{-1} \int_{\mathcal{E}_i} exp(-\beta U(x))dx$, and one has $\sum_i \theta_i^* = 1$. The WL algorithm computes estimates θ_i for θ_i^*, which also satisfy $\sum_i \theta_i = 1$. The individual quantities θ_i are of interest since their values provide the relative weights of the strata. However, they do not give access to the partition function Z_β itself, whose calculation requires a re-normalization.

It should also be noticed that incorporating Boltzmann's factor π into Eq. (1) results in quantities θ_i which depend on the particular temperature used. If one uses $\pi(x) = 1/\lambda(\mathcal{E})$ instead, the quantities θ_i are volumes in phase space, and can be used to estimate the partition function at any temperature, using a calculation akin to numerical integration.

Numerical integration. A closely related problem is the calculation of a D-dimensional integral

$$I_D = \int_{\mathcal{E}} f(x)dx. \tag{3}$$

Assume that the range $Y_f = [y_{\min}, y_{\max}]$ spanned by f in the domain I is known, and that this interval has been split into n interval $[y_i, y_i + dy]$. Consider the the estimates from Eq. (1) with $\pi(x) = 1/\lambda(\mathcal{E})$. Assume that WL has been run, and denote $\overline{y_i}$ the average value of function f computed over all points such that $f(y) \in [y_i, y_i + dy]$. The integral can be estimated as [19]

$$I \approx \sum_{i=1,\ldots,n} \theta_i^{Norm}\ \overline{y_i},\ \text{with } \theta_i^{Norm} = \lambda(\mathcal{E})\theta_i. \tag{4}$$

2.2 Algorithm

Sampling from a probability distribution μ. We assume the existence of a random walk q on the $\mathcal{E}$ with transition probability density $q(x, y)$ – one transition from x to y. Using the Metropolis-Hastings transition kernel, and denoting δ_x a point mass at x, we introduce a random walk P_μ whose limiting distribution is μ [20]:

$$P_\mu(x, dy) = q(x, y)\alpha(x, y)dy + \delta_x(dy) \int_{\mathcal{E}} (1 - \alpha(x, z))q(x, dz), \tag{5}$$

where the acceptance probability of the new state y is given by

$$\alpha(x, y) = 1 \wedge \frac{\mu(y)q(y, x)}{\mu(x)q(x, y)} \tag{6}$$

Note that here, we allow q to be non-symmetric by introducing the correction factor $\frac{q(y,x)}{q(x,y)}$, which will be used extensively later.

Algorithm. Consider the strata $\mathcal{E}_i$ defined above, as well at the mapping $J : A \to \{1, \ldots, d\}$ returning the index $J(x)$ of the energy level of x.

The WL algorithm iteratively construct a sequence $\theta(t)$ of estimates of $\theta^* = (\theta_1^*, ..., \theta_d^*)$. For an estimate θ, we introduce the probability density

$$\pi_\theta(x) = \left(\sum_{i=1}^{d} \frac{\theta_i^*}{\theta_i} \right)^{-1} \frac{\pi(x)}{\theta_{J(x)}} \tag{7}$$

The weight of each $\mathcal{E}_i$ under π_θ is proportional to θ_i^*/θ_i; In particular, all energy levels have the same weight $1/d$ under π_{θ^*}. The algorithm is then an importance sampling-like strategy, using π_θ as the bias.

Observe that points sampled according to π_θ fall on average more in bins with underestimated density. The rationale of the algorithm is to sample at each step a point x according π_θ, multiply $\theta_{J(x)}$ by an increment $\gamma > 1$ called the **learning rate**, and finally decrease the learning rate by a small fraction. A Markov kernel P_θ with invariant density π_θ is build using Eq.5 model and is used to sampled points from π_θ.

How to decrease the learning rate γ requires a small discussion. Historically [5], the rule used the Flat Histogram criterion. Let $\nu_t(i)$ be the number of samples up to iteration t falling into bin

$\mathcal{E}_i$. The vector $\{\nu_t(i)\}$ is said to verify the **flat histogram** (FH) criterion provided that, given a constant c:

$$\max_{i=1,...,d} \mid \frac{\nu_t(i)}{t} - 1/d \mid < c. \tag{8}$$

If the criterion was verified, γ is decreased using $\gamma = \sqrt{\gamma}$. Unfortunately, this too fast rate yields an error known as the saturation error [7, 8]. Also, the flat histogram variant is sensitive to the particular analytical form of the update rule [11]. To circumvent this difficulty, the rate $\gamma_t = exp(1/t)$ rule was proposed [7]. Practically, one combines the two update strategies by starting with the flat histogram strategy and switching as soon as the proposed γ is smaller than $exp(1/t)$ – [7] and Fig. 1.

The complete algorithm (Algo. 1) depends on the following parameters which influence its convergence speed: (i) the constant c for the flat histogram, (ii) the value of γ_0, (iii) the energy discretisation, (iv) the random walk q.

Algorithm 1 Wang Landau

Set $\theta = (1/d, ..., 1/d)$
Set exponential regime = True
Set $\gamma = \gamma_0$ with $\gamma_0 > 1$
while $t < t_{max}$ **do**
  Sample $x_{t+1} \sim P_\theta(x_t, .)$
  Set $\theta_{J(x_{t+1})} = \gamma\ \theta_{J(x_{t+1})}$
  Renormalise θ
  if Exponential regime **then**
    if Flat histogram **then**
      $\gamma = \sqrt{\gamma}$
    if $\gamma < \exp(\frac{1}{t+1})$ **then**
      Set exponential regime = False
      Set $\gamma = \exp(\frac{1}{t+1})$
  else
    $\gamma = \exp(\frac{1}{t+1})$

2.3 Theoretical convergence

The theoretical convergence has been studied [12], using suitable assumptions on (i) the equilibrium measure, (ii) the Metropolis-Hastings kernel, and (iii) the sequence of learning rates. Under these assumptions, the Wang-Landau algorithm has been proven to converge. The authors proved a central-limit like theorem which give a theoretical convergence speed of $O(1/\sqrt{n})$ where n is the number of step.

This theoretical convergence speed is the *same* than the one of classical Monte Carlo integration. Practically though, Monte-Carlo integration often fails while Wang-Landau does not, stressing the role of constants in the convergence speed.

2.4 Convergence rate: further insights

The convergence speed of the algorithm is tightly coupled to the mixing times of the Markov chains P_θ, which is roughly the time it take for $P^t_\theta(x, .)$ to converge to π_θ for any x.

In many cases, the bottleneck for the mixing time is the visit of all the energy levels. In [9], a refinement rule for the discretisation is provided as well as a rule to find suitable parameters for a multi-modal Gaussian random walk. The paper do not provide any explicit insights on the link between the random walk and the discretisation. However, they use a symmetric random walk. Such random walk will sample the space uniformly. Hence, to obtain a high transition probability between two energy levels, the ratio of their respective volumes must be controlled: should the ratio be too small (or to high), the probability of proposing a move going from the smallest energy level to the biggest is so small that is never occurs. Observe that this restriction vanishes in using a non symmetric random walk, a strategy we will be using.

For multi-modal distributions, the difficulty to switch from one mode to another can also be a bottleneck for the mixing time. In [21], a strategy called darting is proposed. It consists in attempting long range jumps between regions associated to precomputed modes. The knowledge of the volume of the targeted regions allows one to guarantee detailed balance [22] whence a procedure sampling the desired distribution. Note that for molecular systems, where Boltzmann distribution yields one mode for each local minimum of the potential energy, local minima can be obtained by gradient descents and associated search methods such as basin hopping and variants [23, 24].

2.5 MCMC and adaptivity

For general MCMC algorithm, it has been shown that an adaptive random walk can lead to erroneous results [25]. Practically, for a given probability π, there might exist a sequence P_i of Markov kernels with limiting distribution π for all i such that for a given X_0, the sequence of random variables defined by $X_i \sim P_i(X_{i-1}, .)$ does not converge to the limiting distribution π. This does not affect the Wang-Landau algorithm itself. However, any adaptivity must be stopped before the end of the algorithm. The choice we make is to stop any adaptivity once the flat histogram has been met a given number of times denoted N_{FHE} in the sequel.

3 Improving convergence speed

3.1 Rationale

The performances of WL result from a subtle interplay between various ingredients, notably the energy discretization, the topography of the landscape, and the random walk. The improvements presented thereafter target the following difficulties:

- Difficulty 1 – section 3.2: topography adapted random walk to avoid overstepping strata. The random walk should exploit the geometry of the landscape, to foster the diffusivity between strata.

- Difficulty 2 – section 3.3: curse of dimensionality and concentration phenomena. In high dimensions, when the probability mass is concentrated in a *small* typical set, move sets exploring uniformly the entire space face difficulties to sample such sets. We introduce a biasing strategy (in terms of directions for the move set), promoting diffusivity between strata.

- Difficulty 3 – section 3.4: multimodal distributions. To deal with the case of multimodal distributions, we resort to darting, a strategy meant to connect parts of the energy landscape which are separated by regions of low probability.

- Difficulty 4 – section 3.5: energy range discretization. Slow mixing of the random walk may be due to an inappropriate energy discretization. We resort to a refinement strategy to fix such problems.

All in all, we aim for a *ladder-like* random walk as described in Fig. 2 which connects each energy level with the one bellow and the one on top with an as high as possible probability.

3.2 Overstepping strata

3.2.1 Problem

Strata of *small* thickness tend to be stepped over. This typically happens when the landscape is steep‘ or the discretization is fine. It is thus important to adapt the travel distance of the random walk in such regions.

A Gaussian mixture identical for all strata has been used [9]. However, the mixture is symmetric (see section 2.4) and does not exploit the geometry of the landscape.

3.2.2 Solution

We estimate the local "steepness " of the energy function using an order 2 Taylor expansion of the energy. Normally, one would think that the "steepness" of the energy function (and thus the diameter of the energy levels) is encoded by gradient. However, around locals minimums this fails spectacularly as the gradient becomes close to 0 making the energy function appear flat.
First we pick a direction $\vec{u}$ in the unit sphere $\mathbb{S}^{n-1}$. Then we compute the Taylor expansion in the direction u with $h \in \mathbb{R}$:

$$U(x + h\vec{u}) = U(x) + h(\nabla U \cdot \vec{u}) + 1/2h^2(\vec{u}^T \mathrm{Hess}\, \vec{u}). \tag{9}$$

Remark 1. *A numerical approximation can be efficiently computed using the gradient only, avoiding the costly computation of the Hessian.*

Assuming that x is in $\mathcal{E}_i$, we compute using the Taylor expansion the interval $[h_0, h_1]$ such that for $h \in [h_0, h_1]$ (Fig. 3)

$$x + h\vec{u} \in \mathcal{E}_i \tag{10}$$

Doing the same for $\mathcal{E}_{i-1}$ and $\mathcal{E}_{i+1}$ yields $[h_{-1}, h_0]$ and $[h_1, h_2]$. The last steps are to pick any of these 3 intervals with probability $1/3$ and to sample h uniformly in the chosen interval.

Doing so effectively adapt the random walk to the local steepness of the energy landscape, allowing multiple scales. Even better, it also changes and adapt to the chosen direction $\vec{u}$.

For any x and y, with $u = \frac{y-x}{\|y-x\|}$, we can compute the probability of going from x to y with this random walk:

$$q_{flat}(x, y) = P_{dir}(u) \frac{1}{\|y - x\|^{n-1}} \sum_{i=0}^{2} 1_{[h_{i-1}, h_i]} < y - x, u > \frac{1}{|h_i - h_{i-1}|} \tag{11}$$

$$= \frac{\Gamma(n/2)}{2\pi^{n/2}\|y - x\|^{n-1}} \sum_{i=0}^{2} 1_{[h_{i-1}, h_i]} < y - x, u > \frac{1}{|h_i - h_{i-1}|} \tag{12}$$

Note that the term before the sum is symmetric in x and y, hence it simplifies when computing $\frac{q_{flat}(y,x)}{q_{flat}(x,y)}$.

Remark 2. *In the ideal case where the level sets are planes (meaning that the gradient dominates), the Metropolis-Hastings correction factor* $\frac{q_{flat}(y,x)}{q_{flat}(x,y)}$ *introduced by the non symmetry of the random walk when sampling a point in* $\mathcal{E}_i$ *starting from* x_0 *in* $\mathcal{E}_1$ *is* $V(\mathcal{E}_i)/V(\mathcal{E}_1)$*, which cancels out the metropolis acceptance ratio in Wang-Landau when* θ *is close to* θ^**. It effectively bringing the asymptotic rejection rate of the Wang-Landau random walk to 0.*

3.3 High dimensionality and concentration

3.3.1 Problem

Consider the problem of lowering the energy by moving from stratum $\mathcal{E}_i$ to $\mathcal{E}_{i-1}$ (Fig. 4). To do so, a direction in a cone of angle α must be chosen–any direction outside of this cone never intersects $\mathcal{E}_{i-1}$. As the dimension increases, the probability of sampling a point in a cone of aperture α decreases exponentially with the dimension (Fig. 5), preventing the random walk to reach the stratum $\mathcal{E}_{i-1}$.

3.3.2 Solution

The most straightforward way to overcome this problem is to decrease the bin size. Indeed doing so makes α closer to $\pi/2$, increasing the probability of picking a direction allowing going from $\mathcal{E}_i$ to $\mathcal{E}_{i-1}$. In practice, the bin splitting strategy from [9] achieves this goal. However, the number of strata increases with dimension, making this strategy less effective as the dimension increases.

Another way to solve this problem is to bias the random walk toward the cone allowing reaching $\mathcal{E}_{i-1}$.

Let $C_{down}(x) \subset S^{n-1}$ the subset of direction which allow moving downward (or upward) from a point x. An approximation of $C_{down}(x)$ could be found using a full second order Taylor expansion (thus requiring the full Hessian matrix). However sampling points uniformly in this subset is hard, hence we fall back on a simpler approximation stipulating that a single cone can be used for each stratum. This hypothesis relies on the following two assumptions:

- the strata is not too wide,
- the curvature of the strata does not vary to much.

For each point x in stratum $\mathcal{E}_i$, we define a cone of direction $\nabla U(x)$ and angle α_i. The angle α_i is estimated in the course of the algorithm until the flat histogram has been reached N_{FHE} times – see section 2.5.

Estimating the angle For a stratum $\mathcal{E}_i$, we wish to select an angle amidst the set $\alpha_i^{(0)}, ..., \alpha_i^{(k)}$. We apply the following procedure–which is independent from the generation of x_{t+1}. For a given point $x_t \in \mathcal{E}_i$ sampled by Wang-Landau, consider the cones of apex x_t, direction $\nabla U(x_t)$, and aperture angles $\alpha_i^{(j)}$. We sample M directions uniformly in each cone, and check for each such direction whether the stratum $\mathcal{E}_{i-1}$ (or respectively $\mathcal{E}_{i+1}$) can be reached. Once a prescribed number of points x_t has been processed, we compute the probability of picking a direction which reaches $\mathcal{E}_{i-1}$ (or respectively $\mathcal{E}_{i+1}$) for each cone. Then, we select the widest angle such that this probability is larger than a user defined threshold. If no such angle exist, we might rely on what is described in section 3.5.

Remark 3. *The previous strategy calls for the following comments:*

- *The aperture angle of the cone is actually critical. Consider the set of directions delimited on S^{d-1} by a given cone. When the dimension increases, the mass of this set of directions concentrates on the boundary of the cone. Therefore, if the cone is too large, sampled directions will end up with high probability in this region, and the corresponding random walk will miss the targeted stratum.*

- *Since cones are isotropic, the method is not suited to handle highly non isotropic cases. However as we will see in the experiments, it works well enough even for moderately non isotropic cases.*

- *Even if a suitable cone is found by the previous procedure, the Metropolis Hasting acceptance rate might be low – for instance if strata are too wide or the curvature of level sets non constant.*

Uniform direction in a cone. Sampling a uniform direction is non trivial. we provide, a detailed sampling algorithms in the appendix – section 7.

3.4 Handling multimodal distributions via darting

3.4.1 Problem

For classical random walks, transitions between minima of multimodal distributions are rare events, inducing long mixing times.

3.4.2 Solution

Assuming one has a *a priori* knowledge of the positions $m_1, .., m_K$ of theses minima, it is natural to introduce another type of move allowing jumps from one minima to another. To implement this, we use a darting strategy – see [21, 22] and section 2.4.

Darting in its simplest form defines a radius ρ, then add transitions between the balls $B(m_i, \rho)$. In practice, if $x_t \in B(m_i, \rho)$, one picks a ball at random, call its index j, and proposes the following move: $x_{t+1} \sim Unif(B(m_j, \rho))$. However the balls $B(m_i, \rho)$ do not match the level set surfaces of their respective basins. Hence $U(x_{t+1}) - U(m_j)$ might be much larger than $U(x_t) - U(m_i)$, leading to poor acceptance rates in the Wang-Landau algorithm.

Choice of the candidate point. Denoting m_{k_t} the local minimum whose basin contains x_t, define $\Delta U_t = U(x_t) - U(m_{k_t})$. Our rationale to optimize the acceptance ratio in Wang-Landau is to control both $\Delta U_t - \Delta U_{t+1}$ and the ratio $q(y, x)/q(x, y)$. For the former, we proceed as follows in two steps. First, we chose a target energy. For a given x_t, let k be the index of the minimum chosen at random. For some $\beta > 0$, we choose a target energy

$$T_U \sim Unif(U(m_k) + \Delta U_t - \beta, U(m_k) + \Delta U_t + \beta). \tag{13}$$

Second, we propose a point x_{t+1} such that $U(x_{t+1}) = T_U$ (Fig. 6). To this end, we sample a direction u uniformly in the ellipsoid defined by $v^T H(k) v = 1$ where $H(k)$ is the Hessian of U at m_k. Then we do a line search to find the intersection between the target energy T_U and the half line $m_k + \mathbb{R}^+ u$ (Fig. 7).

When to jump. The previous strategy requires a line search which is expensive if the target point is far from the local minimum. Furthermore, under some assumptions which are true if jumps are only allowed close to the local minima, the expression of the transition kernel can be simplified. Hence we introduce M a user defined parameter, and the darting move set is only used if $\Delta U_t \leq M$.

Transition kernel. Computing the transition kernel of the darting move is non trivial, but can be done using a suitable change of variable. The full computation is detailed in the appendix – section 8, however for the sake of brevity, we only give here the final result. For any x and y in $\mathbb{R}^n$, let k be the closest minima to y, $I_k = [U(m_k) + \Delta U - \beta, U(m_k) + \Delta U + \beta]$, λ_i and e_i the eigenvalues and eigenvectors of the Hessian of U at m_k. Finally, let

$$l = \sqrt{\sum_i \lambda_i < y - m_k, e_i >^2}. \tag{14}$$

Using the latter, the the transition probability is given by:

$$q_{dart}(x, y) = \frac{1}{K} \frac{\Gamma\left(\frac{n}{2}\right) 2\beta}{2\pi^{\frac{n}{2}}} 1_{I_k}(U(y)) \frac{1}{l^n} \nabla U(y)^T (y - m_k) \prod_{i \leq n} \sqrt{\lambda_i}. \tag{15}$$

3.5 Splitting energy bins

As a general rule, more bins means slower convergence speed. Thus we only split bins when the cone strategy fails, as a fallback. We monitor the failure of the cone strategy by computing the proportion of steps in which the random walk has the *possibility* to go up or down in energy (see 3.2) and the success rate of the metropolis hasting criterion when going up or down. If either of these statistics are too low for a given bin, the bin is split in half.

3.6 The random walk

The final random walk combines the previous random walks. Let $p = (p_{flat}, p_{cone}, p_{darting})$ such that $p_{flat} + p_{cone} + p_{darting} = 1$. The final random walk is:

- chose one of the random walk at random with probability vector p
- sample a point according to the chosen random walk

Such random walk has the following transition probability:

$$q(x, y) = p_{flat} q_{flat}(x, y) + p_{cone} q_{cone}(x, y) + p_{darting} q_{darting}(x, y) \tag{16}$$

Remark 4. *The quantities $p_{flat}, p_{cone}, p_{darting}$ are parameters, which may depend upon the location x. Note that in the setting $p_{cone} = 0$, the the improvement yielded by the cone is unused.*

4 Experiments

4.1 Setup

4.1.1 Statistics of interest

Our experiments target three points: correctness, mixing time, and ability to handle complex systems. For the sake of conciseness, time t refers to t steps of Wang-Landau.

Correctness, stability, and convergence. When an analytical solution for θ^* is known, we simply resort to the relative error for estimates at time t, defined by:

$$\text{error}(t) = \sum_i \frac{|\theta_i^* - \theta_i(t)|}{\theta_i^*}. \tag{17}$$

We plot this function along time.

When no analytical solution is known–see the dialanine model below, we assess convergence in two ways, based on several runs (N=60). First, we plot an observable along time, akin to the partition function, at a fixed temperature:

$$\frac{Z}{\lambda(\mathcal{E})} = \frac{1}{\lambda(\mathcal{E})} \int_{\mathcal{E}} \exp\left(-U(x)/kT\right) \approx \sum_{\text{Energy levels U}} \theta_i^* \exp\left(-U/kT\right). \tag{18}$$

Second, we provide box plots on a per bin basis. We also resort to violin plots when more details are required–in terms of modes of the distribution.

Mixing time. A classical assessment of the mixing time is in terms of auto-correlation as a function of the lag time [26]. In our setting, where diffusivity across energy strata is targeted, a simpler proxy for the mixing time of P_θ is provided by the climbing and descending times.

A *climb* across d strata is defined by two times t_0 and t_1 such that

- $x_{t_0} \in \mathcal{E}_0$ and $x_{t_0-1} \notin \mathcal{E}_0$.
- $x_{t_1} \in \mathcal{E}_{d-1}$ and $\forall t \in [t_0, t_1[$, $x_t \notin \mathcal{E}_{d-1}$.

The *climbing time* is then $t_1 - t_0$.

Note that we do not normalize the climb times by the number of strata. Indeed, as seen in section 3.5, increasing the number of strata can decrease the mixing time. Hence the number of strata is a parameter tuned for convergence speed and therefore should not be taken into account when measuring mixing time.

In the context of multi-modal distributions, another proxy for the mixing is the time taken by the random walk to go from one mode to the other. To that end, we monitor the time evolution of the proportion of time spent in one of the modes.

4.1.2 Contenders

As a yardstick, we compare our random walk (section 3.6) against the Gaussian random walk. However, while our random walk do not require parameter tuning, the Gaussian variance needs to be tuned for a fair comparison [26]. Hence we compare with 3 Gaussian walks with high, adequate and low variances with respect to the tuned variance.

4.1.3 Models

Analytical models. We study three analytical models. The first model is the isotropic harmonic potential. The second is a non isotropic harmonic potential, to ensure that the algorithm behaves correctly in non trivial settings (Remark 3). The last model is a potential with two local minimum designed to study darting.

Molecular model. Finally, our last system is the usual toy molecular system, namely the blocked alanine peptide Ace-Ala-Nme (Fig 11), referred to as dialanine for the sake of conciseness.

We use the amber99-sb force field in vacuum and aim to compute the density of state between -21 kcal/mol and 4 kcal/mol associated to one local minima – by enforcing the simulation to remain inside the basin of this local minimum ($\phi = 59.8862, \psi = -35.5193$).

4.2 Results

4.2.1 Single well potential

We study here the simple harmonic model

$$U(x) = \sum x_i^2$$

with state space the unit ball in dimension $n = 30$. Since the exact result is know, we plot the exact error.

Let us first focus on the error (Fig. 8(Top)). The cone strategy yields the best results; all remaining strategies fail to converge in the imparted time (10^7 steps). This radical improvement owes to a climb time orders of magnitude smaller for the cone strategy (Fig. 8(Bottom)).

The comparable performances of the Gaussian random walk with ours (with cone off) owes to the fact that on this example, with the chosen discretization, strata overstepping is not critical.

4.2.2 Single well potential - no isotropic

To challenge all methods with a non-isotropic case (Remark 3), we use the following non isotropic potential energy:

$$U(x) = \sum_{i=1}^{n} i x_i^2$$

Also in dimension $n = 30$, the results are on par with the isotropic case (Fig. 9).

4.2.3 Dual wells potential: darting

To challenge darting, we use the usual one-dimensional dual well potential energy function $x^4 - x^2$, and add a quadratic potential in other dimensions:

$$U(x) = x_1^4 - x_1^2 + \sum_{i=2}^{n} x_i^2.$$

This potential energy has 2 local minimum at $(-\frac{1}{\sqrt{2}}, 0, ..., 0)$ and $(\frac{1}{\sqrt{2}}, 0, ..., 0)$. The additional coordinates makes it harder to travel from one minimum to the other by making it hard to choose suitable directions.

We setup the darting move set with these two minima, and compare our random walk with and without darting.

Both methods yield correct values (Fig. 10(Top)). (Data not shown for darting disabled: since the potential energy function is symmetric for the first coordinate, the algorithm computes the correct value even if it never crosses the energy barrier.)

It appears that the random walk allows crossing the energy barrier almost instantly, while with darting disabled the first jump appears after 10^5 samples (Fig. 10(Bottom)). This induces a large difference in the mixing time.

4.2.4 Dialanine

To compute the value defined by Eq. (18) restricted to the basin of the local minimum with torsion angles, we enforce the simulation to remain within this basin. Checking whether a point is in a given minima basin of attraction requires a minimization of the potential energy. Since this is costly operation, we check this condition every N (=100) steps. If the random walk has escaped, we roll back to the latest point in this basin. (Note that this requires downgrading all statistics and random number generators [27].)

Remark 5. *The previous roll back strategy may introduce some bias, as an excursion outside the basin may not be detected. The effectiveness of this strategy relies on a bounded proportion of roll backs, see [27].*

We perform 60 runs, each with 10^7 steps.

To analyze convergence, we plot the time evolution of the observable defined from the partition function at $T = 300K$ (Eq. (18), Fig. 12(Top)). Since all simulations use the same number of bins, we also provide a box plot for each bin Fig. 12(Middle)). Finally, to check whether the observable is unimodal or not, we perform a violin plot at three different time frames along the course of the simulation (Fig. 12(Bottom)).

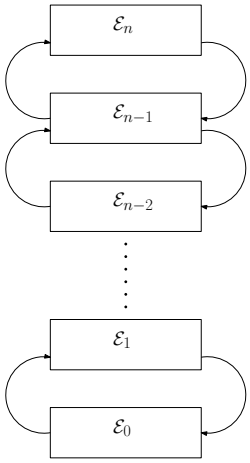
We note that the convergence was reached to a different extent as a function of the volume of strata: the smaller the volume of a stratum, the higher the variance of estimates. This is expected as sampling rare events is always more challenging. It appears, though, that the observable converges (Fig. 12(Bottom)), since values concentrate along a single mode.

To get further insights, we study the contribution of individual strata reweighted by Boltzmann's factor (Fig. 13). We first note that the energy range used is sufficiently large since contributions of the last stratum is one order of magnitude smaller than the highest one (Fig. 13(Top)). The more detailed violin plots–not in log scale, also shows that distributions within strata are unimodal.

5 Outlook

Given a physical system characterized by an energy, the Wang-Landau algorithm is a stochastic method returning an estimation of the density of states in terms of histogram. A core component of the method is the random walk used to navigate between the strata i.e. the preimages in configuration space of the energy slices defining the histogram. In this work, we make an explicit link between the convergence of the Wang-Landau algorithm and mixing properties of the underlying random walk. This analysis prompted the development of a novel, parameter-free random walk. This random walk embarks three components which respectively target the following three difficulties: avoiding overstepping strata, coping with the curse of dimensionality, and accommodating multi-model distributions. The geometry awareness removes the necessity of tuning the variance of the random walk while performing well with strata of varying width. The cone improvement is crucial to deal with concentration phenomena in high dimensional

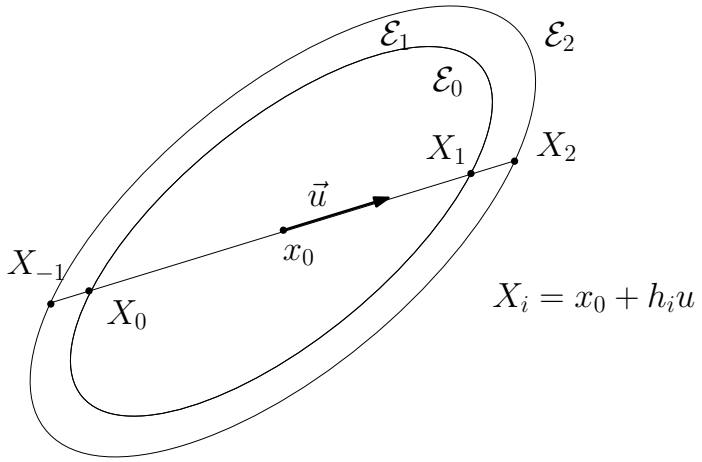
Figure 1 Connectedness between pre-images of energy bins is prime to fast convergence. Energy levels may be seen as the nodes of a graph and may be connected in a variety of ways. In this work, we exploit a random walk aiming at describing a *ladder* to connect these nodes.



problems. Darting is necessary for multimodals distributions with low transition probabilities between modes. The performances of our random walk are assessed by measuring so-called climbing times which quantify the diffusivity across strata. All in all, the resulting Wang-Landau algorithm is effective is computing observables for small biomolecules, within hours on a laptop computer.

Our work calls for developments on two types of questions. On the design side, while our random work operates in Cartesian coordinates, switching to internal coordinates is an appealing strategy to handle biomolecules whose conformational changes are best described by valence and torsion angles. On the analysis side, our assessment is experimental and prompts challenging analysis issues. On the one hand, a rigorous analysis of the mixing time of our random walk would provided insights on which geometric features of the conformational space / landscape matter. On the other hand, bridging the gap between the mixing time of the random walk and the convergence speed of WL would be of high interest.

Figure 2 Exploiting the geometry of the landscape to avoid overstepping strata. The intersection between a random line through x_0 with the level set surfaces of a quadratic approximation of the potential yields points $\{X_i\}$ from which the random walk is defined – see main text.



6 Artwork

6.1 Method

Figure 3 Reaching region $\mathcal{E}_{i-1}$ from $\mathcal{E}_i$: cone of suitable directions.

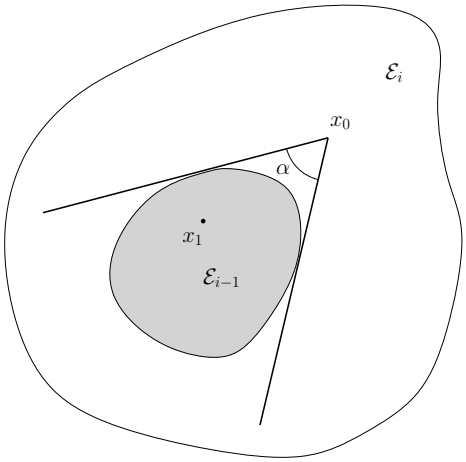


Figure 4 Ratio between the area of the spherical cap subtained by an angle θ and that of the whole n-dimensional hemisphere $S^{n-1}/2$. Ranges explored: dimension $n \in [3, 100]$, and angle $\theta \in [0, \pi/2]$.

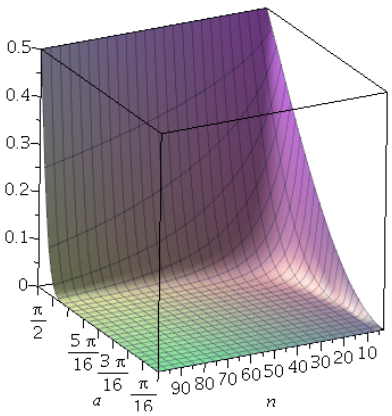


Figure 5 Handling multimodal distributions via darting: jumping between 2 minima. While darting, the difference of energy with the local minima is controlled to monitor the acceptance rate.

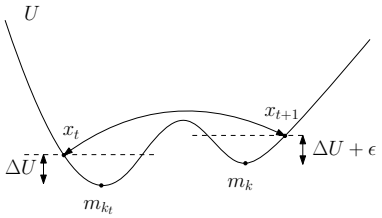
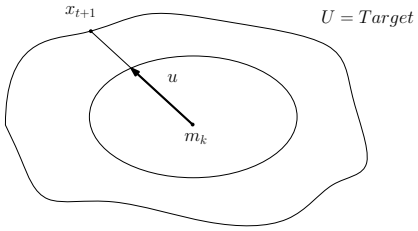
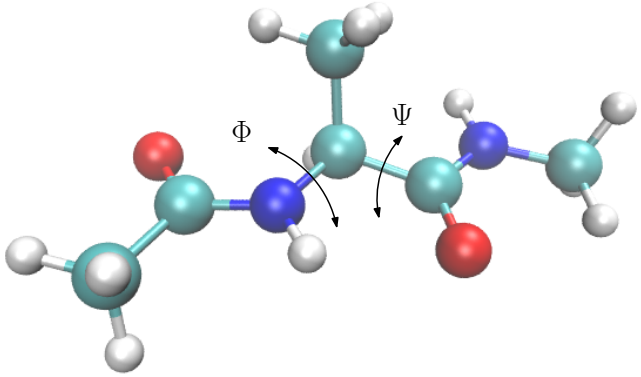


Figure 6 Darting: reaching a prescribed energy level via line search in direction u. The line search starts from minimum m_k with target energy $Target$.



6.2 Results

Figure 10 Dialanine (Ace-Ala-Nme) and the two dihedral angles Φ and Ψ

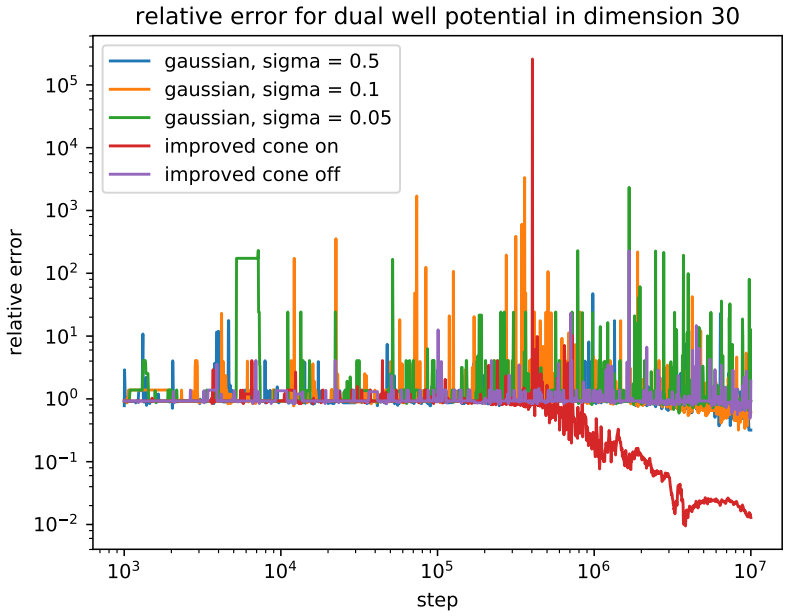


References

[1] D. J. Wales. *Energy Landscapes*. Cambridge University Press, 2003.

[2] T. Lelièvre, G. Stoltz, and M. Rousset. *Free energy computations: A mathematical perspective*. World Scientific, 2010.

Figure 7 Isotrop single well in dimension 30: comparison of the five random walks. The five random walks used are the three Gaussian based RW, plus the improved random walk with and without the cone improvement. **(Top) Comparison of the evolution of relative error – Eq. (17) (Bottom) Box plot of the climbing times.**



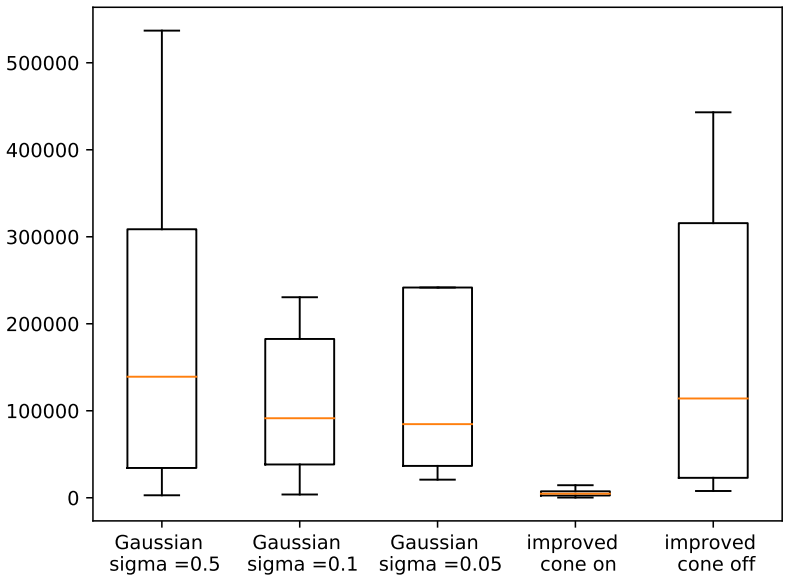
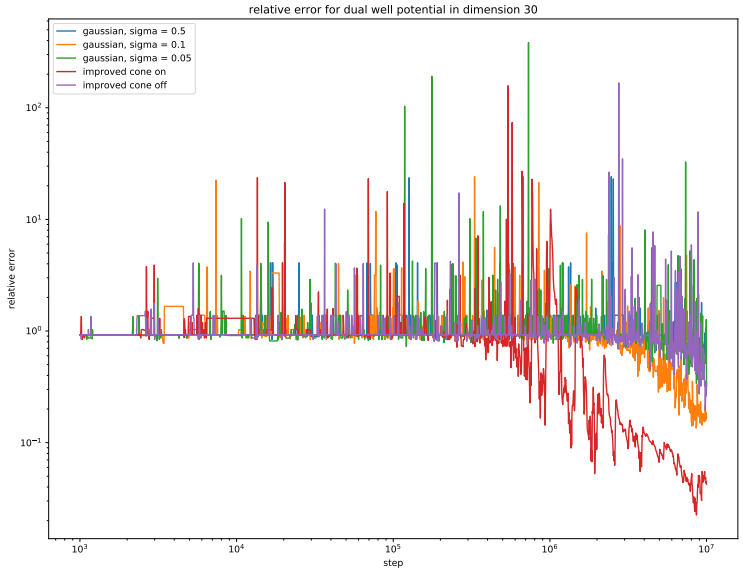


Figure 8 Non isotropic single well in dimension 30: comparison of the five random walks. The five random walks used are the three Gaussian based RW, plus the improved random walk with and without the cone improvement. **(Top) Comparison of the evolution of relative error – Eq. (17) (Bottom) Box plot of the climbing times.**



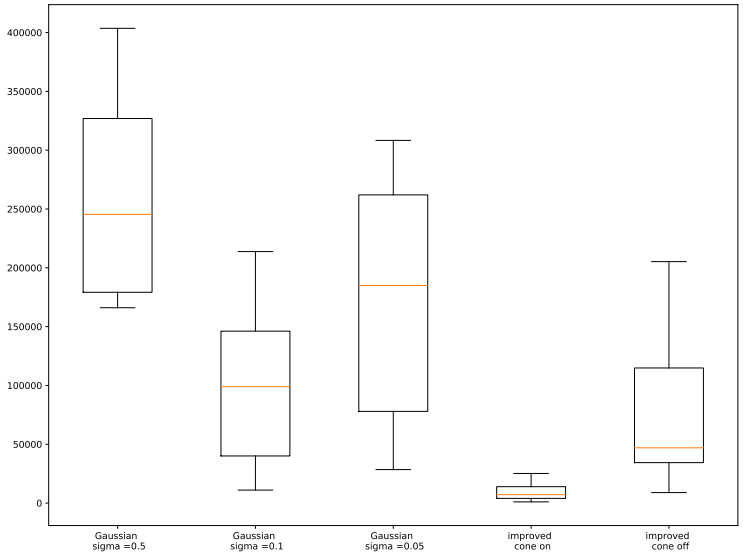


Figure 9 Dual well potential in dimension 30: analysis of darting. (Top) Evolution of relative error – Eq. (17) (Bottom) Comparison of time spent in the first well with and without darting

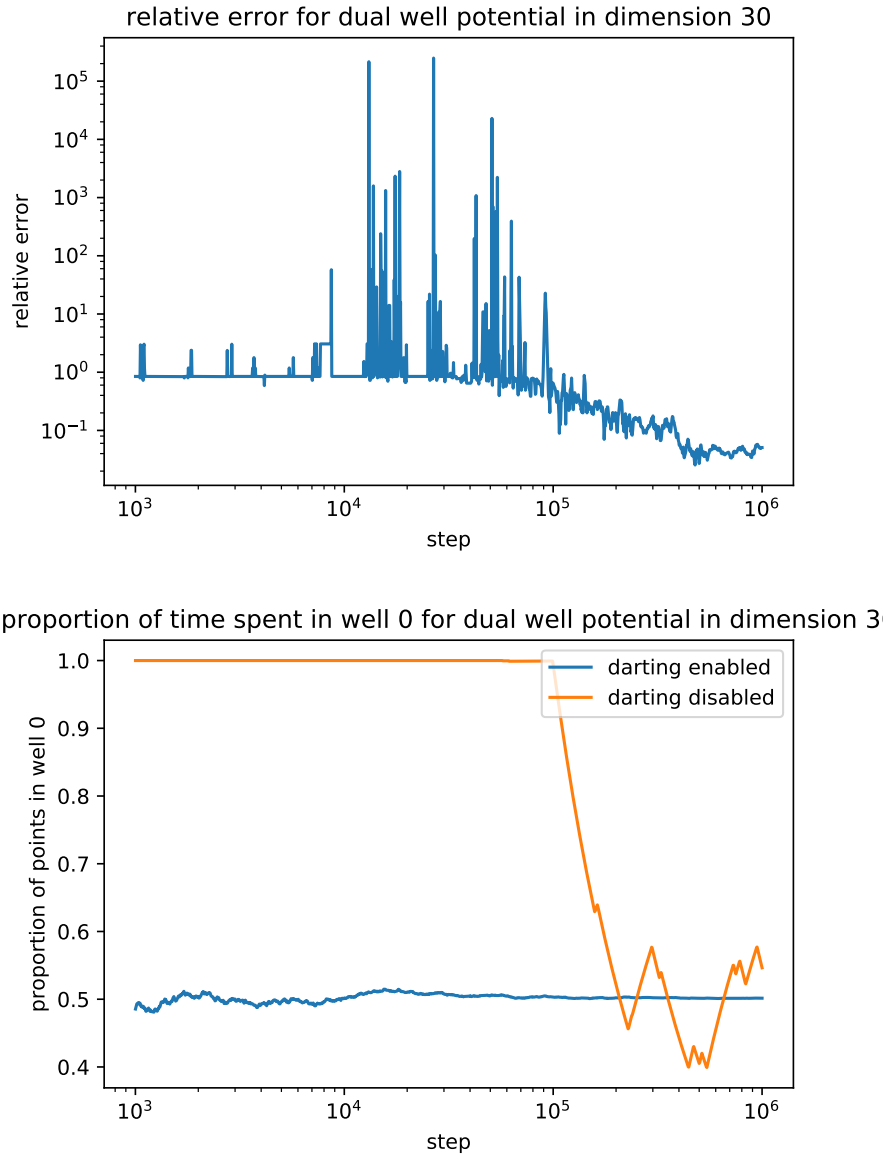
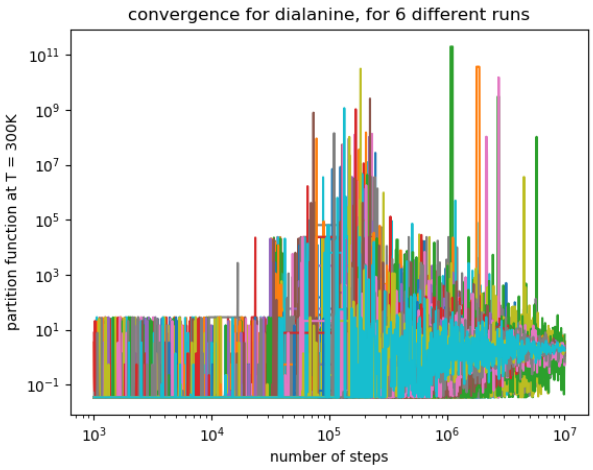
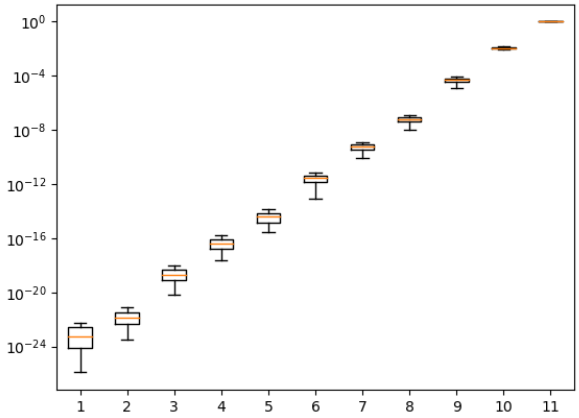


Figure 11 Analysis of convergence for dialanine, using amber-69sb forcefield in vacuum. Results averaged over 60 independant simulations. **(Top) evolution of the partition function. (Middle) Box plot of the estimation θ_i for each bin i. (Bottom) Violin plot of the partition function at $T = 300K$, at three different time frames along the course of the simulation.**





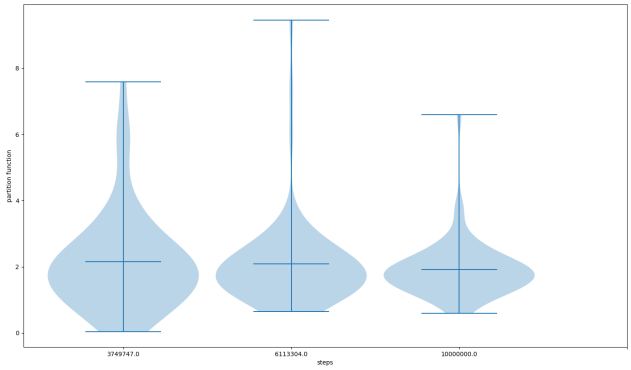
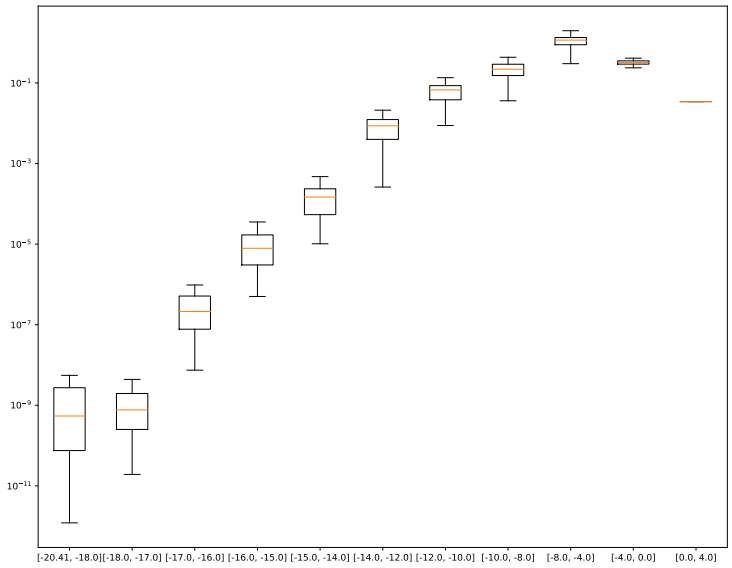
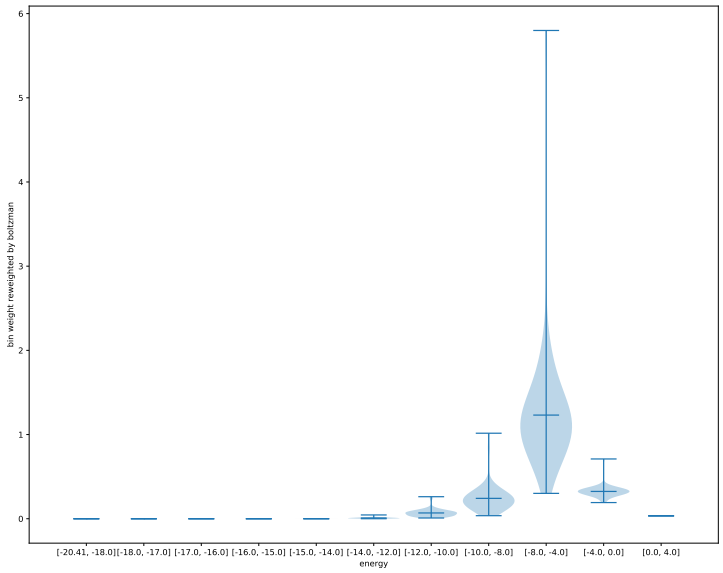


Figure 12 Analysis of convergence for dialanine, using amber-69sb forcefield in vacuum. Results averaged over 60 independant simulations. **(Top) Violin plot of the final bins volume with respect to the Boltzman distribution at $T = 300K$ (Bottom) Violin plot of the final bins volume with respect to the Boltzman distribution at $T = 300K$**





[3] W. Janke. Monte carlo simulations in statistical physics: From basic principles to advanced applications. *Order, Disorder and Criticality: Advanced Problems of Phase Transition Theory*, 3:93–166, 2012.

[4] D. Landau and K. Binder. *A guide to Monte Carlo simulations in statistical physics*. Cambridge university press, 2014.

[5] F. Wang and D.P. Landau. Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical review letters*, 86(10):2050, 2001.

[6] D.P Landau, S-H. Tsai, and M. Exler. A new approach to monte carlo simulations in statistical physics: Wang-landau sampling. *American Journal of Physics*, 72(10):1294–1302, 2004.

[7] R.E. Belardinelli and V.D. Pereyra. Fast algorithm to calculate density of states. *Physical Review E*, 75(4):046701, 2007.

[8] R.E. Belardinelli and V.D. Pereyra. Wang–Landau algorithm: A theoretical analysis of the saturation of the error. *The Journal of chemical physics*, 127(18):184105, 2007.

[9] L. Bornn, P. Jacob, P. Del Moral, and A. Doucet. An adaptive interacting Wang–Landau algorithm for automatic density exploration. *Journal of Computational and Graphical Statistics*, 22(3):749–773, 2013.

[10] R. Belardinelli and V. Pereyra. Nonconvergence of the wang-landau algorithms with multiple random walkers. *Physical Review E*, 93(5):053306, 2016.

[11] P. Jacob and R. Ryder. The Wang–Landau algorithm reaches the flat histogram criterion in finite time. *The Annals of Applied Probability*, 24(1):34–53, 2014.

[12] G. Fort, B. Jourdain, E. Kuhn, T. Lelièvre, and G. Stoltz. Convergence of the wang-landau algorithm. *Mathematics of Computation*, 84(295):2297–2327, 2015.

[13] F. Lou and P. Clote. Thermodynamics of RNA structures by wang–landau sampling. *Bioinformatics*, 26(12):i278–i286, 2010.

[14] P. Poulain, F. Calvo, R. Antoine, M. Broyer, and P. Dugourd. Performances of wang-landau algorithms for continuous systems. *Physical Review E*, 73(5):056704, 2006.

[15] Pedro Ojeda-May and Martin E Garcia. Electric field-driven disruption of a native β-sheet protein conformation and generation of a helix-structure. *Biophysical journal*, 99(2):595–599, 2010.

[16] A. Swetnam and M. Allen. Improving the wang–landau algorithm for polymers and proteins. *Journal of computational chemistry*, 32(5):816–821, 2011.

[17] W. Janke and W. Paul. Thermodynamics and structure of macromolecules from flat-histogram monte carlo simulations. *Soft matter*, 12(3):642–657, 2016.

[18] F. Cazals and T. Dreyfus. The Structural Bioinformatics Library: modeling in biomolecular science and beyond. *Bioinformatics*, 7(33):1–8, 2017.

[19] W. Atisattaponga and P. Marupanthornb. A 1/t algorithm with the density of two states for estimating multidimensional integrals. *Computer Physics Communications*, 220(122–128), 2017.

[20] L. Tierney. A note on Metropolis-Hastings kernels for general state spaces. *Annals of applied probability*, pages 1–9, 1998.

[21] I. Andricioaei, J.E. Straub, and A.F. Voter. Smart darting Monte Carlo. *The Journal of Chemical Physics*, 114(16):6994–7000, 2001.

[22] C. Sminchisescu and M. Welling. Generalized darting Monte Carlo. *Pattern Recognition*, 44(10):2738–2748, 2011.

[23] Z. Li and H.A. Scheraga. Monte Carlo-minimization approach to the multiple-minima problem in protein folding. *PNAS*, 84(19):6611–6615, 1987.

[24] A. Roth, T. Dreyfus, C.H. Robert, and F. Cazals. Hybridizing rapidly growing random trees and basin hopping yields an improved exploration of energy landscapes. *J. Comp. Chem.*, 37(8):739–752, 2016.

[25] J.S. Rosenthal and G.O. Roberts. Coupling and ergodicity of adaptive MCMC. *Journal of Applied Probablity*, 44:458–475, 2007.

[26] G. Roberts and J. Rosenthal. Optimal scaling for various metropolis-hastings algorithms. *Statistical science*, 16(4):351–367, 2001.

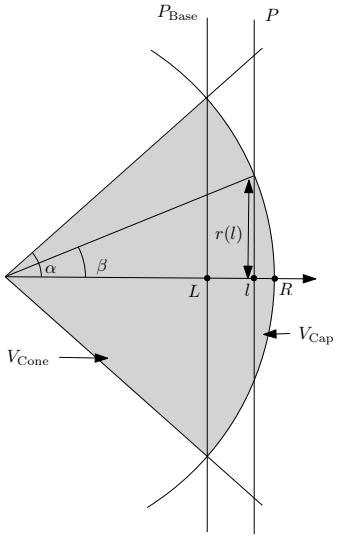
[27] A. Chevallier and F. Cazals. A generic framework for the Wang-Landau algorithm.

[28] S. Li. Concise formulas for the area and volume of a hyperspherical cap. *Asian Journal of Mathematics and Statistics*, 4(1):66–70, 2011.

7 Appendix: uniform sampling in a hypercone

We wish to sample uniformly at random in the intersection of a cone of aperture α intersected with a n-dimensional ball $B^n(R)$ as described in Fig.14. The algorithm and the calculations use the notations of Fig. 14.

Figure 13 Uniform sampling within a conic region. The volume defined by the grey region is the union of a cone and of a spherical cap.



7.1 Pre-requisites

Special functions. We shall need the Beta and incomplete Beta functions, defined by

$$\begin{cases} B(a,b) = \int_0^1 t^{x-1}(1-t)^{y-1}dt, \\ B(x;a,b) = \int_0^x t^{x-1}(1-t)^{y-1}dt(\text{ with } 0 < x < 1). \end{cases} \tag{19}$$

Using both, one defines the regularized incomplete Beta factor

$$I_x(a,b) = \frac{B(x;a,b)}{B(a,b)}. \tag{20}$$

Spheres and balls: surface and volume. The surface area of a sphere of a $n-1$ sphere $S^{n-1}(R)$ of radius R in $\mathbb{R}^d$

$$\text{Area}_{n-1}(R) = R^{n-1}\frac{2\ \pi^{n/2}}{\Gamma\left(d/2\right)} \equiv R^{n-1}A_n. \tag{21}$$

The volume of the corresponding ball $B^n(R)$ satisfies

$$\text{Vol}_n(R) = R\frac{\text{Area}_n(R)}{n} = R^n\frac{2}{n}\frac{\pi^{n/2}}{\Gamma\left(n/2\right)} = R^n\frac{\pi^{n/2}}{\Gamma\left(n/2+1\right)} \equiv R^nV_n. \tag{22}$$

To generate a point X uniformly at random on the unit the unit sphere S^n, we generate a point $X = (x_1, \ldots, x_n)^t$ whose coordinates are iid Gaussian with $\mu = 0$ and $\sigma = 1$. The corresponding density is given by

$$f_G(X) = \frac{1}{(2\pi)^{n/2}} e^{-\frac{x_1^2+x_2^2+\cdots+x_n^2}{2}}. \tag{23}$$

To obtain a unit vector, we normalize the latter as $\frac{X}{\|X\|}$. (NB: due to normalization the coordinates of this vector are not independent.)

Random generation within a ball. To generate a point uniformly at random inside $B^n(R = 1)$, observe that the volume of $B^n(r) = r^n V_n$. Differentiating yields

$$\frac{d}{dr}(r^n V_n) = dr^{n-1} V_n. \tag{24}$$

Therefore one generate a random value using the density dr^{n-1} for $r \in [0, 1]$.

Spherical caps of the n-dimensional ball. We consider a conic region inside the n-dimensional ball, consisting of the union of a pyramid and that of a spherical cap defined by the cone of aperture α (Fig. 14). Surface and volume of such a cap is easily computed [28].

To compute the volume of the cap, we integrate the volume of a $n-1$ dimensional sphere or radius $r \sin\beta$ whose height element is $d(r\cos\beta) = r\sin\beta$:

$$V_{n,\alpha}^{\text{Cap}}(r) = \int_0^\alpha \text{Vol}_{n-1}(r\sin\beta) d(r\cos\beta) = \frac{\text{Vol}_n(r)}{2} I_{\sin^2\alpha}(\frac{n+1}{2}, \frac{1}{2}). \tag{25}$$

Note that the incomplete Beta factor as the probability for a point of the ball to also be inside the spherical cap.

To compute the surface of the cap, we integrate the area of a $n-1$ dimensional sphere or radius $r\sin\beta$ with arc element $rd\beta$:

$$A_{n,\alpha}^{\text{Cap}}(r) = \int_0^\alpha \text{Area}_{n-1}(r\sin\beta) r d\beta = \frac{\text{Area}_{n-1}(r)}{2} I_{\sin^2\alpha}(\frac{n-1}{2}, \frac{1}{2}). \tag{26}$$

7.2 Algorithm to uniformly sample a hypercone

7.2.1 Overview

The algorithm proceeds in 3 steps:

- (i) Decide whether one samples from the cone or the spherical cap,
- (ii) Pick a slice in the cone or spherical cap,
- (iii) Sample the slice.

More formally:

- (i) Draw $u \in [0, V_{n,\alpha}^{\text{Cone}}(1) + V_{n,\alpha}^{\text{Cap}}(1)]$. If $u < V_{n,\alpha}^{\text{Cone}}(1)$, we pick in the cone (ie $l < L$), else we pick in the cap (ie $l > L$).

- (ii) Pick a slice of the cone or the cap at distance l from the center, using the density

$$f_{cone}(l) = C_{cone} r(l)^{n-1} 1_{l \leq L} = C_{cone} \tan(\alpha)^{n-1} l^{n-1} 1_{l \leq L} \tag{27}$$

or

$$f_{cap}(l) = C_{cap} r(l)^{n-1} 1_{l > L} = C_{cap} (1 - l^2)^{\frac{n-1}{2}} 1_{l > L} \tag{28}$$

with C_{cone} and C_{cap} normalization constants used to define probability densities.

- (iii) Draw uniformly at random in the corresponding $n-1$ ball, using the density from Eq. (24).

7.2.2 Sampling from f_{cap}

The previous algorithm requires sampling from f_{cap} defined in eq.(28). The most straightforward way to sample from a probability density is to compute the inverse of the cumulative distribution function ($F(x) = \int_{-\infty}^{x} f(y)dy$). This requires to compute a primitive of the density. However, see BEFORE, there is no simple analytic expression for the primitive of f_{cap}. Hence, we fall back to rejection sampling with a well chosen base distribution such that the rejection rate do note depend on the dimension n.

Observe that while $(1 - l^2)^{\frac{n-1}{2}}$ do not have a simple primitive, the function $l(1 - l^2)^{\frac{n-1}{2}}$ do. Therefore we define

$$g_{cap}(l) = M C_{cap} l (1 - l^2)^{\frac{n-1}{2}} \tag{29}$$

with M such that for all l, $g_{cap}(l) \geq f_{cap}(l)$ which is required for rejection sampling. The optimal choice for M is:

$$M = \frac{1}{L} = \frac{1}{cos\alpha}$$

. L $\tilde{g}_{cap}$ the renormalized version of g_{cap}. Assuming we can sample point from $\tilde{g}_{cap}$, the acceptance rate for each l in the rejection algorithm used with f_{cap} and g_{cap} is

$$\frac{f_{cap}(l)}{g_{cap}(l)} = \frac{1}{lM} \leq \frac{1}{M}$$

as $l \leq 1$. Hence the acceptance rate do not depend on n and only on α the opening of the cone.

Sampling from $\tilde{g}_{cap}$: To sample from $\tilde{g}_c ap$ we compute the inverse of it's cumulative distribution.

Let

$$B(x) = \int_{L}^{x} l(1 - l^2)^{\frac{n-1}{2}} dl$$

using the change of variable $y = 1 - l^2$, we deduce:

$$\begin{aligned} B(x) &= \left[-\frac{(1 - y^2)^{(1+n)/2}}{1 + n} \right]_{L}^{x} \\ &= \frac{(1 - L^2)^{(1+n)/2}}{1 + n} - \frac{(1 - x^2)^{(1+n)/2}}{1 + n} \end{aligned}$$

The cumulative distribution for $\tilde{g}_{cap}$ is

$$\begin{aligned} F(x) &= 1_{x>L} \frac{B(x)}{B(1)} \\ &= 1_{x>L} \left(1 - \frac{(1-x^2)^{(1+n)/2}}{(1-L^2)^{(1+n)/2}}\right) \end{aligned}$$

And it's inverse:

$$F^{-1}(x) = \sqrt{1 - (1-L^2)(1-x)^{2/(n+1)}}$$

Hence we can sample from $\tilde{g}_{cap}$.

7.2.3 Sampling from f_{cone}

The inverse CDF for f_{cone} is straightforward to compute:

$$F^{-1}_{cone}(x) = Lx^{1/n}$$

Therefore we can sample from f_{cone}.

7.3 Changing the cone axis

The previous section algorithm generates a point in a cone whose axis is fixed: $e_1 = (1, 0, ...0)$. In practice, the axis of a cone is aligned with the gradient of the potential energy – Section 3.3.

To handle arbitrary cones, we apply a linear transformation. We describe here how to apply this transformation with a contained complexity. Let $d \in \mathbb{R}^n \setminus \{e_1\}$ be the desired axis of the cone.
Let H be the hyperplane orthogonal to e_1. In the algorithm, we generates points in H. Suppose we generate $(x_2, ..., x_n)$ in H. For any orthonrmal basis $\epsilon_2, ..., \epsilon_n$ of H, the points $x_2\epsilon_2 + ... + x_n\epsilon_n$ will have the same distribution in H. Hence we try to find a basis $\epsilon_2, ..., \epsilon_n$ adapted to our problem.

We choose $\epsilon_2 = \frac{d - < d, e_1 > e_1}{\|d - < d, e_1 > e_1\|}$.
We complete this base with $\epsilon_3, ..., \epsilon_n$, and we will see that the choice of these $\epsilon_3, ..., \epsilon_n$ do not matter.
Let R the rotation such that $R(e_1) = d$ and $R(\epsilon_i) = \epsilon_i$ for $i > 2$.

Let $H_0 = Vect(e_1)$, $H_1 = Vect(e_1, \epsilon_2)$ and $H_2 = Vect(\epsilon_3, ..., \epsilon_n)$.
Let $x \in \mathbb{R}^n$. Then there exists u_1, u_2 and v such that

$$x = u_1 e_1 + u_2 \epsilon_2 + u_3 v$$

with $v = x - < x, e_1 > e_1 - < x, \epsilon_2 > \epsilon_2 \in H_2$. u_1, u_2 and v are straightforward to compute. We easily get:

$$R(x) = R(u_1 e_1 + u_2 \epsilon_2) + u_3 v$$

Thus the transformation R can be reduced to a simple rotate in $\mathbb{R}^2$. Let $\theta = < e_1, d >$. Then

$$R(u_1 e_1 + u_2 \epsilon_2) = u_1 d + u_2 \left(cos(\theta + \pi/2) e_1 + sin(\theta + \pi/2) \epsilon_2\right)$$

Thus we full transform is as follow:

- compute
$$\epsilon_2 = \frac{d- < d, e_1 > e_1}{\|d- < d, e_1 > e_1\|}$$
- compute $u_1 =< x, e_1 >$, $u_2 =< x, \epsilon_2 >$ and $v = x - u_1 e_1 - u_2 \epsilon_2$
- compute $\theta = (e_1, x)$ and $\tilde{d} = (cos(\theta + \pi/2)e_1 + sin(\theta + \pi/2)\epsilon_2)$
- $R(x) = u_1 d + u_2 \tilde{d} + v$

8 Appendix: transition probability for darting

8.1 Notations

We give here a detailed computation of the transition probability for darting given by eq. 15. We use the same notations than in section 3.4. Let us write P_{dart} the Markov kernel associated to the darting move. Let x be a point of $\mathcal{E}$. The transition kernel has a density, hence we write $P(x, y)$ instead of $P(x, dy)$. For a minimum k, let $H(k)$ the Hessian of U at m_k. Let $\lambda_1, ..., \lambda_n$ it's eigenvalues and $e_1, .., e_n$ it's eigenvectors as an orthonormal basis. Finally let $A_k \subset \mathcal{E}$ be the basin of attraction of minimum k and let k_x the minimum such that $x \in A_{k_x}$. We consider the following rescaling of state space:

$$h_k(y) = m_k + \sum_i \sqrt{\lambda_i}(y - m_k|e_i)e_i. \tag{30}$$

Let $\tilde{U}_k(z) = U(h_k^{-1}(z))$ the potential energy in the rescaled space. Let $\tilde{f}_k(u, T_U)$ the application which associates the first intersection between $m_k + \alpha u$ and $\tilde{U} = T_U + U(m_k)$ with $\alpha > 0$. Formally, $\tilde{f}_k$ is an application defined on $S^{n-1} \times \mathbb{R}^+$.

Let $f_k(u, T_U) = h^{-1}(\tilde{f}(u, T_U))$. Also let $f_{k*}(\mu_{k,x})$ be the pushforward measure of $\mu_{k,x}$ by f_k.

Then, the Markov kernel seen as an operator on measures is given by:

$$P_{dart}(x, .) = \frac{1}{K} \sum_k \frac{f_{k*}(\mu_{k,x})}{\int \mu_{k,x}} \tag{31}$$

where $\mu_{k,x}$ is the product measure of the Lebesgue measure on S^{n-1} and the Lebesgue measure of

$$I_k(x) = [U(x) - U(k_x) + U(k) - \beta, U(x) - U(k_x) + U(k) + \beta] \tag{32}$$

8.2 Assumptions

The following assumption ensures that function f_k defines a bijection between the set of directions and the restriction of the target energy level set surface to the basin of a local minimum:

Assumption 1. *For every local minimum $k \leq K$, $u \in S^{n-1}$, $T_U \in [U(m_k), U(m_k) + M]$, the intersection $\{y|y = m_k + \alpha u, \alpha > 0\} \cap \{y|U(y) = T_U\} \cap A_k$ is a single point. See Fig. 15 and Fig. 16.*

Doing a line search for every minimum is expensive. Using constant M defined in Section 3.4(see paragraph *When to jump*), we introduce the following assumption to simplify Eq. (31):

Assumption 2. *For every y such that $U(y) - U(m_{k_y}) \leq M$, then for every $k \leq K$,*

$$\|y - m_{k_y}\| \leq \|y - m_k\|$$

The simplified expression for Eq. (31) reads as

$$P(x, y) = \frac{1}{K} \frac{f_{k_y *}(\mu_{k_y,x})}{\int \mu_{k_y,x}}$$

where k_y is the closest minimum to y. As a final observation, assumptions 1 and 2 are true if M is small enough (using a second order Taylor expansion for the proof at the bottom of the local minima)

Figure 14 Not allowed by assumption 1 as there are multiple intersection point between a direction and the restriction of an energy level set to a basin.

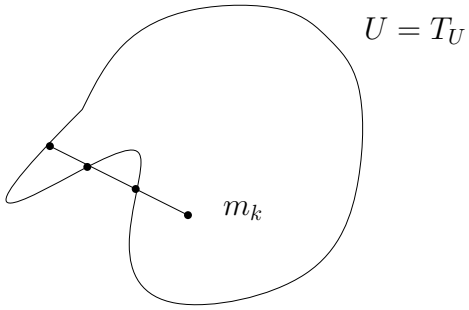
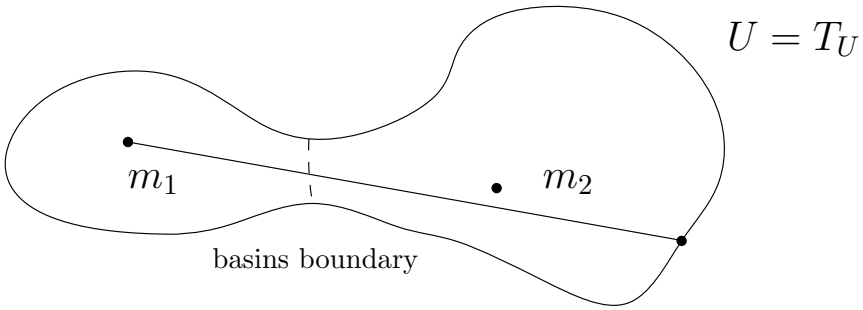


Figure 15 Not allowed by assumption 1 as selected directions yield intersection points outside the basin of m_1.



8.3 Derivation of the transition probability

Under assumption 1, f_k is a bijection from $S^{n-1} \times [U(m_k), U(m_k) + M]$ to the connected component containing m_k of the set of point $\{y | U(y) \leq U(m_k) + M\}$. Hence it's inverse is well defined. The density of the pushfoward measure can be computed using the usual change of variable formula:

$$f_{k*}(\mu_{k,x})(y) = |J(f_k^{-1})(y)| 1_{I_k}(U(y))$$

For notation simplicity, we consider a fixed k and write $f = f_k$ and $h = h_k$ for the following computation. The inverse of f has the following expression:

$$\tilde{f}^{-1}(z) = \left(\frac{z - m_k}{\|z - m_k\|}, \tilde{U}(z) \right)$$

Let $z = h(y)$ and $u = \frac{z - m_k}{\|z - m_k\|}$, and choose $w_1, ..., w_{n-1}$ in $\mathbb{R}^n$ such that $w_1, ..., w_{n-1}, u$ is an orthonormal basis of $\mathbb{R}^n$. Let $l = \|z - m_k\|$. Then:

$$\frac{\partial \tilde{f}^{-1}}{\partial w_i}(z) = \left(\frac{1}{l} w_i, \frac{\partial \tilde{U}}{\partial w_i}(y) \right)$$

Observe that $w_1, ..., w_{n-1}$ is an orthonormal basis of the tangent space of S^{n-1} at u. Then considering that $\tilde{f}^{-1}$ is an application from an open set of $\mathbb{R}^n$ to $S^{n-1} \times \mathbb{R}^+$, the Jacobian of f^{-1} becomes:

$$J(\tilde{f}^{-1})(z) = \begin{pmatrix} \frac{1}{l} & 0 & ... & 0 & 0 \\ 0 & \frac{1}{l} & ... & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & ... & \frac{1}{l} & 0 \\ (\nabla\tilde{U}(z)|w_1) & (\nabla\tilde{U}(z)|w_2) & ... & (\nabla\tilde{U}(z)|w_{n-1}) & (\nabla\tilde{U}(z)|u) \end{pmatrix}$$

Hence

$$|J(\tilde{f}^{-1})(z)| = \frac{1}{l^{n-1}}(\nabla\tilde{U}(z)|u)$$

And using $\tilde{U}(z) = U(h^{-1}(z))$,

$$\frac{\partial\tilde{U}}{\partial u}(z) = \nabla U(y)^T J(h^{-1})(z)u \tag{33}$$

$$= \nabla U(y)^T J(h^{-1})(z)\frac{z - m_k}{l} \tag{34}$$

$$= \nabla U(y)^T J(h^{-1})(z)(h(y) - m_k)\frac{1}{l} \tag{35}$$

$$= \nabla U(y)^T (y - m_k)\frac{1}{l} \tag{36}$$

Where the simplification in equation 36 is justified by the fact that $h(y) - m_k = J(h)(y - m_k) = J(h^{-1})^{-1}(y - m_k)$. Combining the two previous equations:

$$|J(\tilde{f}^{-1})(z)| = \frac{1}{l^n}\nabla U(y)^T (y - m_k)$$

We deduce:

$$|J(f^{-1})(y)| = |J(h)|\frac{1}{l^n}\nabla U(y)^T (y - m_k)$$

The Jacobian matrix of h is easy to compute:

$$|J(h)| = \prod_{i \leq n} \sqrt{\lambda_i}$$

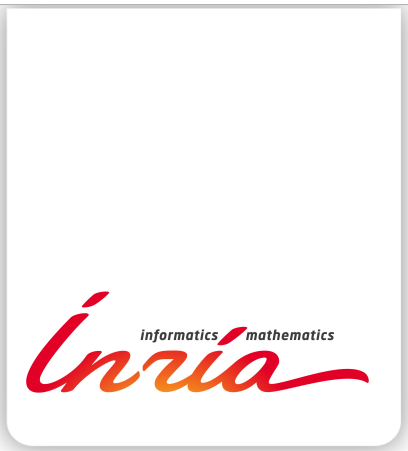
Hence we deduce:

$$f_{k*}(\mu_{k,x})(y) = 1_{I_k}(U(y))\frac{1}{l^n}\nabla U(y)^T (y - m_k) \prod_{i \leq n} \sqrt{\lambda_i} \tag{37}$$

The rescaling factor for measure $\mu_{k,x}$ is:

$$\int \mu_{k,x} = \frac{2\pi^{\frac{n}{2}}}{\Gamma\left(\frac{n}{2}\right) 2\beta} \tag{38}$$

Injecting equations 37 and 38 into equation 31 allows us to compute $P_{dart}(x, y)$.



RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399