



WIKIPEDIA
The Free Encyclopedia

[Main page](#)
[Contents](#)
[Featured content](#)
[Current events](#)
[Random article](#)
[Donate to Wikipedia](#)
[Wikipedia store](#)

Interaction

[Help](#)
[About Wikipedia](#)
[Community portal](#)
[Recent changes](#)
[Contact page](#)

Tools

[What links here](#)
[Related changes](#)
[Upload file](#)
[Special pages](#)
[Permanent link](#)
[Page information](#)
[Wikidata item](#)
[Cite this page](#)

Print/export

[Create a book](#)
[Download as PDF](#)
[Printable version](#)

Languages


[Español](#)
[ไทย](#)

[Edit links](#)

[Create account](#) [Log in](#)

Article [Talk](#)

[Read](#) [Edit](#) [View history](#)

Search 

Wang and Landau algorithm

From Wikipedia, the free encyclopedia

The **Wang and Landau algorithm**, proposed by Fugao Wang and [David P. Landau](#),^[1] is a [Monte Carlo method](#) designed to calculate the [density of states](#) of a system. The method performs a non-markovian random walk to build the density of states by quickly visiting all the available energy spectrum. The Wang and Landau algorithm is an important method to obtain the density of states required to perform a [multicanonical simulation](#).

The Wang–Landau algorithm can be applied to any system which is characterized by a cost (or energy) function. For instance, it has been applied to the solution of numerical integrals^[2] and the folding of proteins.^{[3][4]} The Wang-Landau Sampling is related to the [Metadynamics](#) algorithm.^[5]

Contents

[\[hide\]](#)

- [1 Overview](#)
- [2 Algorithm](#)
- [3 Test system](#)
- [4 Sample code](#)
- [5 Wang and Landau molecular dynamics](#)
- [6 References](#)

Overview [\[edit\]](#)

The Wang and Landau algorithm is used to obtain the [density of states](#) of a system characterized by a cost function. It uses a non-markovian [stochastic process](#) which asymptotically converges to a [multicanonical ensemble](#).^[1] (i.e. to a [Metropolis-Hastings algorithm](#) with sampling distribution inverse to the density of states.) The major consequence is that this sampling distribution leads to a simulation where the energy barriers are invisible. This means that the algorithm visits all the accessible states (favorable and less favorable) much faster than a metropolis algorithm.^[6]

Algorithm [\[edit\]](#)

Consider a system defined on a phase space Ω , and a cost function, E , (e.g. the energy), bounded on a spectrum $E \in \Gamma = [E_{\min}, E_{\max}]$, which has an associated density of states $\rho(E) \equiv \exp(S(E))$, which is to be computed. Because Wang and Landau algorithm works in discrete spectra,^[1] the spectrum Γ is divided in N discrete values with a difference between them of Δ , such that

$$N = \frac{E_{\max} - E_{\min}}{\Delta};$$

Given this discrete spectrum, the algorithm is initialized by:

- setting all entries of the entropy to zero, $S(E_i) = 0 \quad i = 1, 2, \dots, N$
- initializing $f = 1$ and
- initializing the system randomly, by putting in a random configuration $\mathbf{r} \in \Omega$.

The algorithm then performs a [multicanonical ensemble](#) like simulation:^[1] a [Metropolis-Hastings](#) random walk in the phase space of the system with a probability distribution given by

$P(\mathbf{r}) = 1/\rho(E(\mathbf{r})) = \exp(-S(E(\mathbf{r})))$ and a probability of proposing a new state given by a probability distribution $g(\mathbf{r} \rightarrow \mathbf{r}')$. A histogram $H(E)$ of visited energies is stored. Like in the Metropolis-Hastings algorithm, a proposal-acceptance step is performed, and consists in (see [Metropolis-Hastings algorithm overview](#)):

1. proposing a state $\mathbf{r}' \in \Omega$ according to $g(\mathbf{r} \rightarrow \mathbf{r}')$
2. accept/refuse the proposed state according to

$$A(\mathbf{r} \rightarrow \mathbf{r}') = \min \left(1, e^{S - S'} \frac{g(\mathbf{r}' \rightarrow \mathbf{r})}{g(\mathbf{r} \rightarrow \mathbf{r}')} \right)$$

where $S = S(E(\mathbf{r}))$ and $S' = S(E(\mathbf{r}'))$.

After each proposal-acceptance step, the system transits to some value E_i , $H(E_i)$ is incremented by one and the following update is performed:

$$S(E_i) \leftarrow S(E_i) + f.$$

This is the crucial step of the algorithm, and it is what make this Wang and Landau algorithm non-markovian: the [stochastic process](#) now depends on the history of the process. Hence the next time there is a proposal to a state with that particular energy E_i , that proposal is now more likely refused; in this sense, the algorithm forces the system to visit all the spectrum equally.^[1] The consequence is that the histogram $H(E)$ is more and more flat. However, this flatness depends on how well approximated the calculated entropy is to the exact entropy, which naturally depends on the value of f .^[7] To better and better approximate the exact entropy (and thus histogram's flatness), f is decreased after M proposal-acceptance steps:

$$f \leftarrow f/2.$$

It was later shown that updating the f by constantly dividing by two can lead to saturation errors.^[7] A small modification to the Wang and Landau method to avoid this problem is to use the f factor proportional to $1/t$, where t is proportional to the number of steps of the simulation.^[7]

Test system [\[edit\]](#)

We want to obtain the DOS for the [harmonic oscillator](#) potential.

$$E(x) = x^2,$$

The analytical DOS is given by,

$$g(E) = \int \delta(E(x) - E_0) dx = \int \delta(x^2 - E_0) dx,$$

by performing the last integral we obtain,

$$g(E) \propto E^{-1/2},$$

in general, the DOS for a multidimensional harmonic oscillator will be given by some power of E , the exponent will be a function of the dimension of the system.

Hence, we can use a simple harmonic oscillator potential to test the accuracy of Wang–Landau algorithm because we know already the analytic form of the density of states. Therefore we compare the density of states $\rho(E)$ obtained by the Wang–Landau algorithm with $g(E)$.

Sample code [\[edit\]](#)

The following is a sample code of the Wang–Landau algorithm in [Python](#), considering that the proposal is such that

$$\frac{g(\mathbf{x}' \rightarrow \mathbf{x})}{g(\mathbf{x} \rightarrow \mathbf{x}')} = 1$$

The code considers a "system" which is the underlying system being studied.

```
currentEnergy = system.randomConfiguration() # a random initial configuration

while (f > epsilon):
    system.proposeConfiguration() # a proposed configuration is proposed
    proposedEnergy = system.proposedEnergy() # the energy of the proposed
    configuration computed

    if (random() < exp(entropy[currentEnergy]-entropy[proposedEnergy])):
        # if accepted, update the energy and the system:
        currentEnergy = proposedEnergy
        system.acceptProposedConfiguration()
    else:
        # if rejected
        system.rejectProposedConfiguration()

    H[currentEnergy] += 1
    entropy[currentEnergy] += f

    if (isFlat(H)): # isFlat tests whether the histogram is flat (e.g. 95% flatness)
        H[:] = 0
```

Wang and Landau molecular dynamics [edit]

It should be noted that the Wang and Landau algorithm can be implemented not only in a Monte Carlo simulation but also in a molecular dynamics simulation. To do this would require an escalation of the temperature of the system as follows:

$$T'(E) \rightarrow (\partial S(E)/\partial E)T(E),$$

where $S(E)$ is the entropy of the system, $T(E)$ the micro-canonical temperature and $T'(E)$ is the "scaled" temperature used in the simulation.

References [edit]

- ↑ **a** **b** **c** **d** **e** Wang, Fugao and Landau, D. P. (Mar 2001). "Efficient, Multiple-Range Random Walk Algorithm to Calculate the Density of States". *Phys. Rev. Lett.* (American Physical Society) **86** (10): 2050–2053. arXiv:cond-mat/0011174. Bibcode:2001PhRvL..86.2050W. doi:10.1103/PhysRevLett.86.2050. PMID 11289852.
- ↑ R. E. Belardinelli and S. Manzi and V. D. Pereyra (Dec 2008). "Analysis of the convergence of the 1t and Wang–Landau algorithms in the calculation of multidimensional integrals". *Phys. Rev. E* (American Physical Society) **78** (6): 067701. arXiv:0806.0268. Bibcode:2008PhRvE..78f7701B. doi:10.1103/PhysRevE.78.067701.
- ↑ P. Ojeda and M. Garcia and A. Londono and N.Y. Chen (Feb 2009). "Monte Carlo Simulations of Proteins in Cages: Influence of Confinement on the Stability of Intermediate States". *Biophys. Jour.* (Biophysical Society) **96** (3): 1076–1082. Bibcode:2009BpJ....96.1076O. doi:10.1529/biophysj.107.125369.
- ↑ P. Ojeda and M. Garcia (Jul 2010). "Electric Field-Driven Disruption of a Native beta-Sheet Protein Conformation and Generation of alpha-Helix-Structure". *Biophys. Jour.* (Biophysical Society) **99** (2): 595–599. Bibcode:2009BpJ....96.1076O. doi:10.1016/j.bpj.2010.04.040. PMC 2905109. PMID 20643079.
- ↑ Junghans, Christoph, Danny Perez, and Thomas Vogel. "Molecular Dynamics in the Multicanonical Ensemble: Equivalence of Wang–Landau Sampling, Statistical Temperature Molecular Dynamics, and Metadynamics." *Journal of Chemical Theory and Computation* 10.5 (2014): 1843-1847.
- ↑ Berg, B.; Neuhaus, T. (1992). "Multicanonical ensemble: A new approach to simulate first-order phase transitions". *Physical Review Letters* **68** (1): 9–12. arXiv:hep-lat/9202004. Bibcode:1992PhRvL..68....9B. doi:10.1103/PhysRevLett.68.9. PMID 10045099.
- ↑ **a** **b** **c** Belardinelli, R. E. and Pereyra, V. D. (2007). "Wang–Landau algorithm: A theoretical analysis of the saturation of the error". *Jour. Chem. Phys.* **127** (18): 184105. arXiv:cond-mat/0702414. Bibcode:2007JChPh.127r4105B. doi:10.1063/1.2803061.

Categories: Markov chain Monte Carlo | Statistical algorithms | Computational physics

This page was last modified on 29 August 2015, at 08:30.

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.

Privacy policy About Wikipedia Disclaimers Contact Wikipedia Developers Mobile view

