Properties of parallel Wang-Landau algorithm with transition matrix control

Marina A. Fadeeva[a], Lev N. Shchur[a,b]

[a] *HSE University, 101000 Moscow, Russia*
[b] *Landau Institute for Theoretical Physics, 142432 Chernogolovka, Russia*

Abstract

We apply a transition matrix in the energy spectrum to analyze the flatness criteria in each energy window of parallel implementation of the Wang-Landau algorithm. The tunneling time of a random walker in each window does depend on the position of the window in the energy spectrum and is much less than the tunneling time of the entire spectrum. Therefore the parallel Wang-Landau algorithm should be more efficient for larger system sizes. We replace the flatness criterion with the criterion of proximity of the transition matrixes to stochastic ones. A detailed analysis was carried out on the example of the Ising model.

1. Introduction

The Wang-Landau (WL) algorithm [1, 2] is a very powerful tool for direct numerical estimation of the density of states (DOS), which also has a fairly wide application. It overcomes some of the difficulties found in other Monte Carlo algorithms, such as critical slowing down, and allows calculation of thermodynamic observables, including free energy, over a wide range of temperatures in a single simulation. A typical simulation requires at least billions of single Monte Carlo steps even for medium sized systems. Parallel implementation of the Wang-Landau (WL) algorithm is a possible way to speed up the simulation [3].

The two approaches for the parallel WL algorithm was proposed.

The first approach, proposed in the article [4], is implemented by the authors in a simple parallel version, which allows each processor to perform calculations in the entire energy space. An update of the total DoS was performed after some regular simulation intervals, and the updated DoS was used for the next time interval. The histograms were also merged together to check their flatness. Care should be taken in choosing the time interval to tune the efficiency of the entire simulation [4]. Another implementation [5] uses the same energy space parallel simulation approach, but uses parallel threads in shared memory and updates the shared DoS at every MC step. This version can be efficiently implemented on multi-core processors. In any case, both approaches are very sensitive to shared memory size and memory access speed.

The second and more complex approach, named replica exchange Wang-Landau algorithm, is proposed in the articles [6, 7] with the division of the energy space into windows. The trick is that the energy windows must overlap by some sufficient number of energy levels for the algorithm to work correctly, and appropriate flatness criteria protocol and general DoS update procedures must be organized. The algorithm allows simulating very long protein folding models [8].

In this study, we analyze the properties of the replica exchange Wang-Landau (reWL) [6, 7] algorithm. Our analysis is based on the concept of tunneling time [9] and on the properties of the transition matrix in energy space [10].

Tunneling time is the time it takes for a random walker to get from one side of the interval to the other side. The interval in our case is either the entire set of energy level indices, or the set associated with the width of the energy window. Tunneling time is related to the flatness criteria of the WL algorithm because it scales the same as the time it takes a walker to cover the entire possible walking interval.

We analyze the tunneling time in the each energy window and found it depends on the window position in energy spectrum. To clarify our results, we present here a simplified version of the Wang-Landau replica

exchange (reWL) [6, 7] algorithm. We consider only one replica in each window and choose a DoS merge with a simple averaging of overlapping levels.

2. Transition matrix in energy spectrum

It was stated in the paper [10] that the flatness criteria can be understood using the transition matrix in the energy space (TMES)[1], whose elements $T(E_k, E_m)$ are the probabilities of one step random walk for the transition from a configuration with energy E_k to any configuration with energy E_m. The random walk in the Wang-Landau algorithm generated after the two preparatory steps. First, the new configuration chosen using any Markov Chain Monte Carlo algorithm. Second, the new configuration is accepted using the WL-probability, Expr. (1). Accordingly, the walker moves to the energy level E_m or stay at the same energy level E_k. Therefore, this is a combination of two random processes - the matrix elements are affected by both the random process of choosing a new configuration state and the WL-probability of accepting a new state.

It was shown in Ref. [10] that TMES of the WL random walk on true DoS is a stochastic matrix. This means that the probabilities of visiting all energy levels are equal. The largest eigenvalue λ_1 of the stochastic matrix is equal to one. Therefore, the case of the largest eigenvalue close to one is equivalent to the flatness of the histogram in the WL algorithm. In addition, it was proposed to use the parameter $\delta = |1 - \lambda_1|$ as a control parameter in the WL algorithm, i.e. accuracy parameter [10] for DoS estimation.

In this paper, we extend this approach with calculating the transition matrices $T_i(E_k, E_m)$ in each energy window $i = 0, ..., N_w - 1$.

3. Walkers in the windows

The articles [6, 7] present a parallel implementation of the Wang-Landau algorithm using the idea of the Markov chain replica exchange algorithm [12]. At the same time, there are essential difference in the simulations and in the way of information exchange between replicas.

The Markov chain replica exchange method simulates M replicas with a common Hamiltonian $\mathcal{H}$ and at different temperatures and exchanges two replicas with a probability that satisfies the detailed balance of the replica ensemble [12]. Instead, the reWL [6, 7] algorithm simulates kM independent random walkers in M windows of the energy spectrum and exchanges the randomly chosen walkers in two energy windows with a probability based on Wang-Landau probability of acceptance energy level [1, 2]. Both methods are very popular and intensively used in simulations.

Our goal is to shed light on the relaxation processes occurring in the energy windows of the Wang-Landau algorithm and their influence on the accuracy of the global DoS estimate. It is based on the connection between the parameters of the Wang-Landau algorithm and the characteristic times of random walks in the energy spectrum based on the Wang-Landau probabilities of accepting transitions from one energy level to another [13, 9].

There are two characteristic times of the Wang-Landau algorithms, the tunneling time and the mixing time.

3.1. Tunneling time

The Wang-Landau method for estimating the density of energy states [1, 2] can be interpreted as a biased one-dimensional random walk over energy levels [13, 10]. It is known that an unbiased one-dimensional random walk can cover to the distance $\propto \sqrt{n}$ from the starting point after n steps [14]. Therefore, the time required to visit all N energies grows in proportion to the square of the number of levels, i.e. $\propto N^2$. Obviously, in the case of a biased walk, this time will grow faster due to local obstacles in the path of the random walk. Moreover, the randomness in the probabilities could cause a random walker to be trapped in

[1] Do not confuse the TMES matrix with the matrix associated with the transition matrix Monte Carlo method [19, 20].

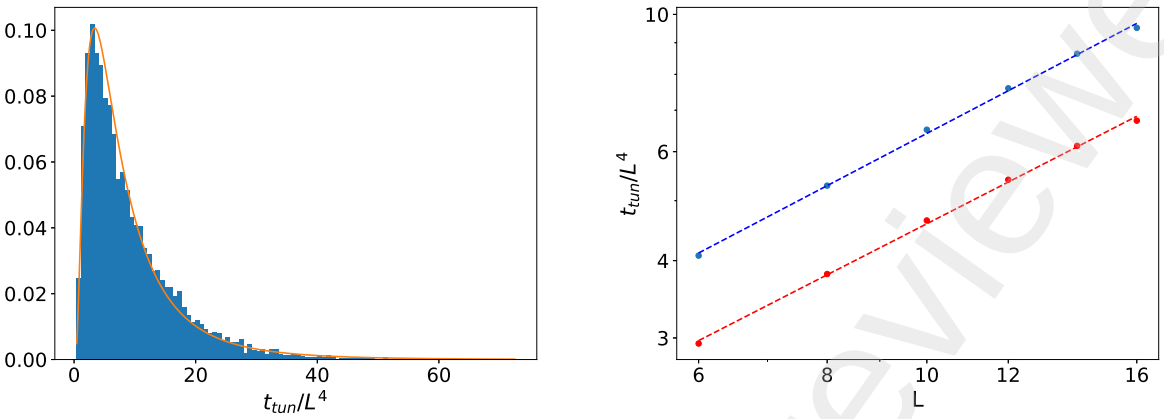

Figure 1: **Left**: Probability density of tunneling time in the Wang-Landau algorithm for the Ising model on a square lattice with linear size $L = 16$, averaged over 10^4 random walks. The result of the fit using the lognormal distribution 2 is shown as a yellow line. **Right**: Dependence of the mean (blue) and median (red) of the tunneling time distribution on the linear size L of the square Ising model. Dashed lines are fittings (see text).

some kind of random walls, and time would be unlimited [15]. This fact may be a likely explanation for the unsuccessful applications of the WL algorithm to a system with a complex energy landscape.

The time it takes to reach flatness of the histogram is related to the tunneling time of the walker [13, 9]. The distribution of these times has long tails, leading to a spontaneous huge increase in the time of a particular simulation implementation. This fact may be the next likely explanation for failed applications - it is known that sometimes the flatness criteria cannot be met and the simulation cannot be completed.

We measure the tunneling time probability distribution for the 2D Ising model by simulating random walks in energy space using the exact DoS $g(E)$ from Ref. [16] in the Wang-Landau probability acceptance of new energy [1]

$$P_{WL} = \min\left[1, g(E_{old})/g(E_{new})\right]. \tag{1}$$

The resulting tunneling time probability distribution is shown in the left panel of Fig. 1 for the Ising model on a square lattice of linear size $L = 16$. The distribution looks like a lognormal distribution, in fact it matches the lognormal distribution

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2}\right) \tag{2}$$

very well, as shown in Fig. 1 by the solid line with values $\sigma \approx 0.83$ and $\mu \approx 13.0$.

The log-normality of distribution means that there are huge fluctuations in the tunneling time, which leads to huge fluctuations in the accumulation of sufficiently flat histogram during the application of the WL algorithm. This fluctuations increase with the size of the system, since the median $\exp(\mu)$ grows with the size of the system as $L^{2d+0.85(2)}$, and the mean $\exp(\mu + \sigma^2/2)$ grows with system size as $L^{2d+0.87(2)}$ as shown by dashed lines in the right panel of Fig. 1.

This fact of increasing tunneling time can be tried to be used in practice, using the fact that the tunneling time becomes shorter with fewer energy levels. It can be assumed that the division of the entire energy window into a sufficiently small part, the simulation of random walkers at smaller intervals corresponding to the size of the window, will lead to shorter tunneling times.

3.2. Mixing time

The mixing time t_{mix} is characteristic of a Markov process. It is defined as the reciprocal of the *spectral gap* G of the stochastic matrix associated with the Markov chain,

$$t_{mix} \propto \frac{1}{G} = \frac{1}{\lambda_1 - \lambda_2}, \tag{3}$$

where λ_1 and λ_2 are the two largest eigenvalues of the stochastic matrix, and $\lambda_2 < \lambda_1 = 1$.

In our case, it is characteristic for the final stage of simulation when the transition matrix in energy space TMES is sufficiently close to the stochastic, and $g(E)$ is sufficiently close to the desired DoS, which scaled with the linear size L for two dimensional Ising model [17] as $t_{mix} \propto L^{2d+0.28(4)}$. The mixing time grows more slowly than the tunneling time, and the tunneling time can be used as a safe characteristic of the convergence process. Since the tunneling time is related to the histogram flattening process, one can expect good convergence properties of the parallel WL algorithm.

4. Toy parallel Wang-Landau algorithm

There are two possible gains from the parallel Wang-Landau algorithm. The first advantage is related to the ability to reduce the computation time by increasing the number of energy windows when simulating parallel walkers. The second advantage is related to the possible reduction in tunneling time in each window due to the shorter walking distance.

We simplify the replica exchange Wang-Landau (reWL) algorithm [6, 7] to test these two possibilities directly and introduce a toy parallel Wang-Landau (tpWL) algorithm. We argue that the main results can be valid for any variants of parallel algorithms that use the Wang-Landau probability for a random walk in the energy spectrum. The main differences between the tpWL algorithm and the parallel reWL algorithm are: 1) An independent single random walk is performed inside each window; 2) An additional element has been introduced: the transition matrix $T(E_K, E_m)$, calculated in each window. 3) No replica exchange; 4) the Wang-Landau 1/t-algorithm [18] is used; 5) After N-steps, the global DoS is calculated using the expression

$$g(E_i) = \frac{1}{l} \sum_{j=0}^{l} g^j(E_i), \tag{4}$$

where l is the number of windows containing the energy level E_i, and $g_j(E_i)$ is the value of the density of states at the level E_i, accumulated in window j.

The formal description of the toy parallel Wang-Landau (tpWL) algorithm is formulated as follows.

In the Algorithm 1 *Wang-Landau step* is the original Wang-Landau step [1, 2] supplemented by calculation of transition matrices $T_i(E_k, E_m)$ in each window $i = 0, ..., N_w - 1$ and by $1/t$-WL algorithm [18] at the second stage of simulation.

5. Tunneling times in windows

In this section, we analyze tunneling times in windows by simulating the Ising model on a square lattice with linear size $L = 16$ and splitting the energy spectrum into 12 windows with a window overlap of 75%, as recommended in [6, 7]. The partition is shown in the left panel of fig. 2 along with the exact DoS $g(E)$ [16]. We use the same partition size in test simulations and initialize all walkers in the ground state, so all walkers start at the left end of the energy spectrum.

It is constructive to take a closer look at the parallel windowed simulations: there are *two extreme stages* of simulation.

The first stage is the initial part of the simulation, it starts with flat $g(E)$ and a random position of the walker in the windows. The walker's initial position can be at any energy level, and for a typical energy window, one must wait until the walker enters a particular window. This can happens on the left side of the window or on the right side of the window. Therefore, we estimate the tunneling time starting from the left edge of the window and measure the time it takes for the vagrant to reach the right edge of the window by updating the current $g(E)$ according to the Wang-Landau algorithm. This will be the tunneling time from left to right, t^{in}_{LR}. The superscript *in* means *initial stage tunneling time*. We estimate the tunneling time

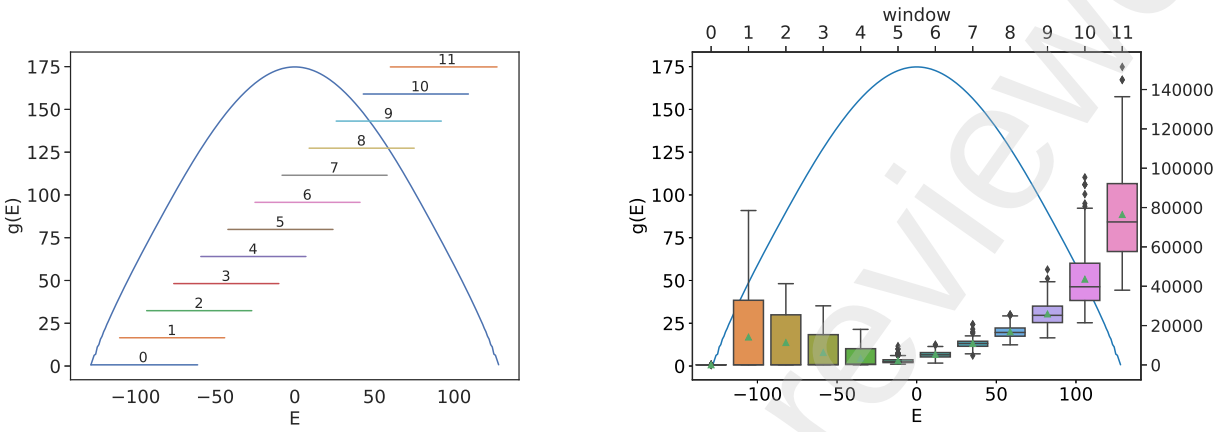

Figure 2: The solid blue line is $\log g(E)$, where $g(E)$ is the exact density of states of the two-dimensional Ising model on a square lattice, linear size $L = 16$. Number of local spin updates between histogram check is 10^6. Number of runs is 240. The walkers initialized at the ground state – at the left edge of the spectrum. **Left**: An example of partition the energy spectrum into 12 intersecting windows. **Right**: Tunneling times t_{tun} in energy windows with the right scale. Mean values marked with triangles and median values are marked by the horizontal internal line. The bottom edge of the rectangle is the first quartile Q1 of distribution and upper edge is the third quartile Q3. The outgoing bars outside the box represent the boundaries of the data scatter within the interquartile range, defined as the difference between Q3-Q1.

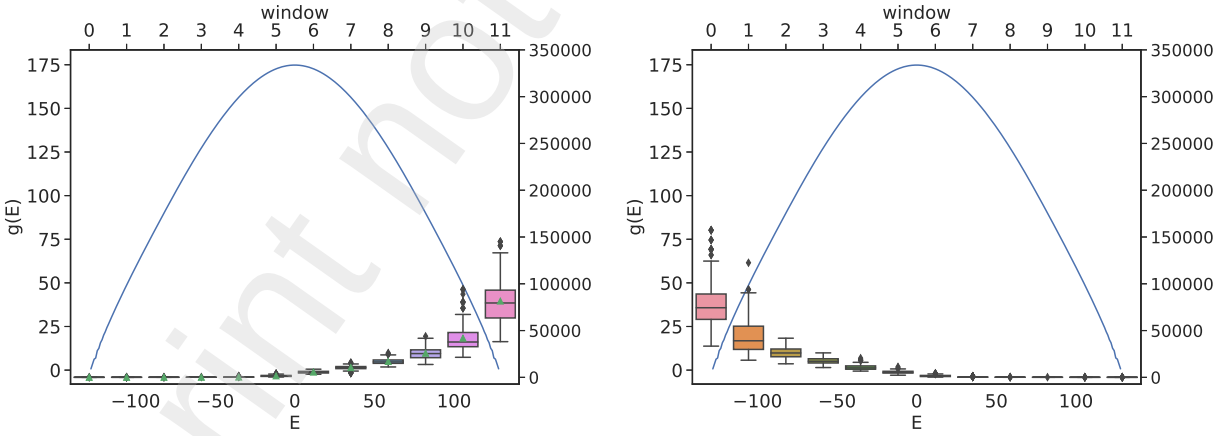

Figure 3: The solid blue line is $\log g(E)$, where $g(E)$ is the exact density of states of the two-dimensional Ising model on a square lattice, linear size $L = 16$. Number of runs is 240 with the flat initial histogram $g(E)$ for each run. **Left**: Tunneling times t^{in}_{LR} in energy windows. **Right**: Tunneling times t^{in}_{RL} in energy windows. Mean values marked with triangles and median values are marked by the horizontal internal line. The bottom edge of the rectangle is the first quartile Q1 of distribution and upper edge is the third quartile Q3. The outgoing bars outside the box represent the boundaries of the data scatter within the interquartile range, defined as the difference between Q3-Q1.

Algorithm 1 – Toy parallel Wang-Landau (tpWL) algorithm with Transition matrix

Parallel segment:
while $F > f_{final}$ **do**

i=0
repeat
Wang-Landau step
$i \leftarrow i + 1$
until $i < M$

if $H(E)$ - "flatness" in all N_w windows **then**
Calculate the largest eigenvalue λ_1 of the transition matrices $T_i(E_k, E_m)$
if $(F > N_E/t)$ **then** ▷ update parameter F
$F \leftarrow F/2$
else $F \leftarrow N_E/t$
end if
$H(E) \leftarrow 0$
$\overline{g}(E_i) \leftarrow \frac{1}{l}\sum_{j=0}^{l-1} g_j(E_i)$, where l is the number of windows with energy E_i
$g_h(E_i) \leftarrow \overline{g}(E_i)$
$T(E_k, E_m) = \frac{1}{l}\sum_{i=0}^{l-1} T_i(E_k, E_m)$ ▷ Calculate global transition matrix
end if
end while
End of parallel segment

Calculate global DoS: $g(E_i) \leftarrow \frac{1}{l}\sum_{j=0}^{l-1} g_l(E_i)$, where l is the number of windows with energy E_i

starting from the right edge of the window and measure the time it takes for the vagrant to reach the left edge of the window by updating the current $g(E)$ according to the Wang-Landau algorithm. This will be the tunneling time from right to left, t^{in}_{RL}. We repeat the simulation 240 times and measure the averages of t^{in}_{LR} and t^{in}_{RL} and analyze some properties of their distributions presented in the table 1. The distributions are almost symmetrically reflected.

The second stage is the final part of the simulation, with DoS close to the desired one. We estimate the typical tunneling time with the walker simulated with the exact DoS and without any updates of $g(E)$. The rest of parameters are the same as used for the estimation of t^{in}_{LR} and t^{in}_{RL} and results are shown in fig 4. The averages of t^{fin}_{LR} and t^{fin}_{RL} together with standard deviation and median are presented in the table 2. Reflective symmetry still exists in the final stage. In this case, the time distribution is also symmetrical to the maximum DoS penalty. This is due to the special DoS symmetry for the Ising model. Indeed, DoS is not symmetric for Potts models with more than two components and the tunneling time in the windows will be asymmetric. In addition, the figures 3 and 4 give an idea why the tunneling time of a random walker with the Wang-Landau probability is larger than that of a simple symmetric one-dimensional random walk.

Indeed, figures 5 compares the accuracy of sequential simulation equivalent to one windows simulation with parallel simulation in twelve windows. The typical realization is shown. The relative deviation of DoS Δ changes almost in the same way at the final stage of simulation close to the exact DoS. It is noteworthy that Δ fluctuations are smaller in parallel simulations, which can possibly be explained by smaller tunneling time fluctuations in shorter windows. It should be noted that the frequency of flatness checks in the first step of the simulation algorithm reported so far is 10^6, which is greater than the mean and median tunneling times in windows. A shorter frequency interval for $L = 16$ will result in less accurate simulations.

	from Left to Right			from Right to Left		
window	t^{in}_{LR}	STD	$Median$	t^{in}_{RL}	STD	$Median$
0	48	5	47	79498	24891	74399
1	60	10	58	44533	17614	39082
2	79	14	77	26382	5936	26049
3	125	30	120	17487	3331	17217
4	298	177	249	10601	2596	10111
5	1571	857	1316	5583	1419	5451
6	5615	1267	5469	1518	871	1402
7	10438	2004	10385	310	216	239
8	16974	2942	16476	118	25	115
9	25762	6268	25407	79	15	77
10	41989	14580	37776	60	10	60
11	81558	24177	79489	49	8	48

Table 1: Average tunneling times t^{in}_{LR} and t^{in}_{RL}, their standart deviations STD and $Median$ for the simulation with the flat initial $g(E) = 0$.

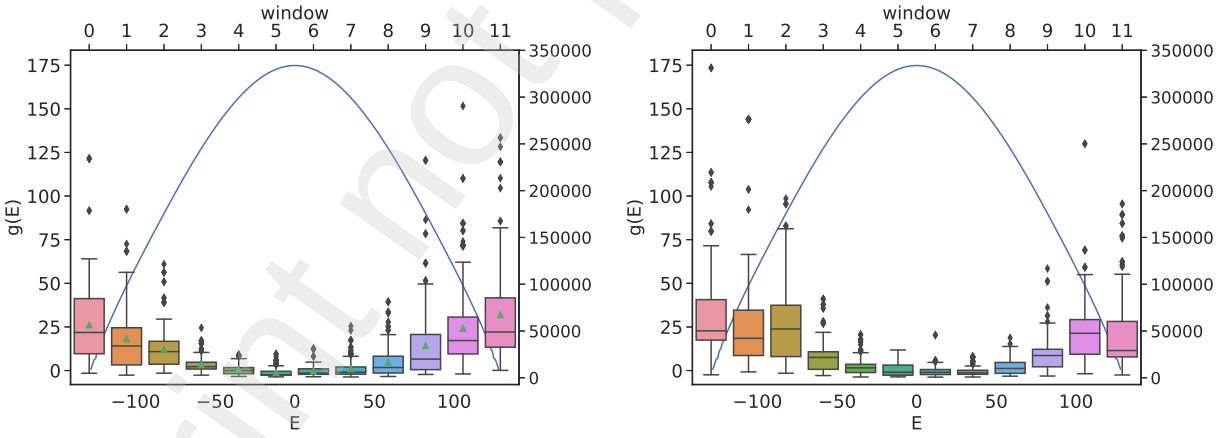

Figure 4: The solid blue line is $\log g(E)$, where $g(E)$ is the exact density of states of the two-dimensional Ising model on a square lattice, linear size $L = 16$. Number of runs is 240 with exact DoS. **Left**: Tunneling times t^{fin}_{LR} in energy windows. **Right**: Tunneling times t^{fin}_{RL} in energy windows. Mean values marked with triangles and median values are marked by the horizontal internal line. The bottom edge of the rectangle is the first quartile Q1 of distribution and upper edge is the third quartile Q3. The outgoing bars outside the box represent the boundaries of the data scatter within the interquartile range, defined as the difference between Q3-Q1.

	from Left to Right			from Right to Left		
window	t_{LR}^{fin}	STD	$Median$	t_{RL}^{fin}	STD	$Median$
0	56633	42428	48419	67291	54025	50178
1	41708	34767	33984	68969	74175	42146
2	30664	21417	27973	54986	40153	52141
3	13971	8323	11858	21779	16512	21709
4	8932	5989	7676	12332	9256	10609
5	5663	4907	3167	8735	6723	5877
6	7119	5392	4968	7097	5609	5808
7	9737	9258	6040	6842	5097	5275
8	16638	15551	10983	11671	7717	9981
9	34294	38112	20000	24513	16937	23797
10	52976	44701	39800	51734	36773	47565
11	67397	61348	48948	48095	42765	29110

Table 2: Average tunneling times t_{LR}^{fin} and t_{RL}^{fin}, their standard deviations STD and $Median$ for the simulation with the exact DoS.

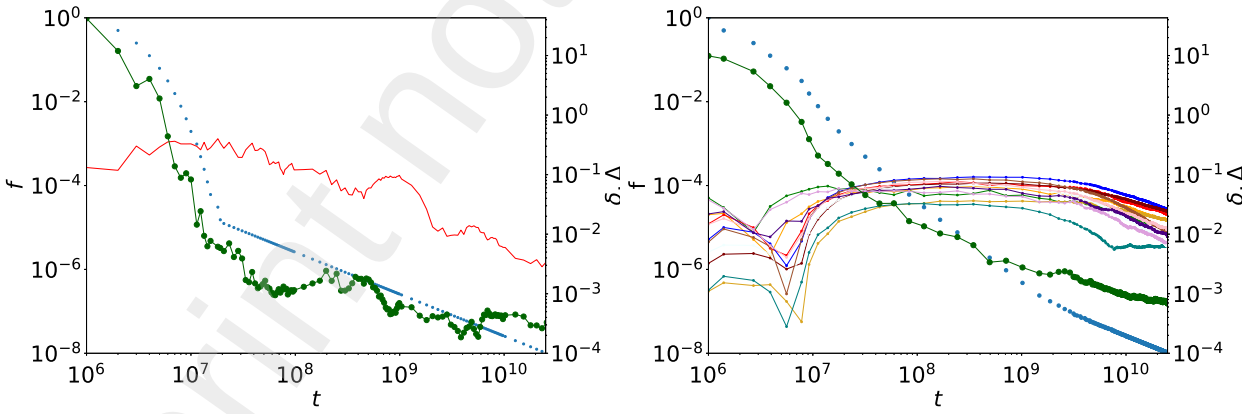

Figure 5: Simulation of a two-dimensional Ising model on a square lattice, linear size $L = 16$. The blue dotted line is the f parameter and the dotted green line is the average DoS deviation Δ. **Left**: Sequential simulation of one window. The red line is the δ accuracy parameter. **Right**: Parallel simulation with 12 windows schematically shown in fig. 2 and 12 colored lines corresponds to the δ accuracy parameters in the 12 windows. Window overlap 75%.

6. Simulation results

When simulating, we start from the ground state of the samples. The random number generator MT19937 [21, 22] was used to generate two random numbers as the trial spin lattice coordinates and to generate one random number when deciding to flip the trial spin with Wang-Landau probability.

The right panel of fig. 2 shows the result of the estimated tunneling time t_{tun} and details of the t_{tun} distribution. The uneven distribution in the windows is visible: a relatively small spread of tunneling times in the windows with numbers 0 and $5-8$ and a large spread of data in the windows $1-3$ and $10-11$.

The largest mean and median values are in the far right window and are around 75,000. These values should be compared with the mean and median values of the entire spectrum, which are approximately 600,000 and 450,000 and are 6-8 times larger. Neglecting a significant spread of tunneling times, we can state the acceleration of real-time simulation, provided that the accuracy of parallel simulation is comparable to the accuracy of sequential simulation in one window with the entire energy spectrum.

Indeed, figures 5 compares the accuracy of sequential simulation equivalent to one windows simulation with parallel simulation in twelve windows – the typical realization is shown. The relative deviation of computed DoS $g(E_i)_{comp}$ from the exact one $g(E_i)_{exact}$

$$\Delta = \frac{1}{N_E}\sum_{1}^{N_E}\left(1 - \frac{g(E_i)_{comp}}{g(E_i)_{exact}}\right) \tag{5}$$

changes almost in the same way at the final stage of simulation close to the exact DoS. It is noteworthy that Δ fluctuations are smaller in parallel simulations, which can possibly be explained by smaller tunneling time fluctuations in shorter windows. It should be noted that the frequency of flatness checks in the first step of the simulation algorithm reported so far is 10^6, which is greater than the mean and median tunneling times in windows. A shorter frequency interval for $L = 16$ will result in less accurate simulations.

With the number of energy levels N_E, the number of energy windows k, and the coefficient of the intersection length of two windows p_{cross}, the number of energy levels in each window N_{Ew} is estimated as

$$N_{Ew} = \frac{N_E}{1 + (1 - p_{cross})(k-1)}. \tag{6}$$

The slowest tunneling time scale (corresponding, for example, to the far right window in fig. 2 is scaled as $N_{Ew}^{d+z/2}$. The number of WL steps in the energy spectrum required to achieve flatness of the histogram in each window grows in proportion to the tunneling time. We can calculate the acceleration of parallel simulations as the ratio of the tunneling time for the entire spectrum to the tunneling time in the slowest window, and the time acceleration factor is

$$q_{accel} \approx \left(\frac{N_E}{N_{Ew}}\right)^{d+z/2} = [1 + (1 - p_{cross})(k-1)]^{d+z/2} \tag{7}$$

due to parallel simulation of k windows. For example, for a 2D Ising model with 12 windows and $p_{cross} = 3/4$ which is shown as the example in Fig. 2, the speedup can reach 20.

7. Discussion

We introduce a simple algorithm for parallel Wang-Landau simulation. It is based on the idea of using random walks in windows with 75% intersection proposed in [6, 7]. We use one pass in each window without replicas and combine the global DoS with a simple average, Ref. (4). Despite this simplification, we do not find any systematic deviations of the specific heat calculations, which bothered the authors of Refs. [6, 7]. Figure 6 shows the specific heat calculated with one paralle DoS simulations are no worse than in serial DoS simulations.

We would like to emphasize the log-normal behavior of the tunneling times and their large scatter, which can lead to sporadic deviations in the DoS estimate and long times to flatten the histogram. The conclusion

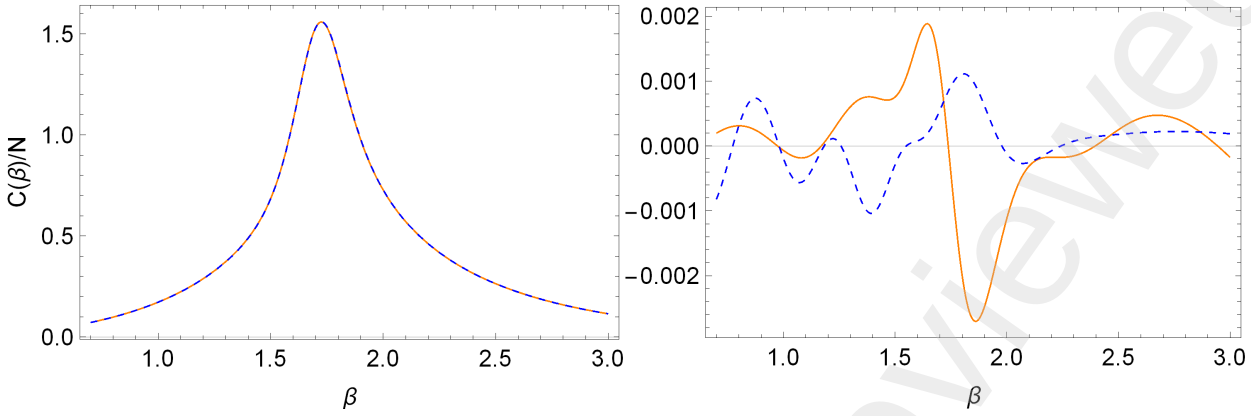

Figure 6: Specific heat of a two-dimensional Ising model on a square lattice, linear size $L = 16$. The blue dotted line corresponds to parallel simualtion and red solid to the sequenttial simulation. **Left**: Specific heat C dependence from inverse temperature β. **Right**: Relative deviation $dev = C(\beta)/C_{ex}(\beta) - 1$ of calculated specific heat C from the exact C_{ex}.

of this article may explain the simulation difficulties experienced by the community using the Wang-Landau algorithm. The calculation of TMES and the analysis of the control parameter $\delta = |1 - \lambda_1|$ will reveal sporadic deviations and can help correct and possibly solve the problem.

8. Acknowledgments

We used the supercomputing facility of the HSE University.
Support from the Basic Research Program of the HSE University is gratefully acknowledged.

[1] F. Wang and D.P. Landau, *Efficient, multiple-range random walk algorithm to calculate the density of states*, Phys. Rev. Lett. **86**, 2050 (2001).
[2] F. Wang and D.P. Landau, *Determining the density of states for classical statistical models: A random walk algorithm to produce a flat histogram*, Phys. Rev. E **64**, 056101 (2001).
[3] D.P. Landau and R. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics* (Cambridge University Press, Cambridge, 2015).
[4] M.O. Khan, G. Kennedy, and D.Y.C. Chan, *A scalable parallel Monte Carlo method for free energy simulations of molecular systems*, J. Comp. Chem. **26**, 72 (2005).
[5] L. Zhan, *A parallel implementation of the Wang–Landau algorithm*, Comp. Phys. Comm. **179**, 339 (2008).
[6] T. Vogel, Y.W. Li, T. Wüst, and D.P. Landau, *Generic, hierarchical framework for massively parallel Wang-Landau sampling*, Phys. Rev. Lett. **110**, 210603 (2013).
[7] T. Vogel, Y.W. Li, T. Wüst, and D.P. Landau, *Scalable replica-exchange framework for Wang-Landau sampling*, Phys. Rev. E **90**, 023302 (2014).
[8] A.C.K. Farris and D.P. Landau, *Replica exchange Wang–Landau sampling of long HP model sequences*, Physica A **569**, 125778 (2021).
[9] L.N. Shchur 2019, *On properties of the Wang–Landau algorithm*, J. Phys.: Conf. Ser. **1252** 012010 (2019).
[10] L.Yu. Barash, M.A. Fadeeva, and L.N. Shchur, *Control of accuracy in the Wang-Landau algorithm*, Phys. Rev E **96**, 043307 (2017).
[11] S. Redner, *A First Look at First-Passage Processes*, arXiv: 2201.10048.
[12] K. Hukushima and K. Nemoto, *Exchange Monte Carlo method and application to spin glass simulations*, J. Phys. Soc. Jpn. **65**, 1604 (1996).
[13] P. Dayal, et al, *Perfomance limitations of flat-histogram method*, Phys. Rev. Lett. **92**, 097201 (2004).
[14] Book on probability
[15] Ya.G. Sinai, *The Limiting Behavior of a One-Dimensional Random Walk in a Random Medium* , Theory Probab. Appl. **27**, 256 (1982)
[16] P.D. Beale, *Exact distribution of energies in the two-dimensional Ising model*, Phys. Rev. Lett. **76**, 78 (1996).
[17] M. Fadeeva and L.N. Shchur, *On the mixing time in the Wang-Landau algorithm*, J. Phys.: Conf. Ser. **955** (2018) 012028.
[18] R.E. Belardinelli and V.D. Pereyra, *Fast algorithm to calculate density of states*, Phys. Rev. E **75** (2007) 046701.
[19] J.-S. Wang, T.K. Tay, and R.H. Swendsen, *Transition Matrix Monte Carlo Reweighting and Dynamics*, Phys. Rev. Lett. **82**, 476 (1999).
[20] J.-S. Wang and R.H. Swendsen, *Transition matrix Monte Carlo method*, J. Stat. Phys. **106**, 245 (2002).

[21] M. Matsumoto and T. Tishimura, *Mersenne Twister: A 623- dimensionally equidistributed uniform pseudorandom number generato*, ACM Trans. on Mod. and Comp. Simul. **8**, 3 (1998).
[22] M.S. Guskova, L.Yu. Barash, and L.N. Shchur, *RNGAVXLIB: Program library for random number generation, AVX realization*, Comp. Phys. Comm., **200** (2016) 402.