MUCA VS WL: A COMPARISON

Elmar Bittner, University Heidelberg and
Wolfhard Janke, University Leipzig

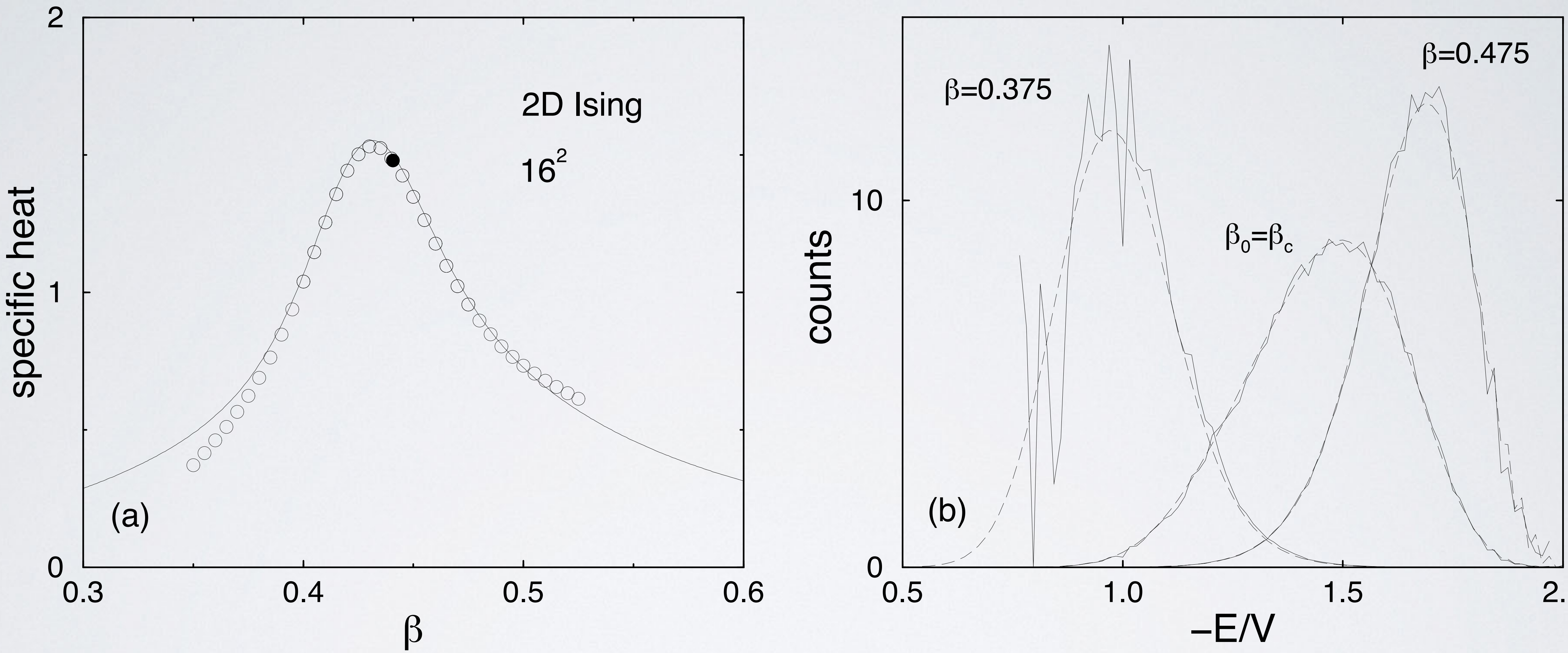
UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

A TIGHT RACE

Elmar Bittner, University Heidelberg and
Wolfhard Janke, University Leipzig

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Motivation I



Motivation 1

$$Z(\beta_0) = \sum_{\{s\}} e^{-\beta_0 H(\{s\})} = \sum_{\{s\}} \Omega(E) e^{-\beta_0 E} \propto \sum_E P_{\beta_0}(E)$$

$$P_{\beta_0}(E) \propto \Omega(E) e^{-\beta_0 E}$$

$$P_\beta(E) \propto \Omega(E) e^{-\beta E} = \Omega(E) e^{-\beta_0 E} e^{-(\beta-\beta_0)E} \propto P_{\beta_0}(E) e^{-(\beta-\beta_0)E}$$

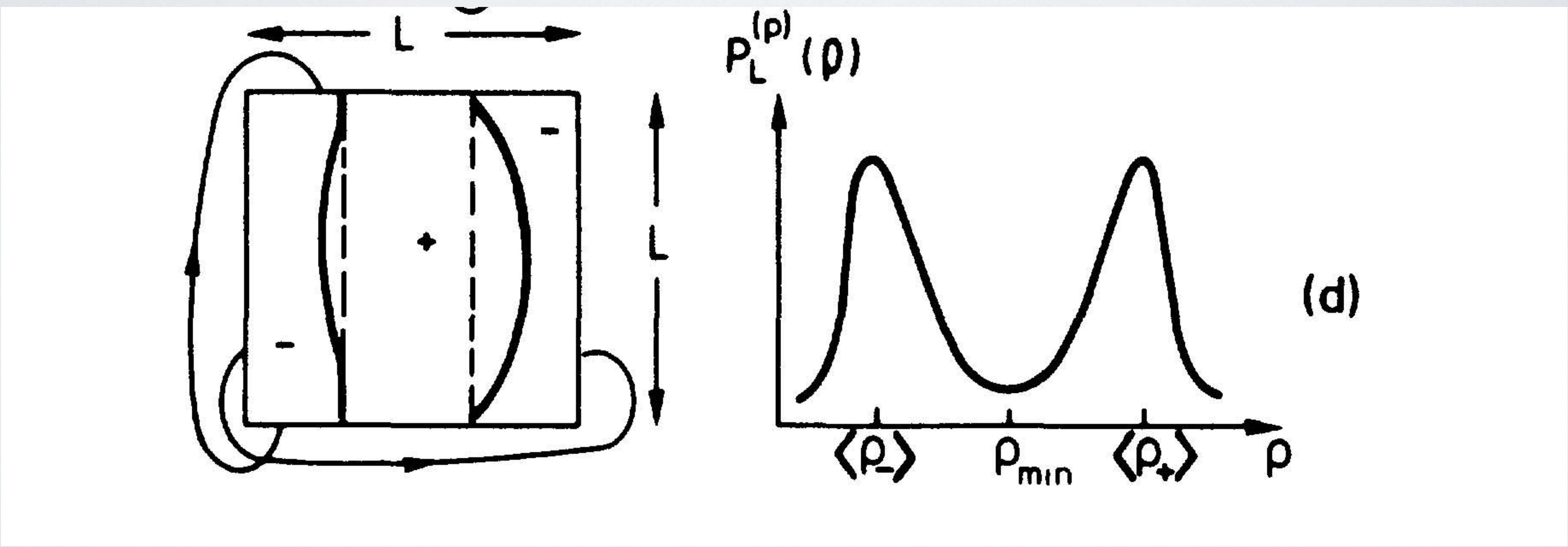
$$\langle f(E) \rangle(\beta) = \sum_E f(E) P_\beta(E) / \sum_E P_\beta(E)$$

Motivation 2

Binder's Method for Estimating Interface Tensions

Binder, Phys. Rev. A 25 (1982) 1699

Simulations with periodic boundary conditions



Motivation 2

Binder's Method for Estimating Interface Tensions

Binder, Phys. Rev. A 25 (1982) 1699

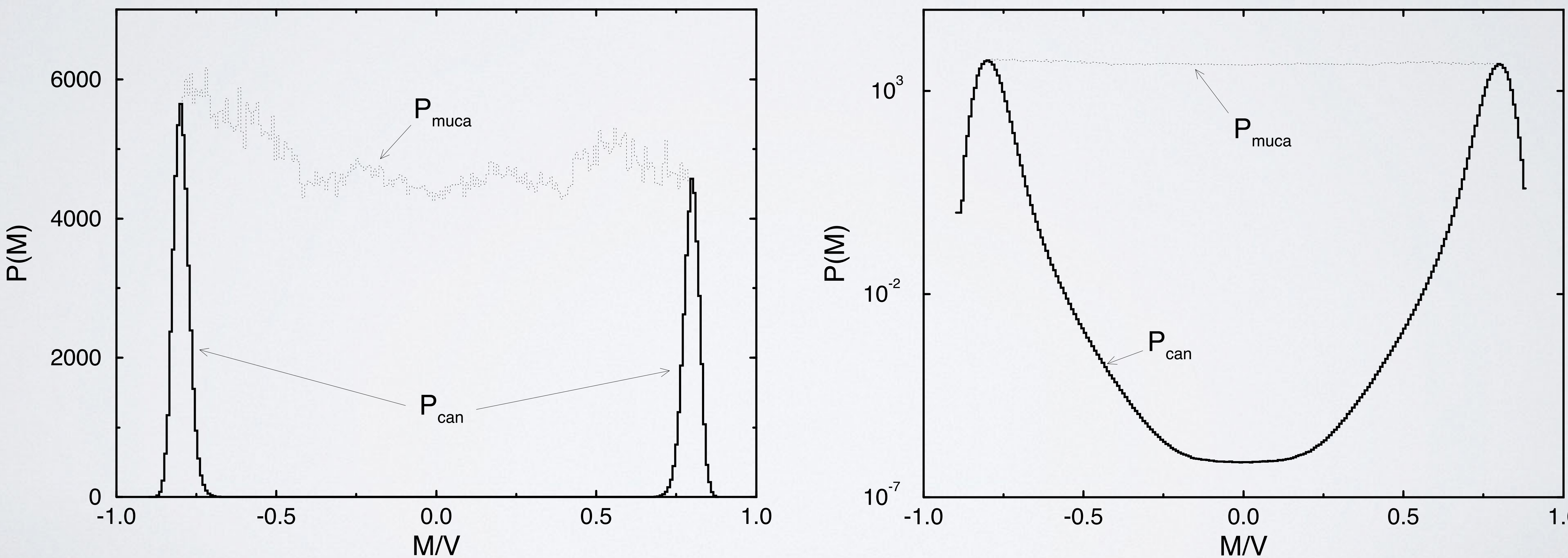
$$F_L^s = -\frac{1}{L^{(D-1)}} \ln\left(\frac{P_L^{\min}}{P_L^{\max}}\right)$$

First results with canonical simulations remained pitiful. Reason: $P_L^{\min}$ is exponentially suppressed in the canonical ensemble.

Motivation 2

$$P_{\mathrm{can}}(\phi) \propto e^{-\beta H(\phi)} \qquad \longrightarrow \qquad P_{\mathrm{muca}}(\phi) \propto e^{-\beta H(\phi) - f(Q_i(\phi))}$$

$$W(Q_i) \equiv e^{-f(Q_i(\phi))}$$



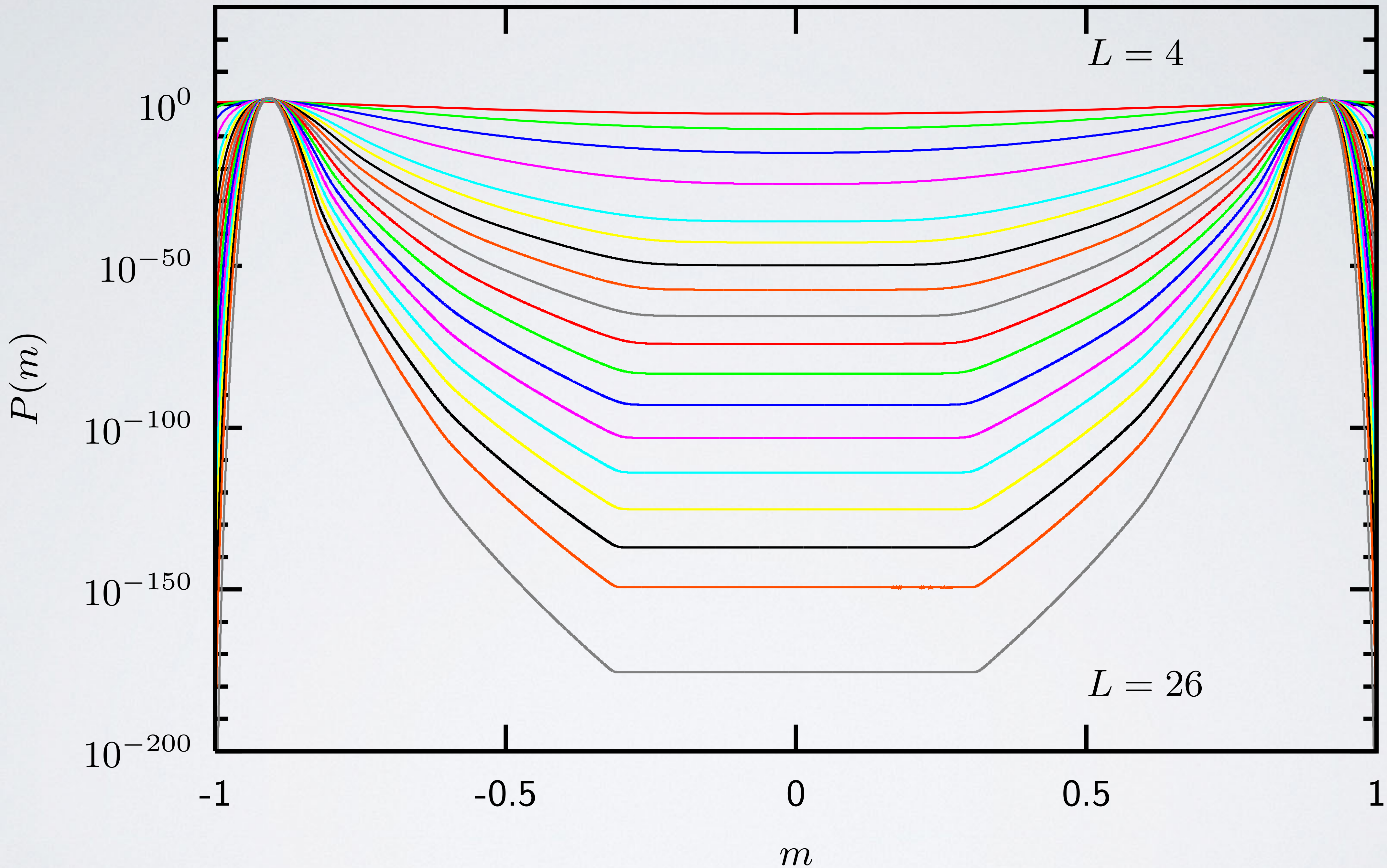
Motivation 2

$$P_{\mathrm{can}}(\phi) \propto e^{-\beta H(\phi)} \quad \longrightarrow \quad P_{\mathrm{muca}}(\phi) \propto e^{-\beta H(\phi) - f(Q_i(\phi))}$$

$$W(Q_i) \equiv e^{-f(Q_i(\phi))}$$

$$\langle \mathcal{O} \rangle_{\mathrm{can}} = \langle \mathcal{O} W^{-1}(\{Q_i\}) \rangle_{\mathrm{muca}} / \langle W^{-1}(\{Q_i\}) \rangle_{\mathrm{muca}}$$

Motivation 2



Distribution of the magnetization m for the 3D Ising model with periodic boundary conditions at $\beta = 0.3$.

Goal

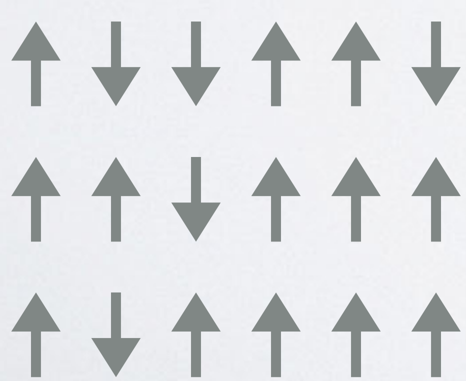
- a competitive analysis to study the relative performance of the two best-known generalized-ensemble algorithms:

  mulitcanonical Monte Carlo vs Wang-Landau method

$$Z = \sum_{\{s_i\}} e^{-\beta H} = \sum_E \Omega(E)\, e^{-\beta E}$$

- keep things as simple and clear as possible:

  take the exactly solvable 2D Ising model as test case



$$H = -\sum_{\langle ij \rangle} s_i s_j \quad s_i = \pm 1$$

Multicanonical Monte Carlo Algorithm

in the MuCa method one constructs auxiliary weights $W(E)$

$$P_{\mathrm{muca}}(E) = P_{\mathrm{can},\beta}(E) W(E)$$

to construct the weights we use an accumulative recursion

defining the weight ratio $R(E) = \dfrac{W(E+\Delta E)}{W(E)}$

Multicanonical Monte Carlo Algorithm

1. set histogram $H(E)$ to zero, perform m update sweeps with $R(E)$ and measure $H(E)$
2. compute for each bin the statistical weight of the current run $p(E) = H(E)H(E+\Delta E)/[H(E)+H(E+\Delta E)]$
3. Accumulate statistics

$$p_{n+1}(E) = p_n(E) + p(E)$$

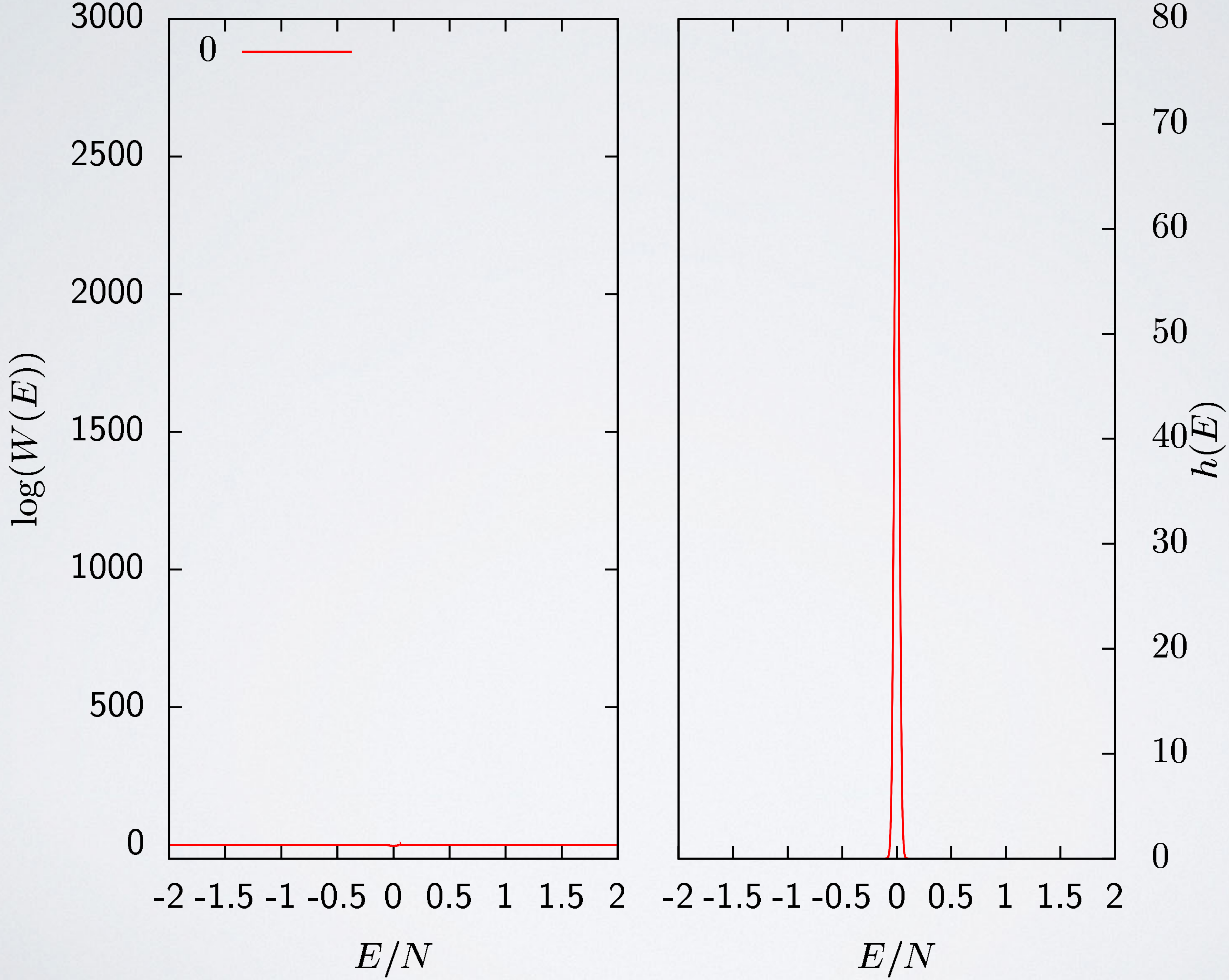
$$\kappa(E) = p(E)/p_{n+1}(E)$$

4. Update weight ratios

$$R_{\mathrm{new}}(E) = R(E)\,[H(E)/H(E+\Delta E)]^{\kappa(E)}$$

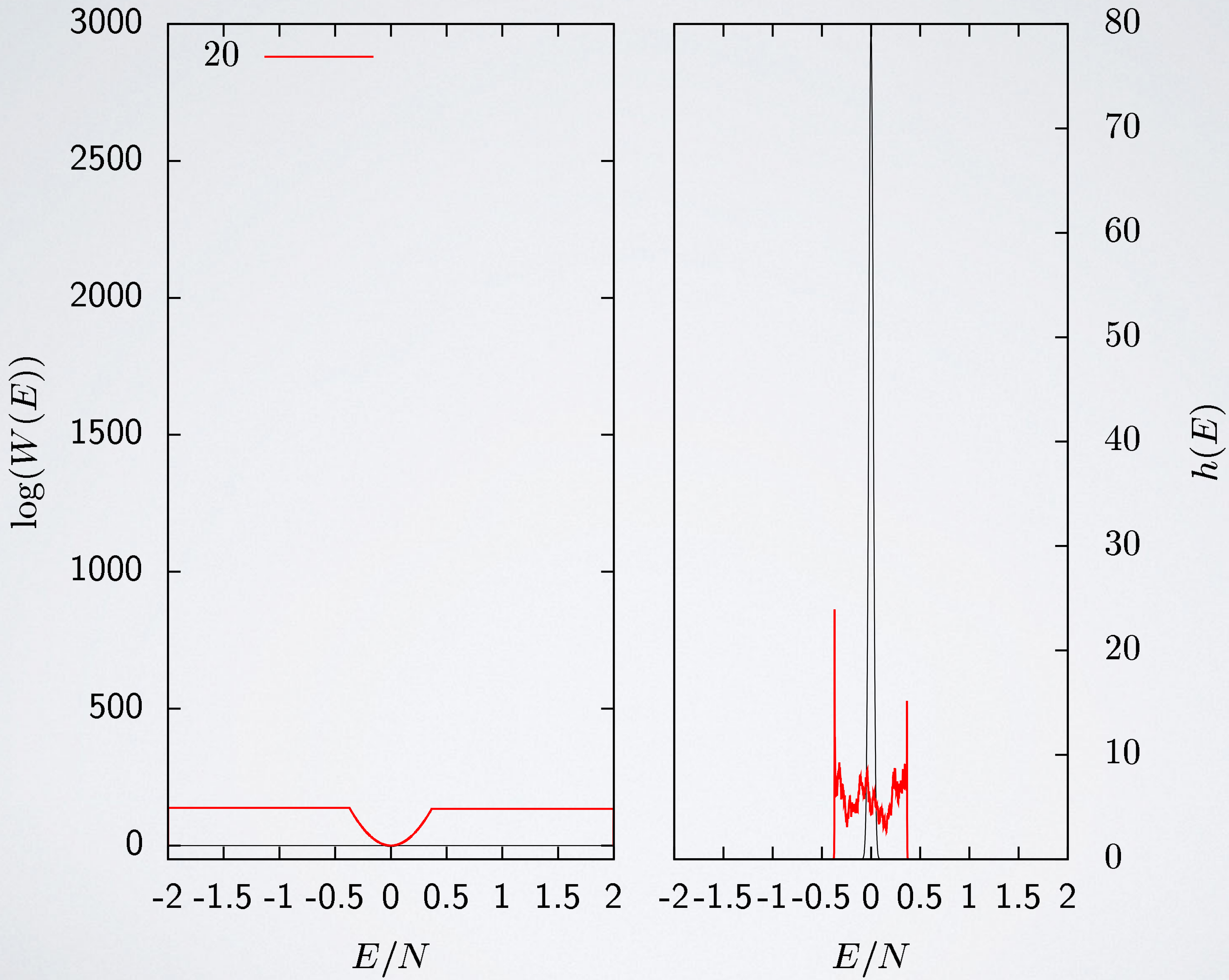
set $R(E) = R_{\mathrm{new}}(E)$ and go to 1

Multicanonical Monte Carlo Algorithm



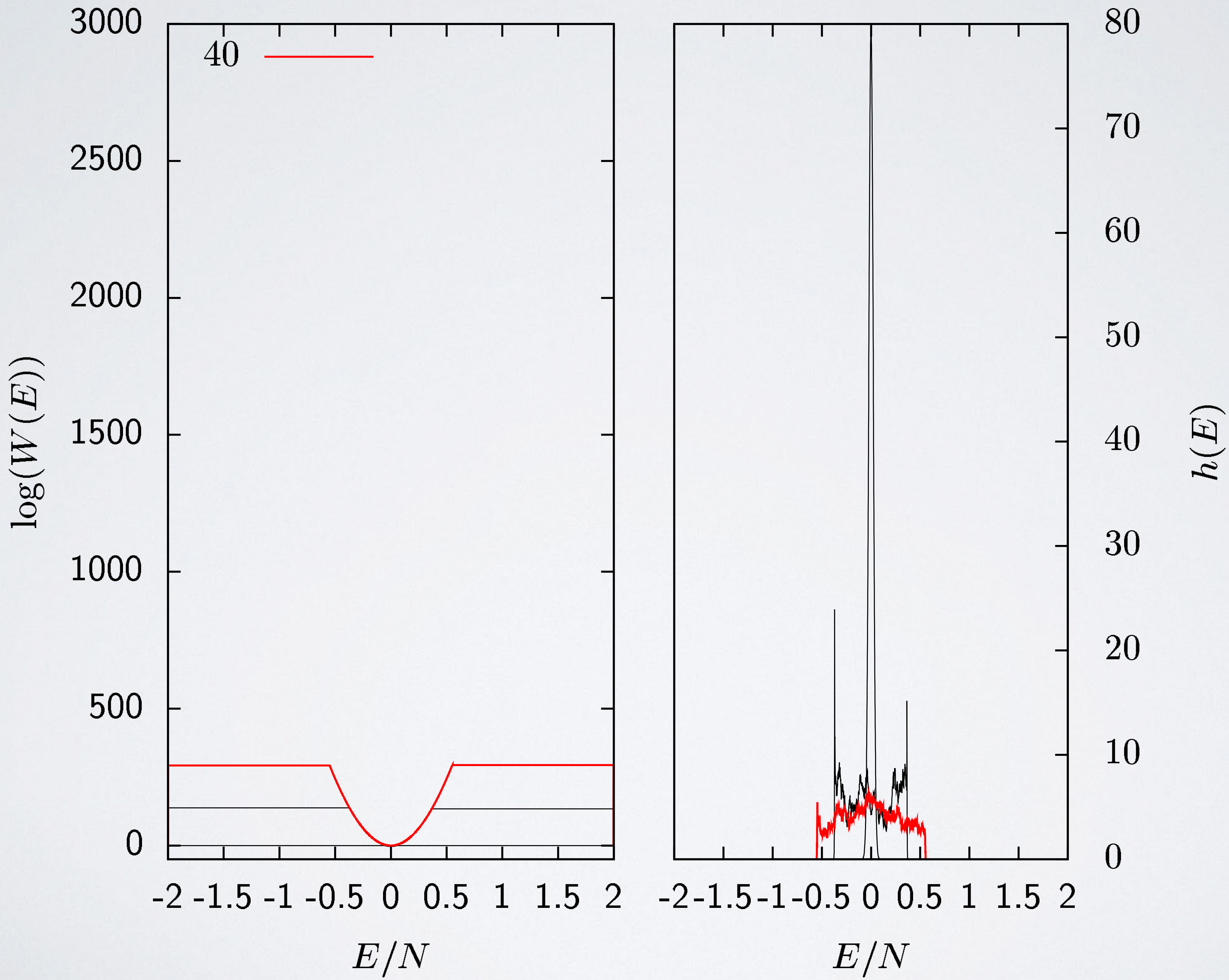
$L = 64$

Multicanonical Monte Carlo Algorithm



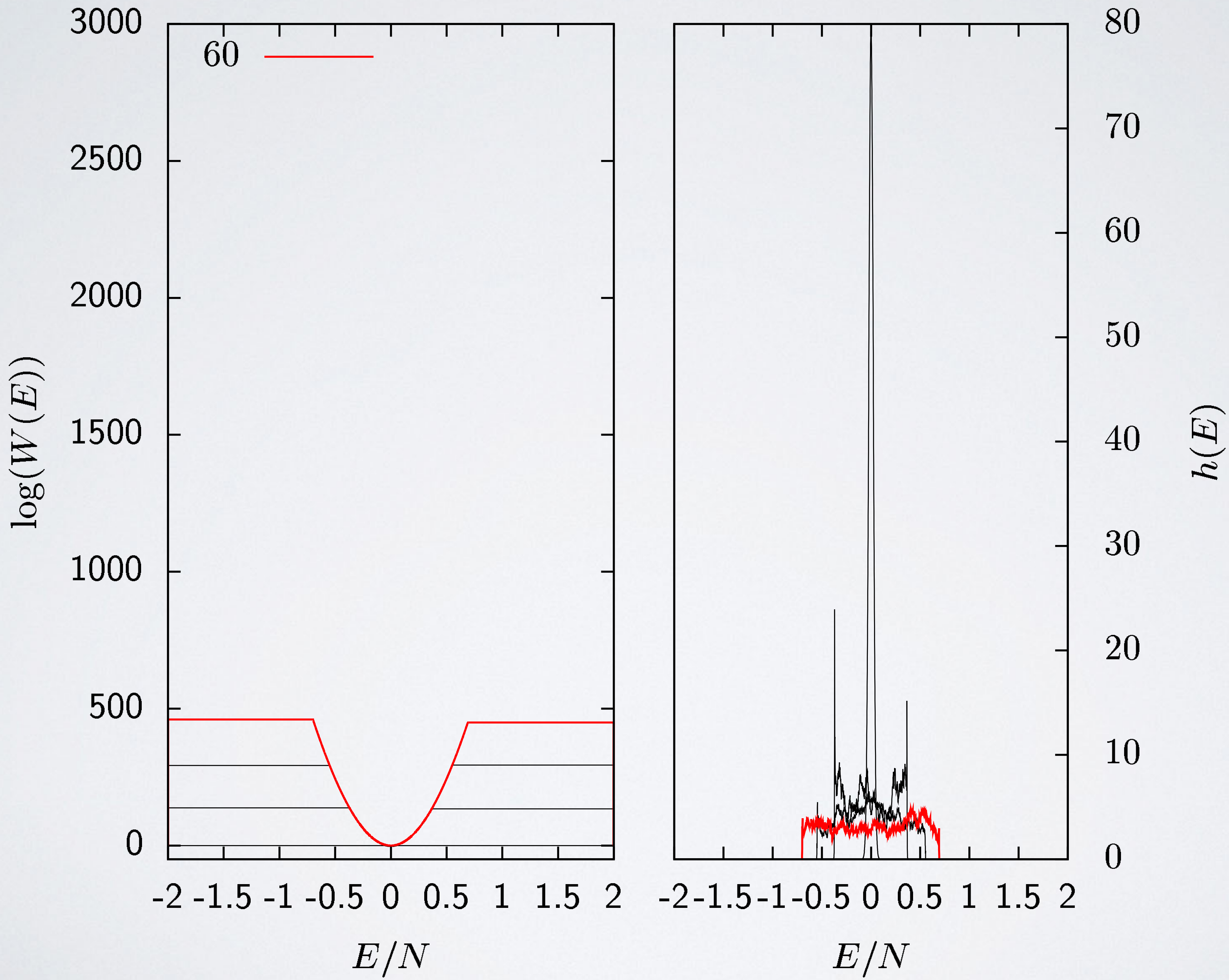
$L = 64$

Multicanonical Monte Carlo Algorithm



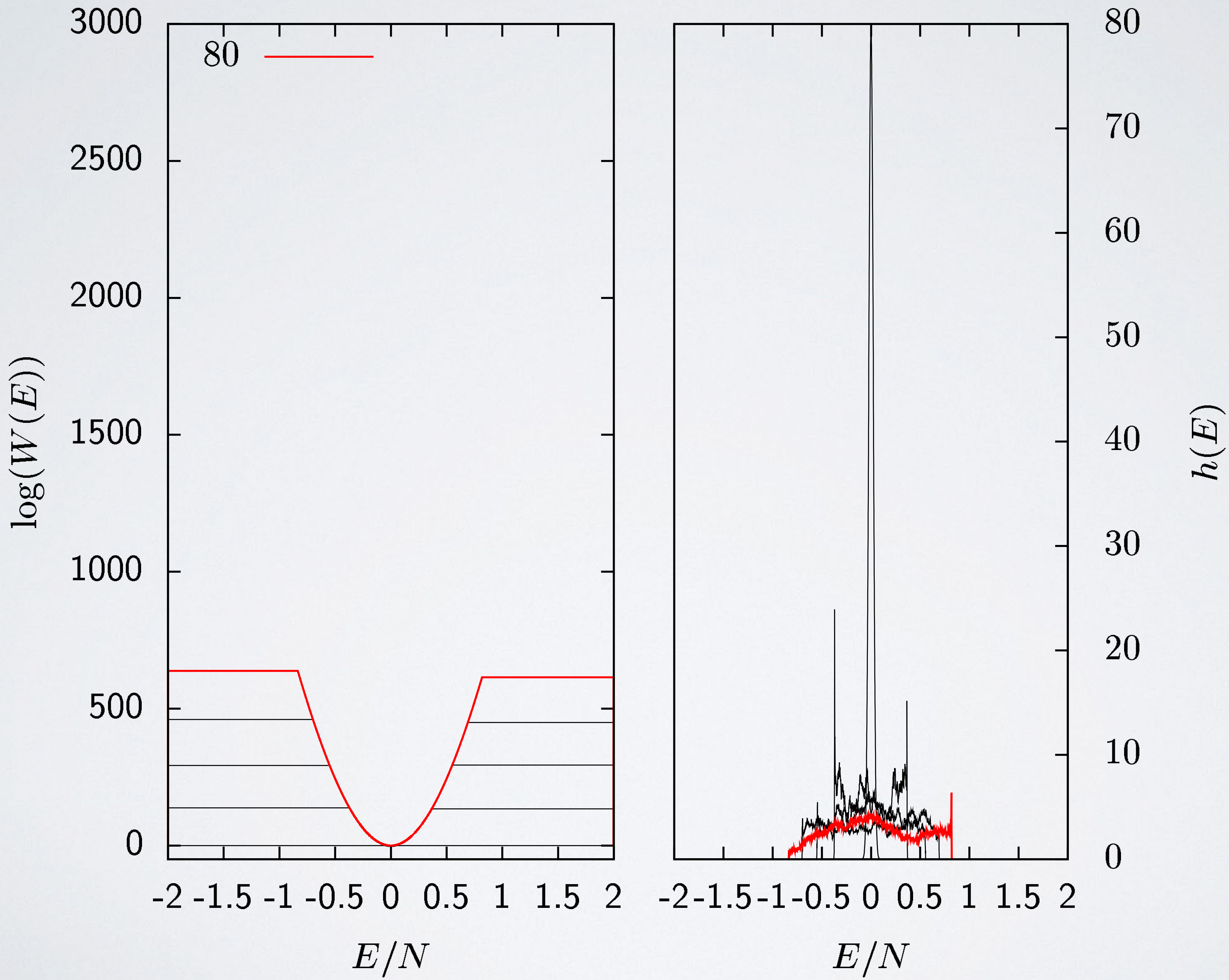
$L = 64$

Multicanonical Monte Carlo Algorithm



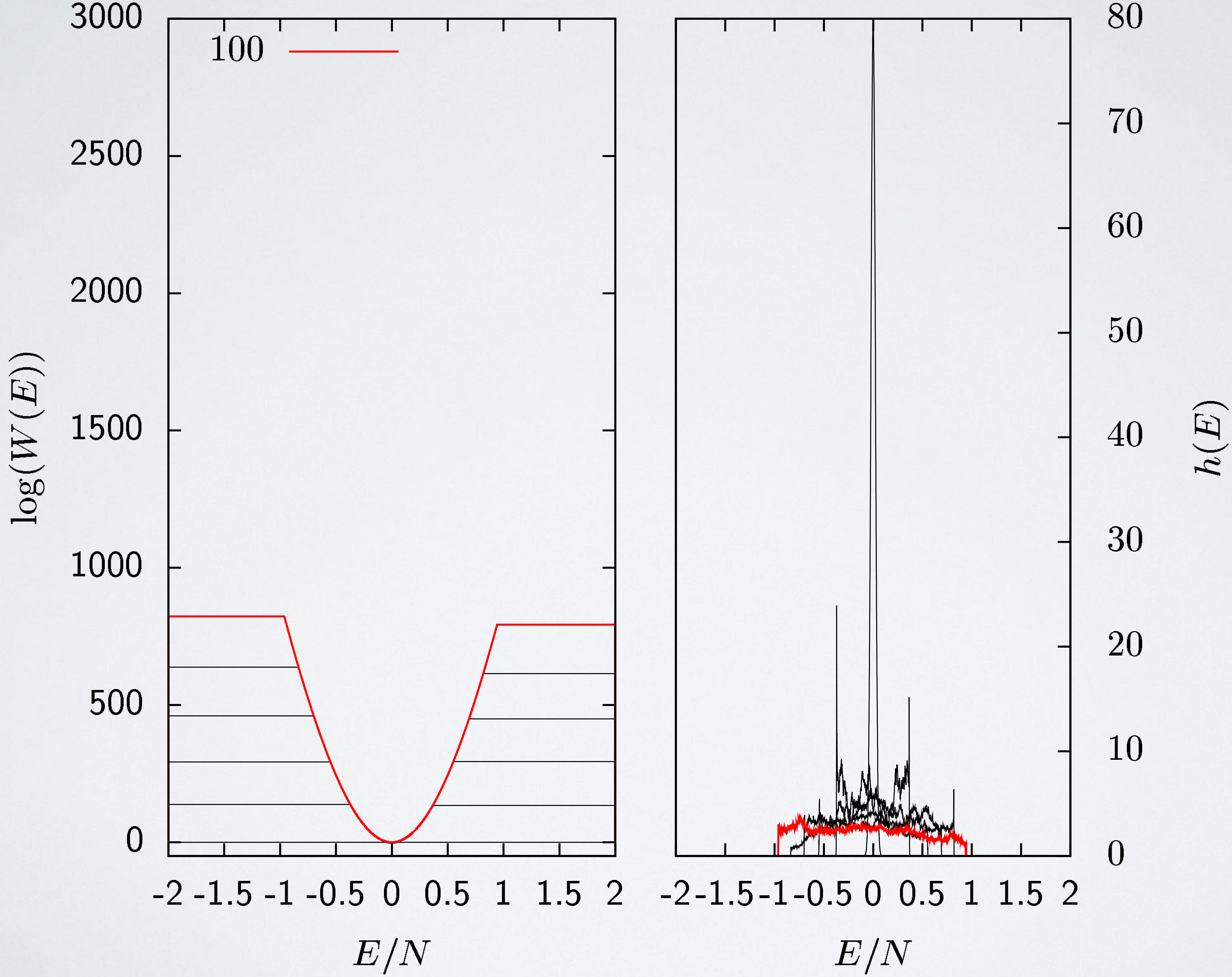
$L = 64$

Multicanonical Monte Carlo Algorithm



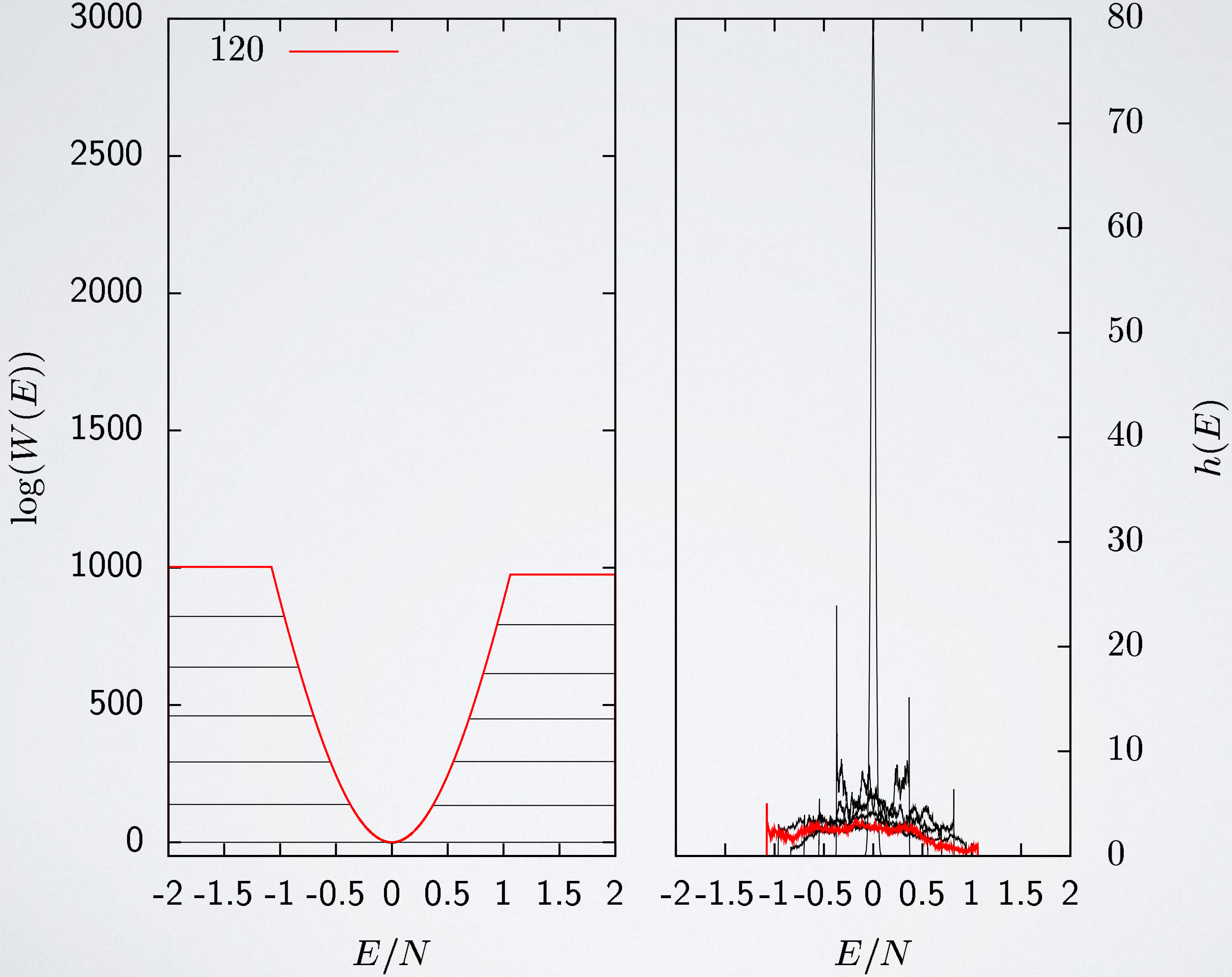
$L = 64$

Multicanonical Monte Carlo Algorithm



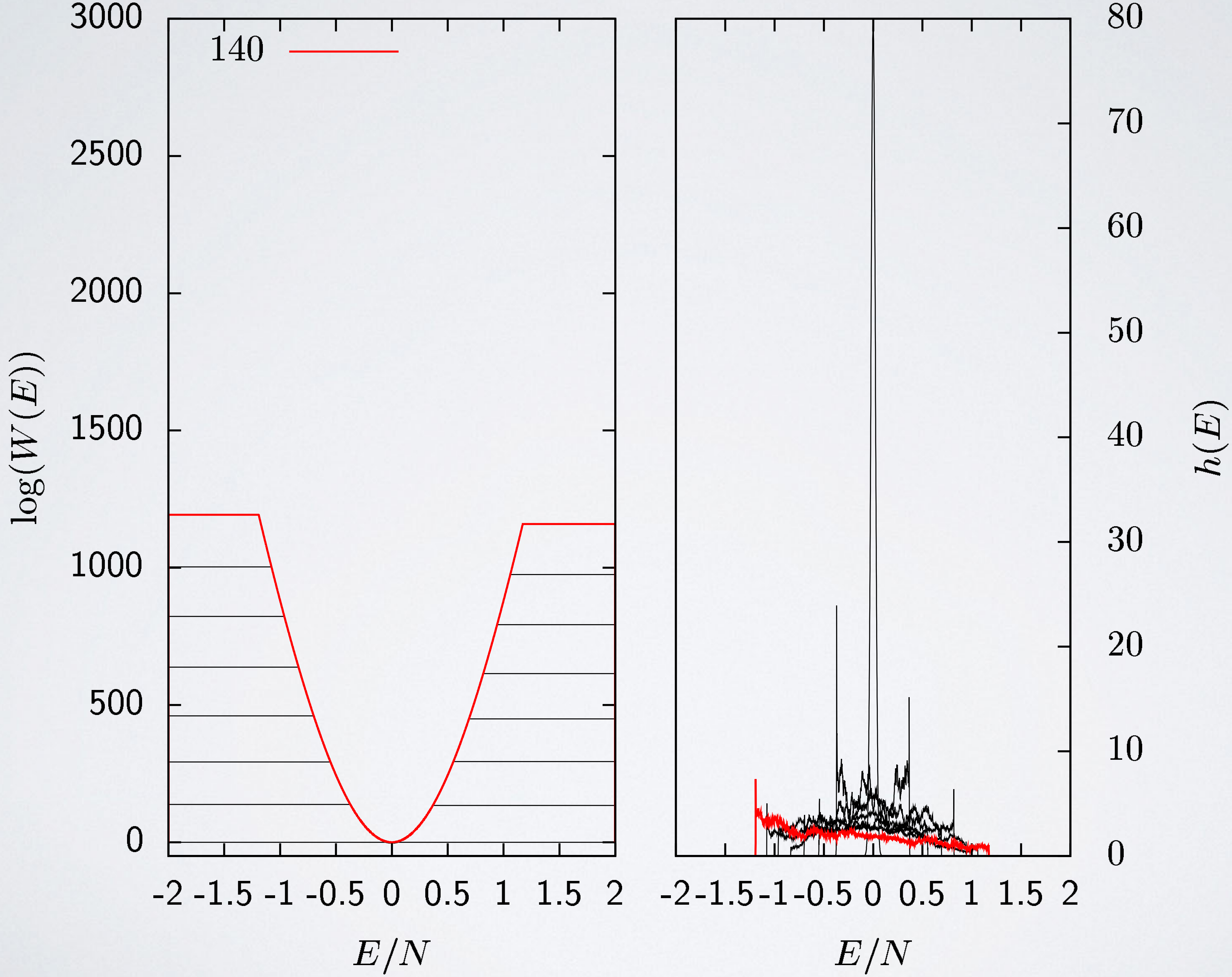
$L = 64$

Multicanonical Monte Carlo Algorithm



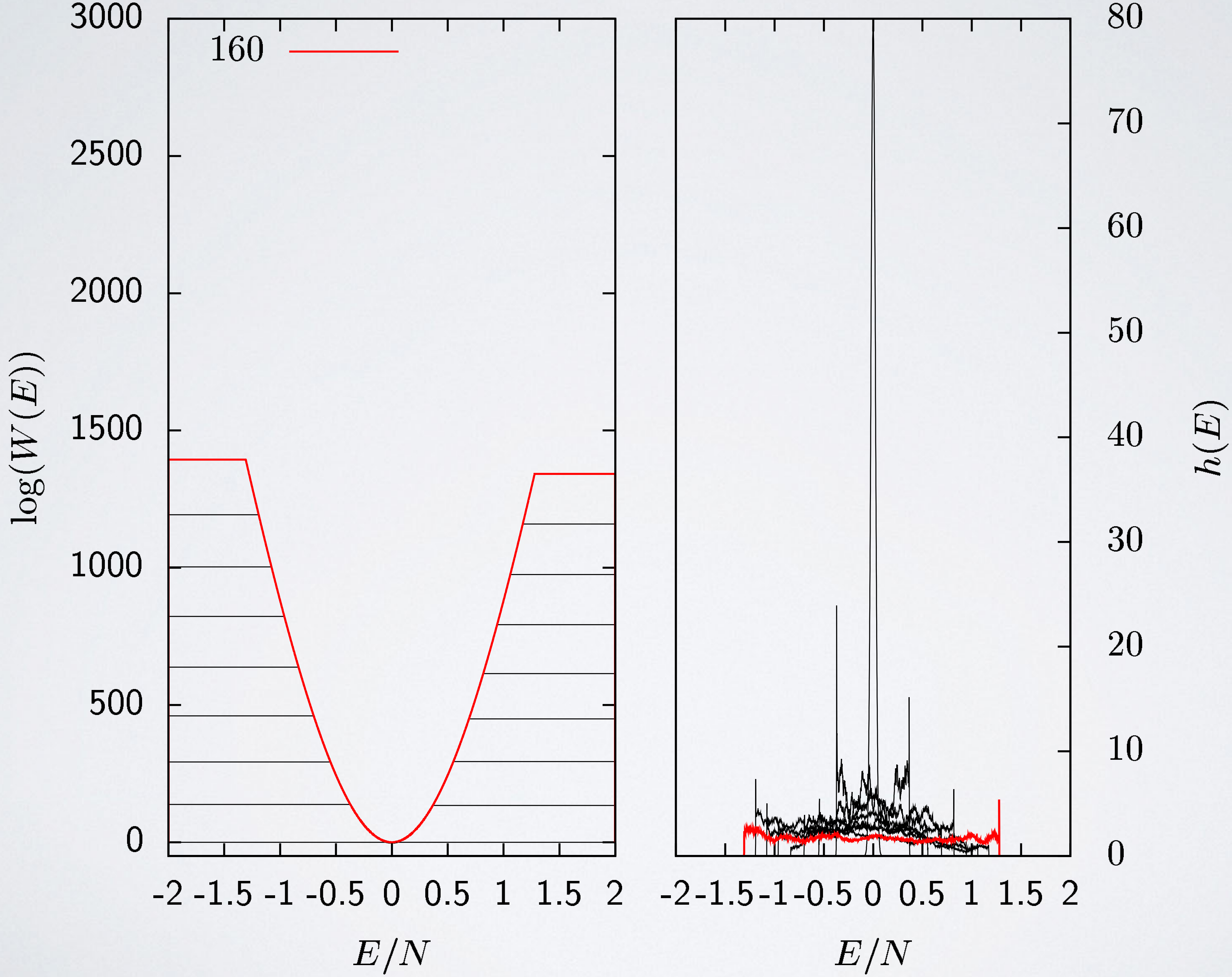
$L = 64$

Multicanonical Monte Carlo Algorithm



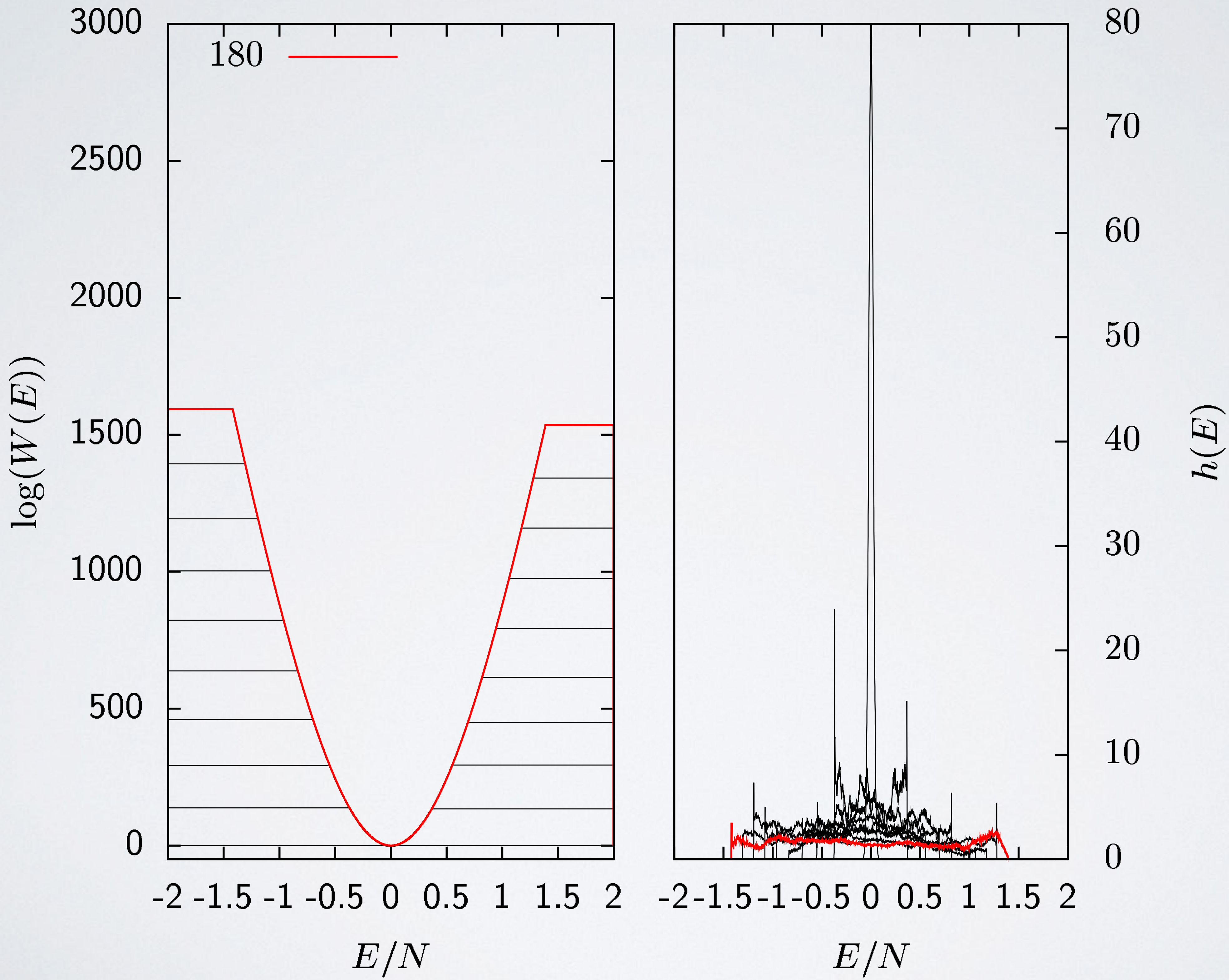
$L = 64$

Multicanonical Monte Carlo Algorithm



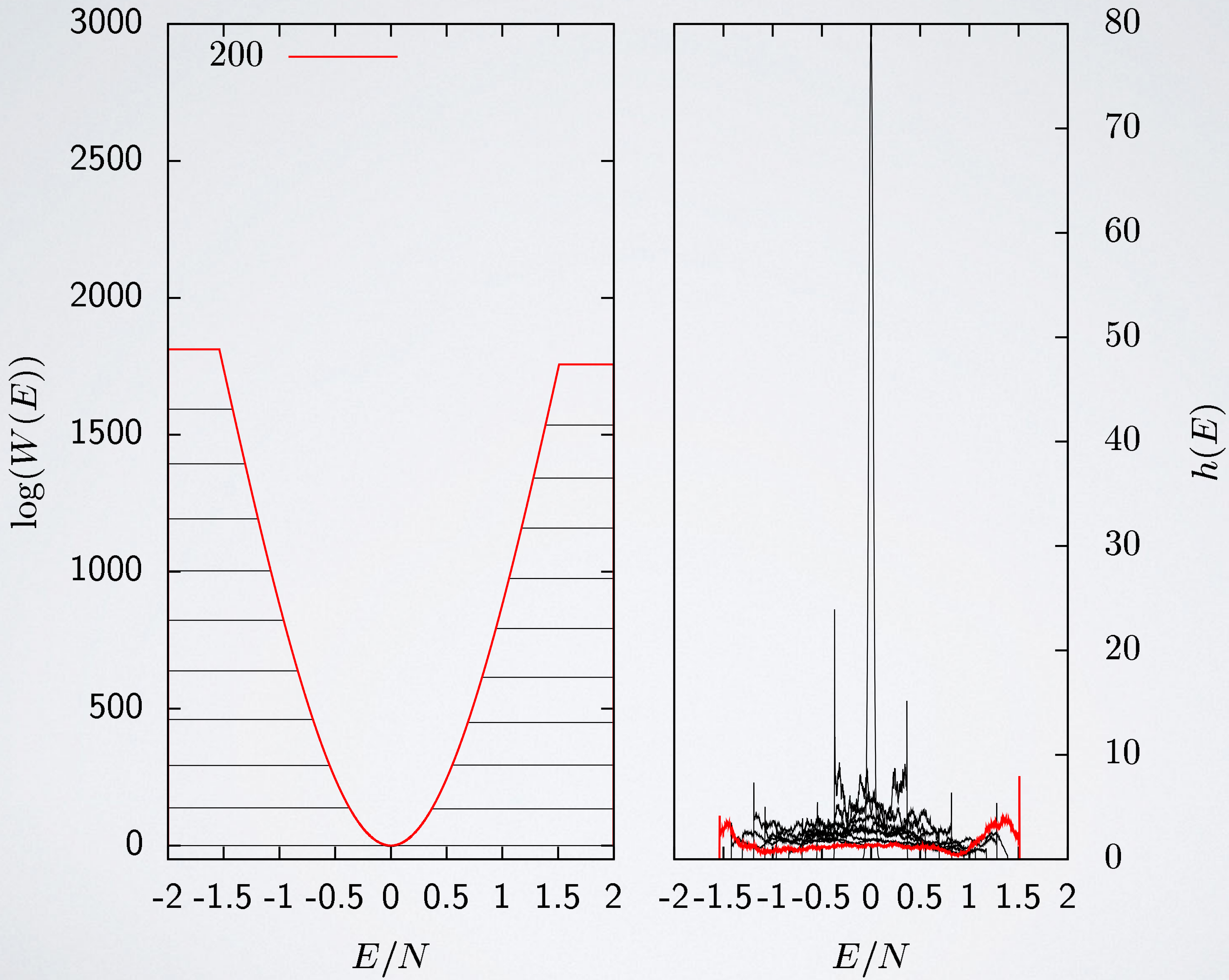
$L = 64$

Multicanonical Monte Carlo Algorithm



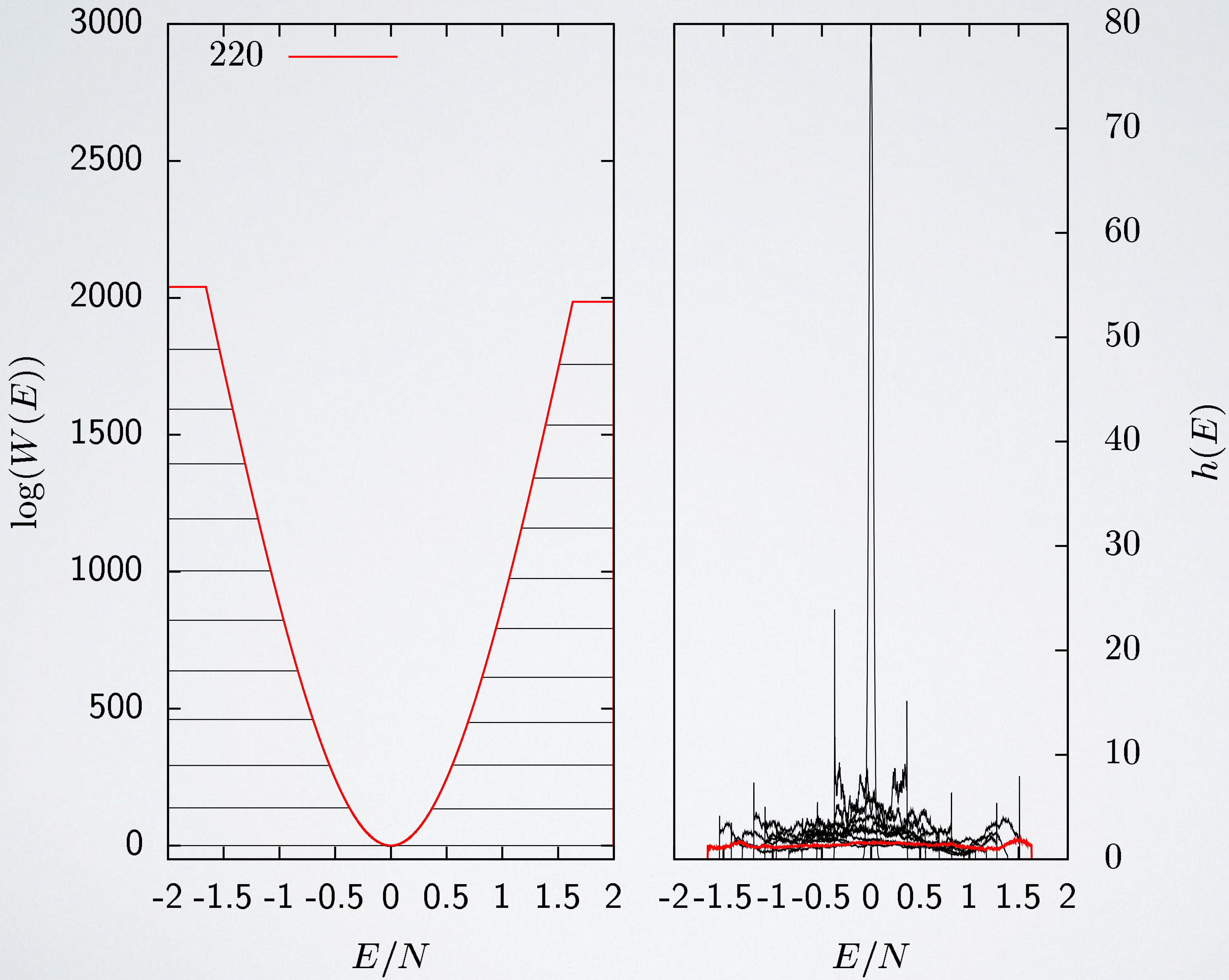
$L = 64$

Multicanonical Monte Carlo Algorithm



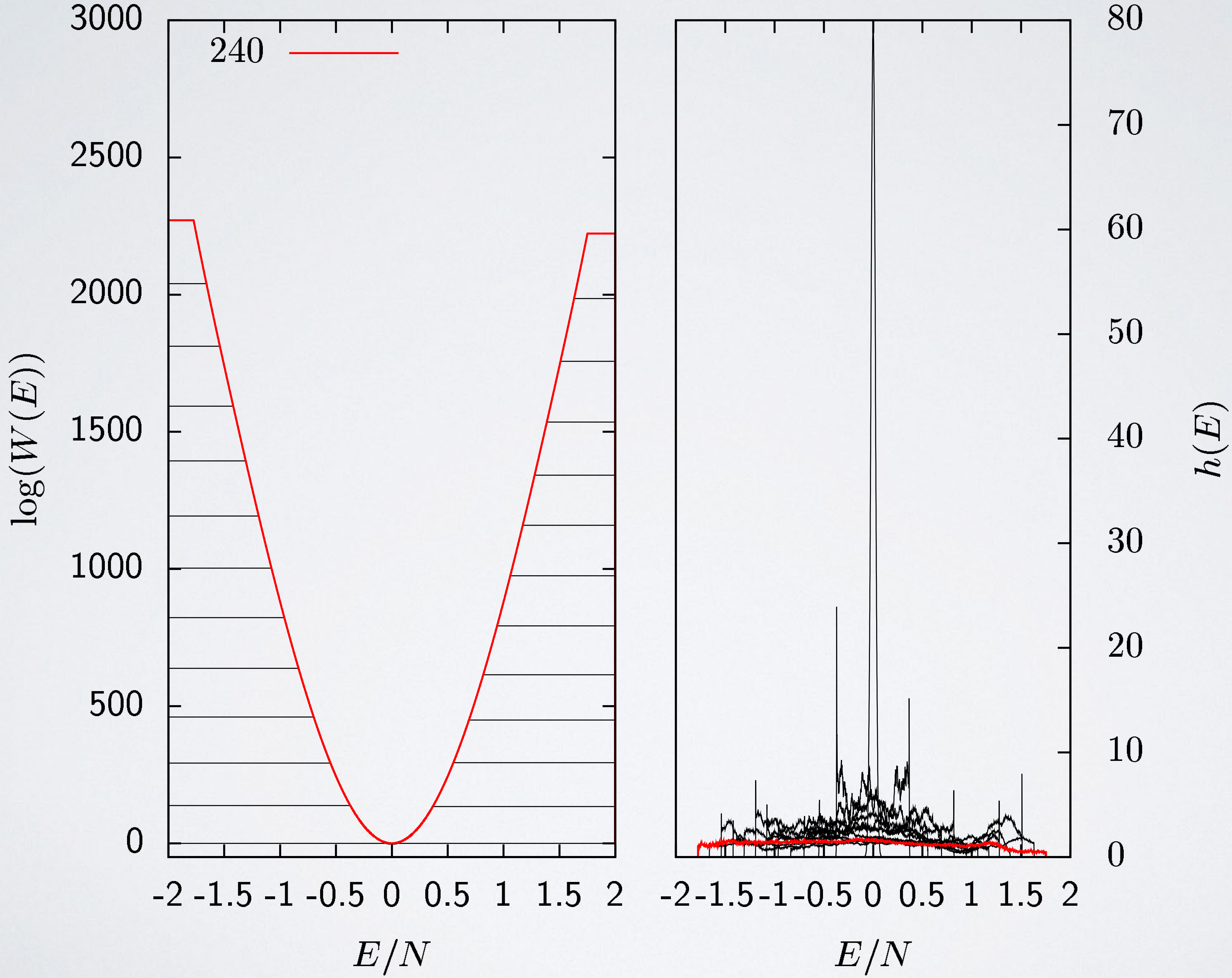
$L = 64$

Multicanonical Monte Carlo Algorithm



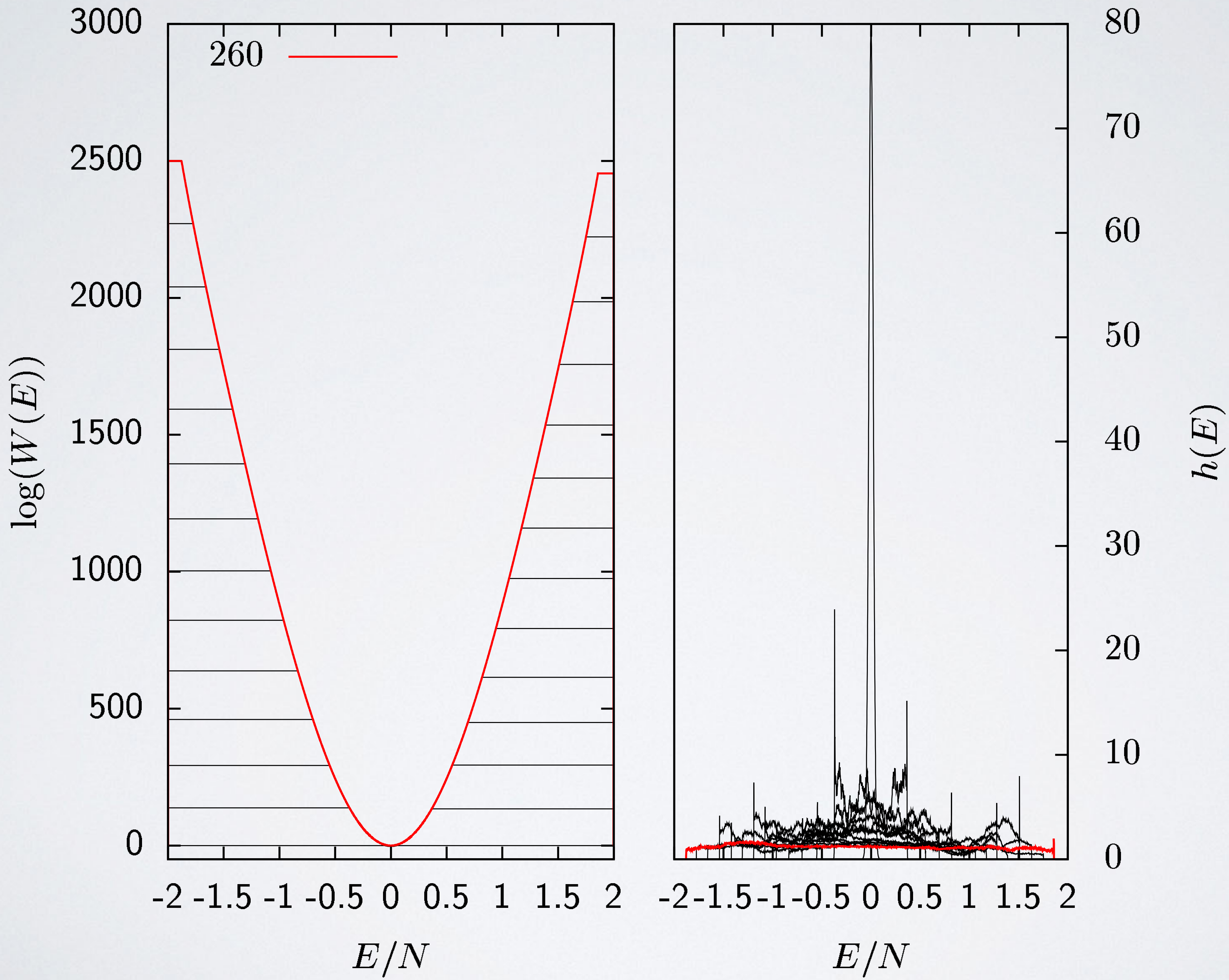
$L = 64$

Multicanonical Monte Carlo Algorithm



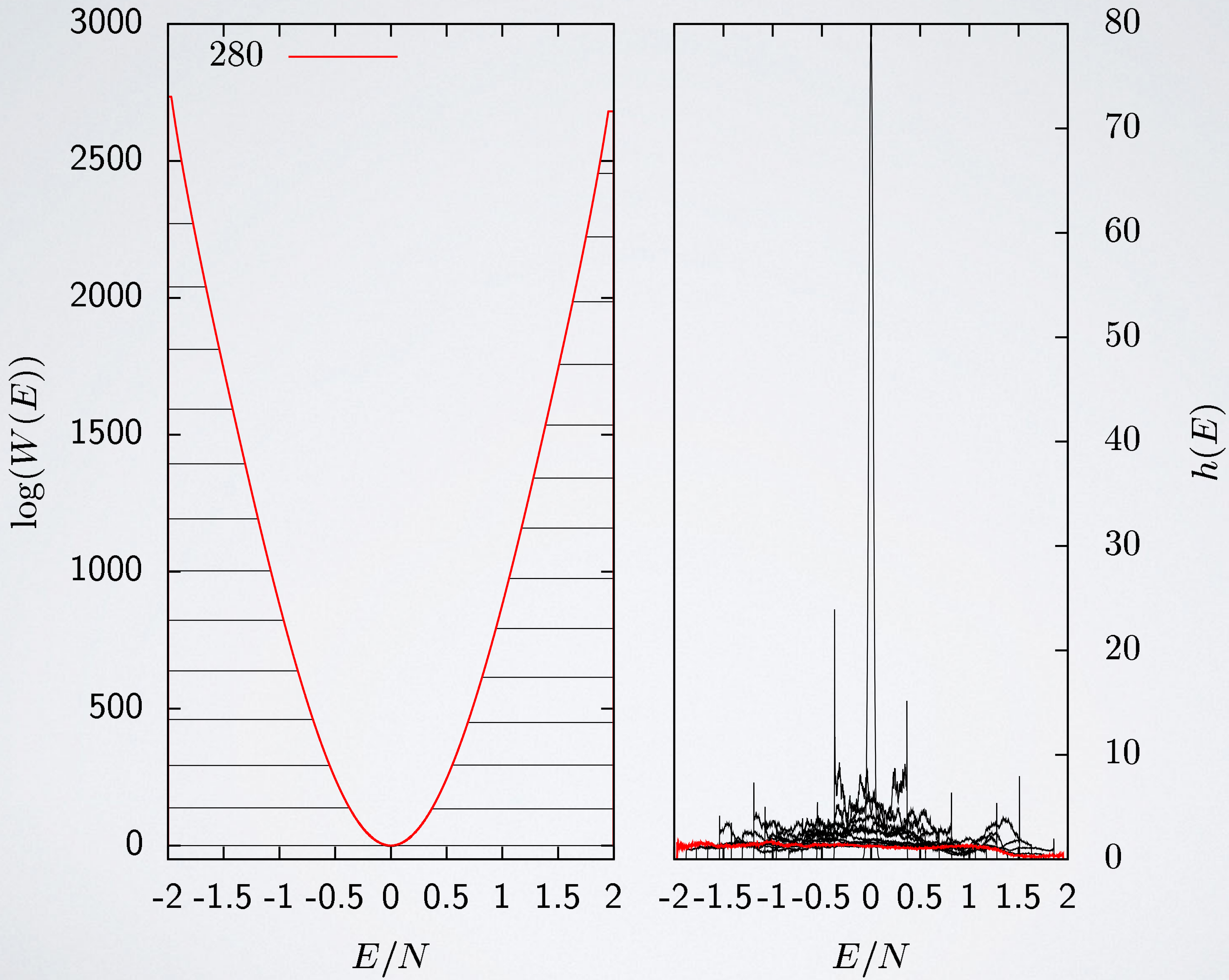
$L = 64$

Multicanonical Monte Carlo Algorithm



$L = 64$

Multicanonical Monte Carlo Algorithm

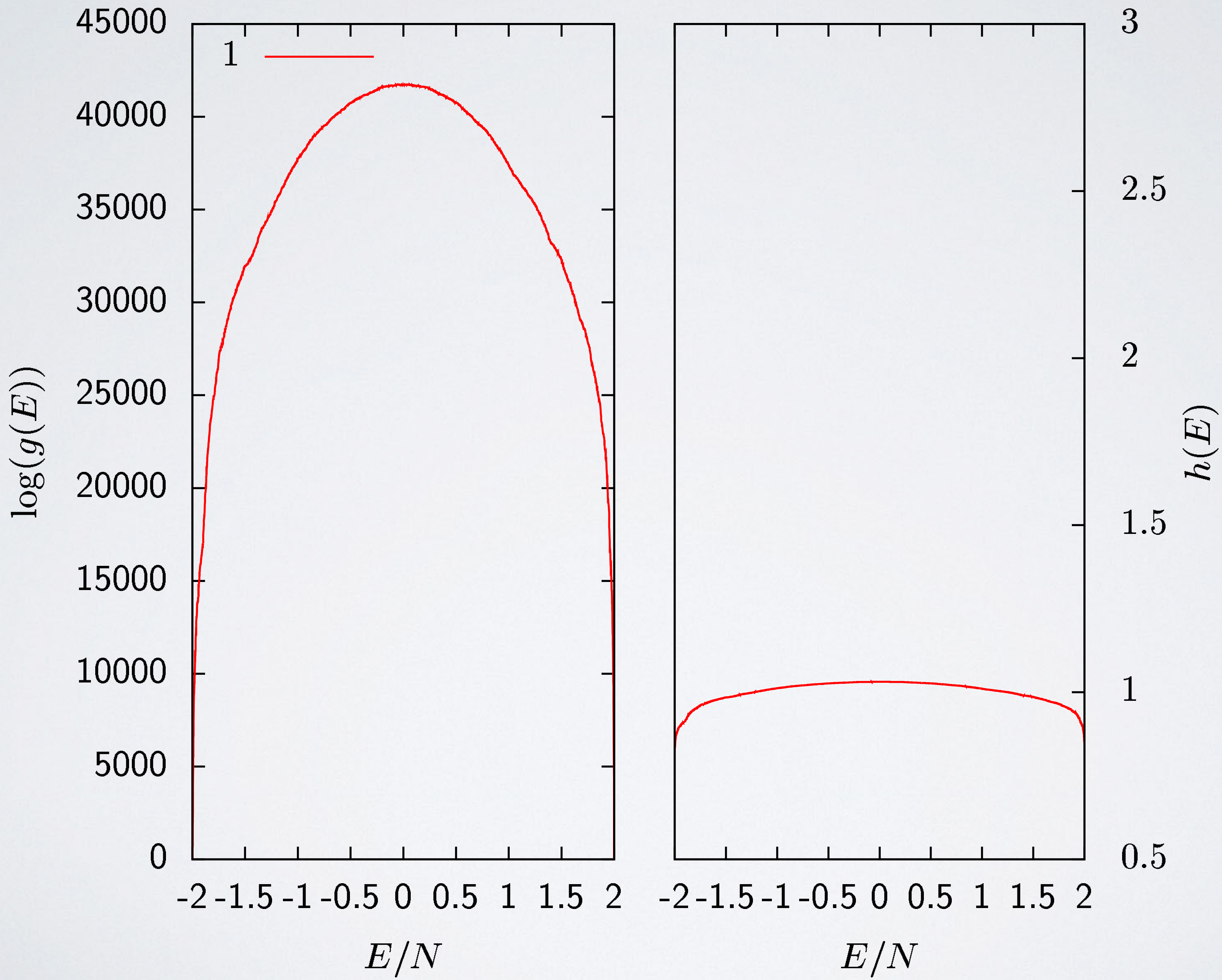


$L = 64$

Wang Landau Algorithm

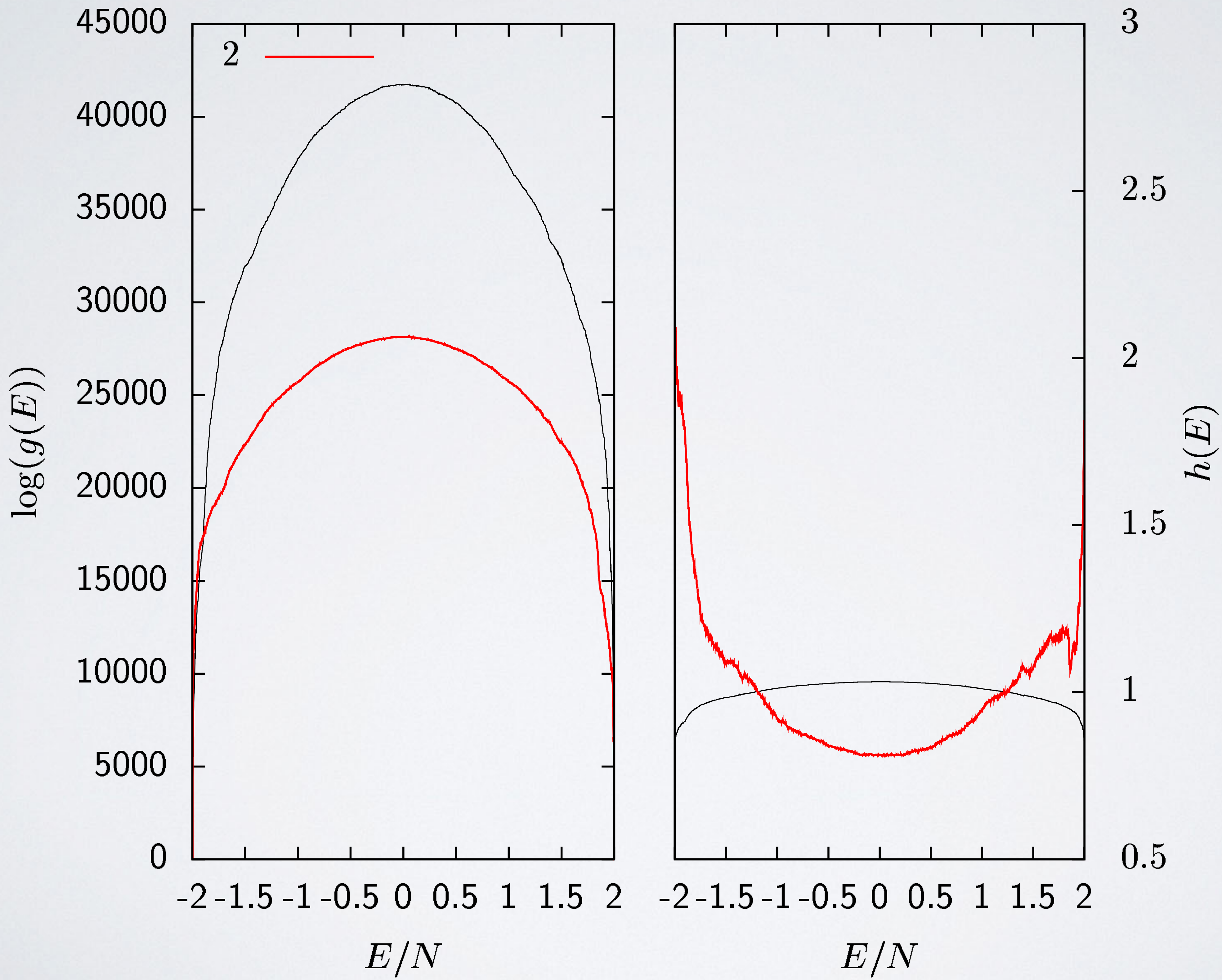
1. Set $g(E) = 1$; choose a modification factor (e.g. $f_0 = e^1$)
2. Choose an initial state
3. Choose a site i
4. Calculate the ratio of the density of states
$$\eta = \frac{g(E_1)}{g(E_2)}$$
which results if the spin at the site i is overturned
5. Generate a random number r such that $0 < r < 1$
6. If $r < \eta$, flip the spin
7. Set $g(E_i) \to g(E_i) * f$
8. If the histogram is not *flat*, go to the next site and go to 4.
9. If the histogram is *flat*, decrease f , e.g. $f_{i+1} = f^{1/2}$
10. Repeat step 3-9 until $f = f_{\min} \sim \exp(10^{-8})$

Wang Landau Algorithm



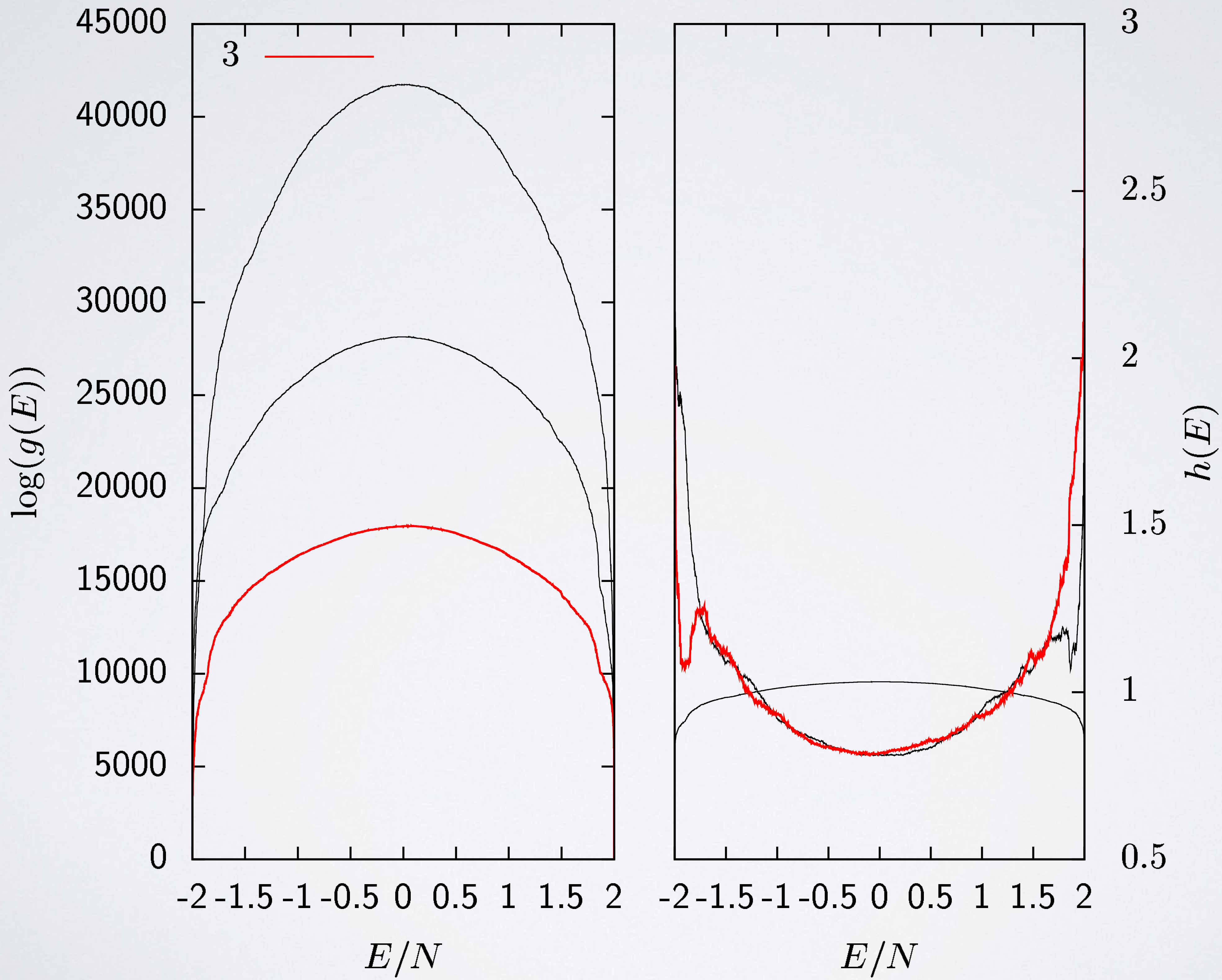
$L = 64$

Wang Landau Algorithm



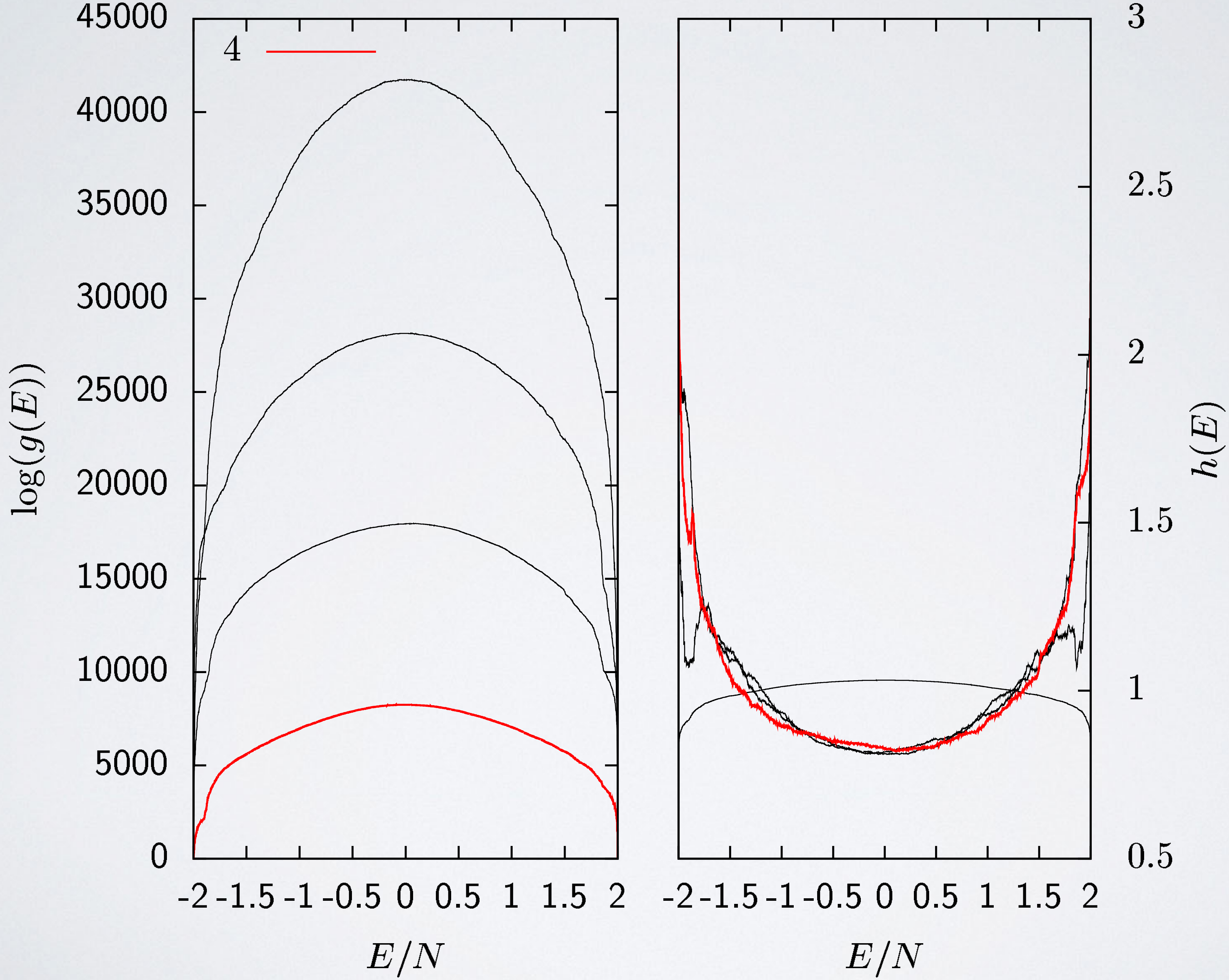
$L = 64$

Wang Landau Algorithm



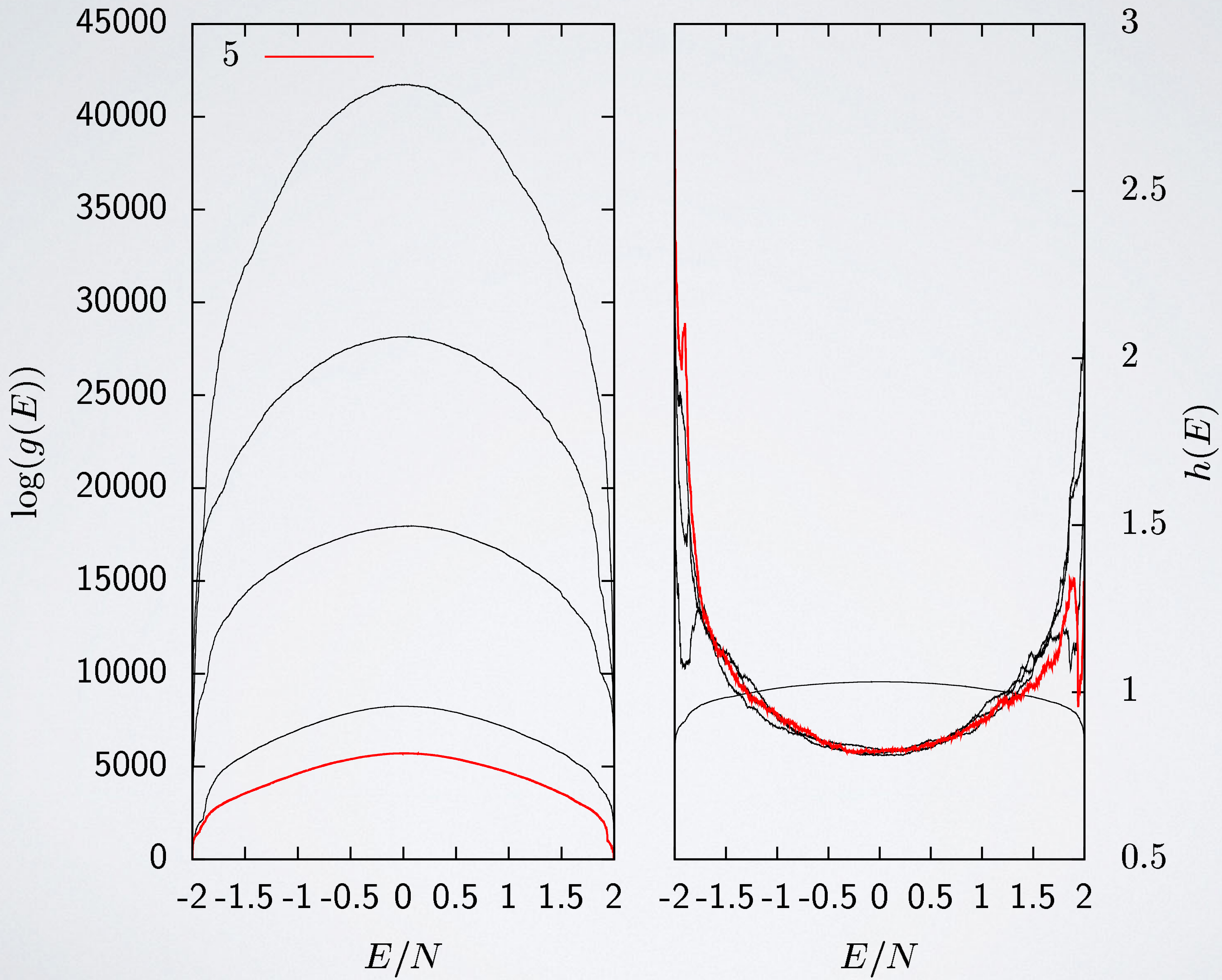
$L = 64$

Wang Landau Algorithm



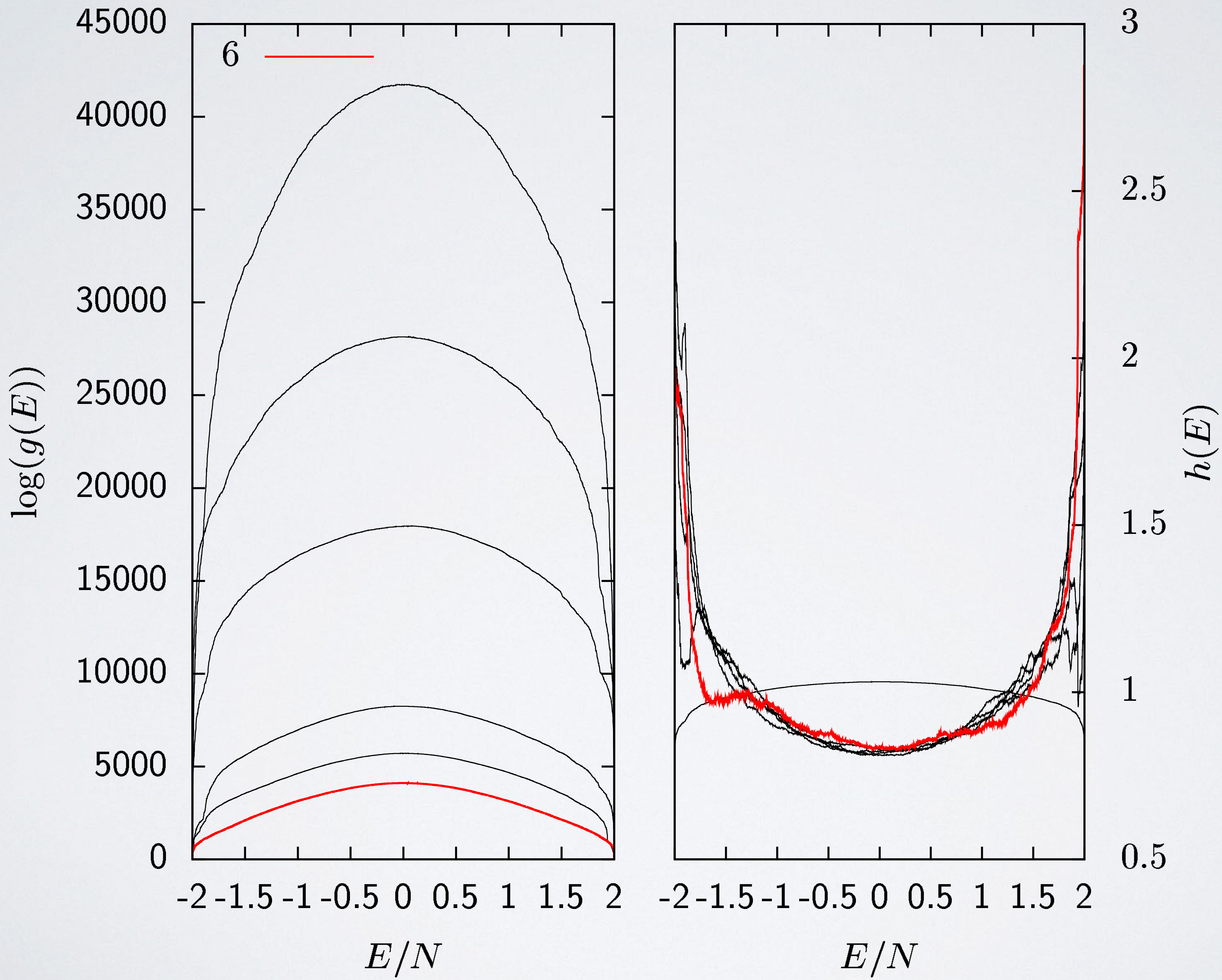
$L = 64$

Wang Landau Algorithm



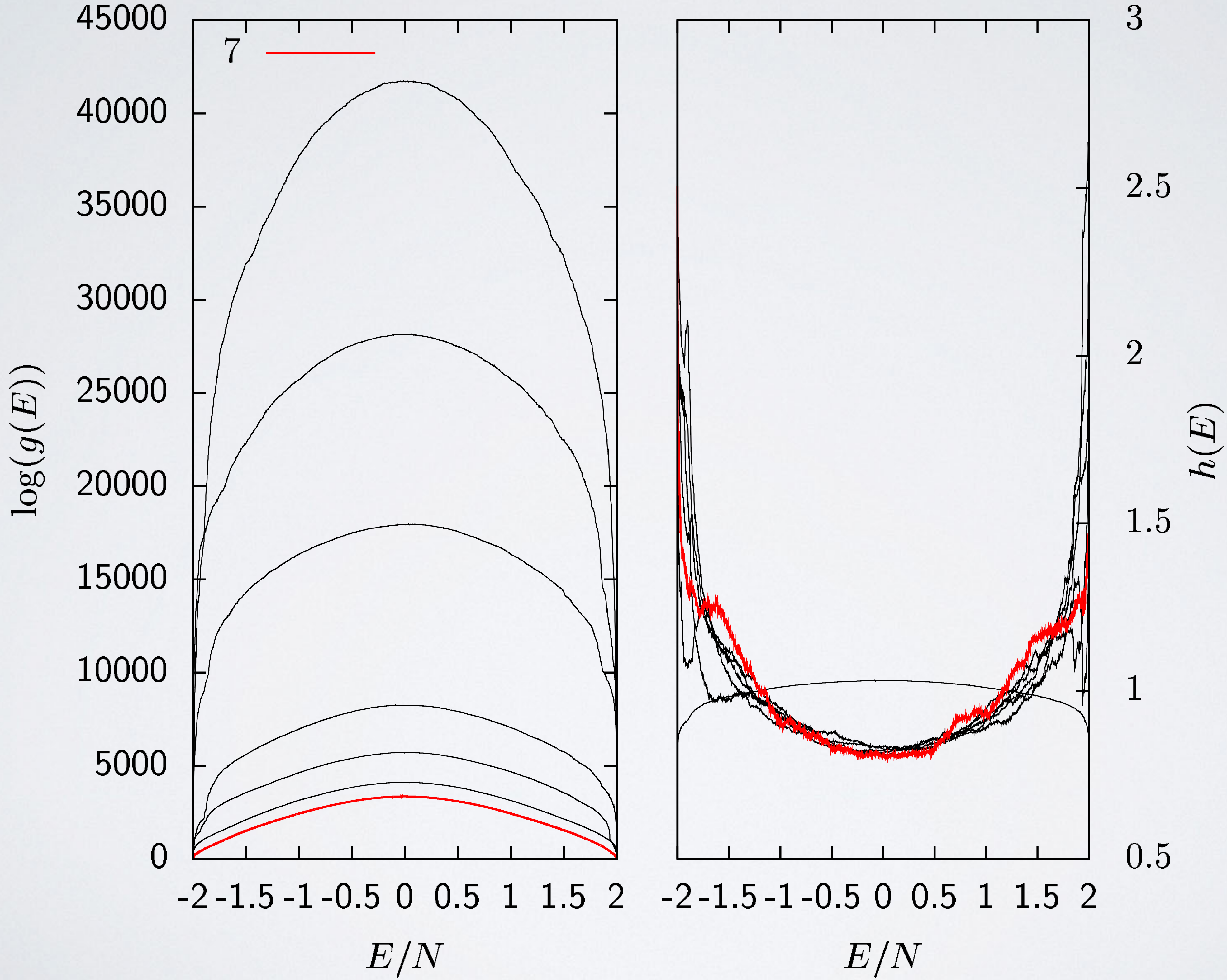
$L = 64$

Wang Landau Algorithm



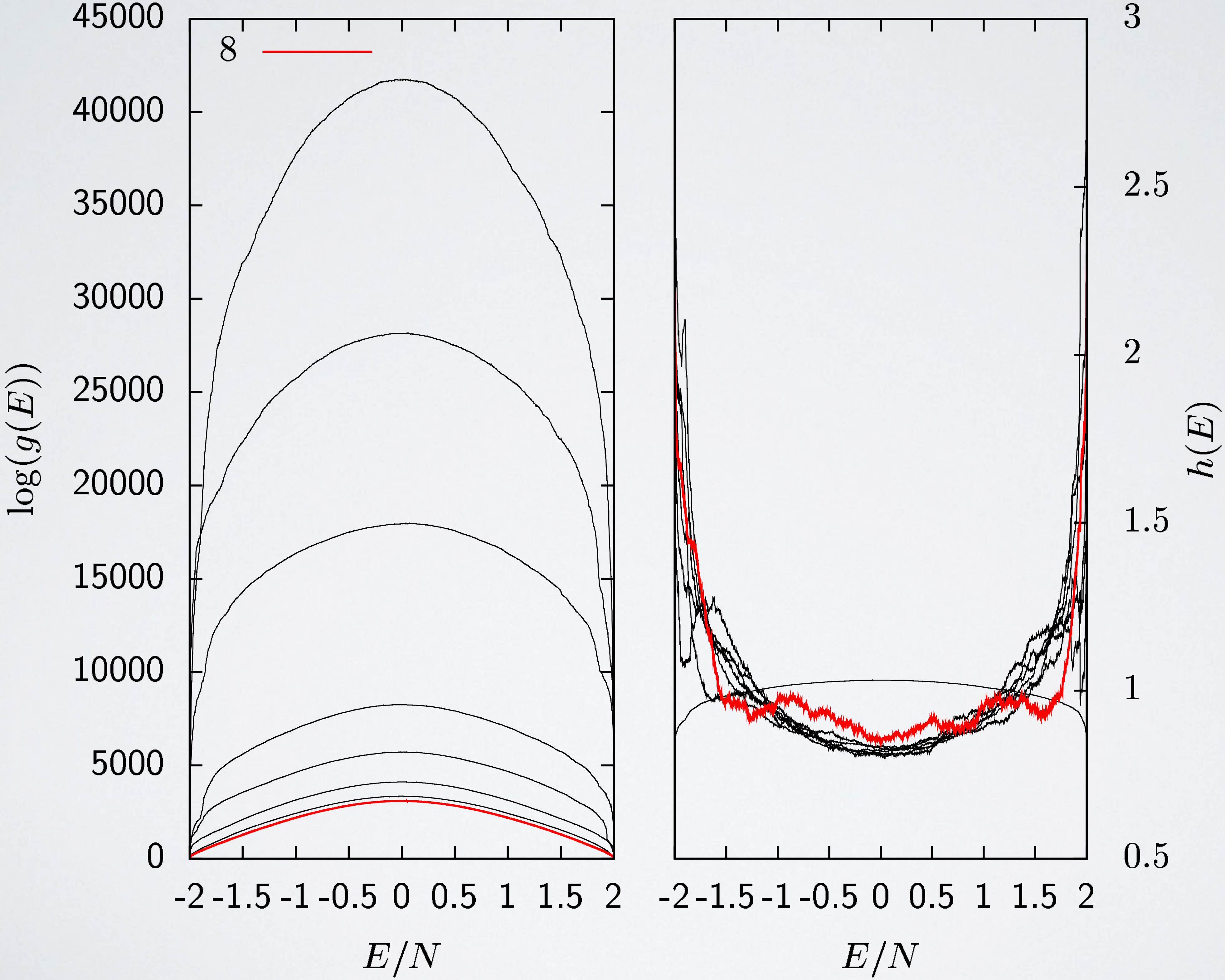
$L = 64$

Wang Landau Algorithm



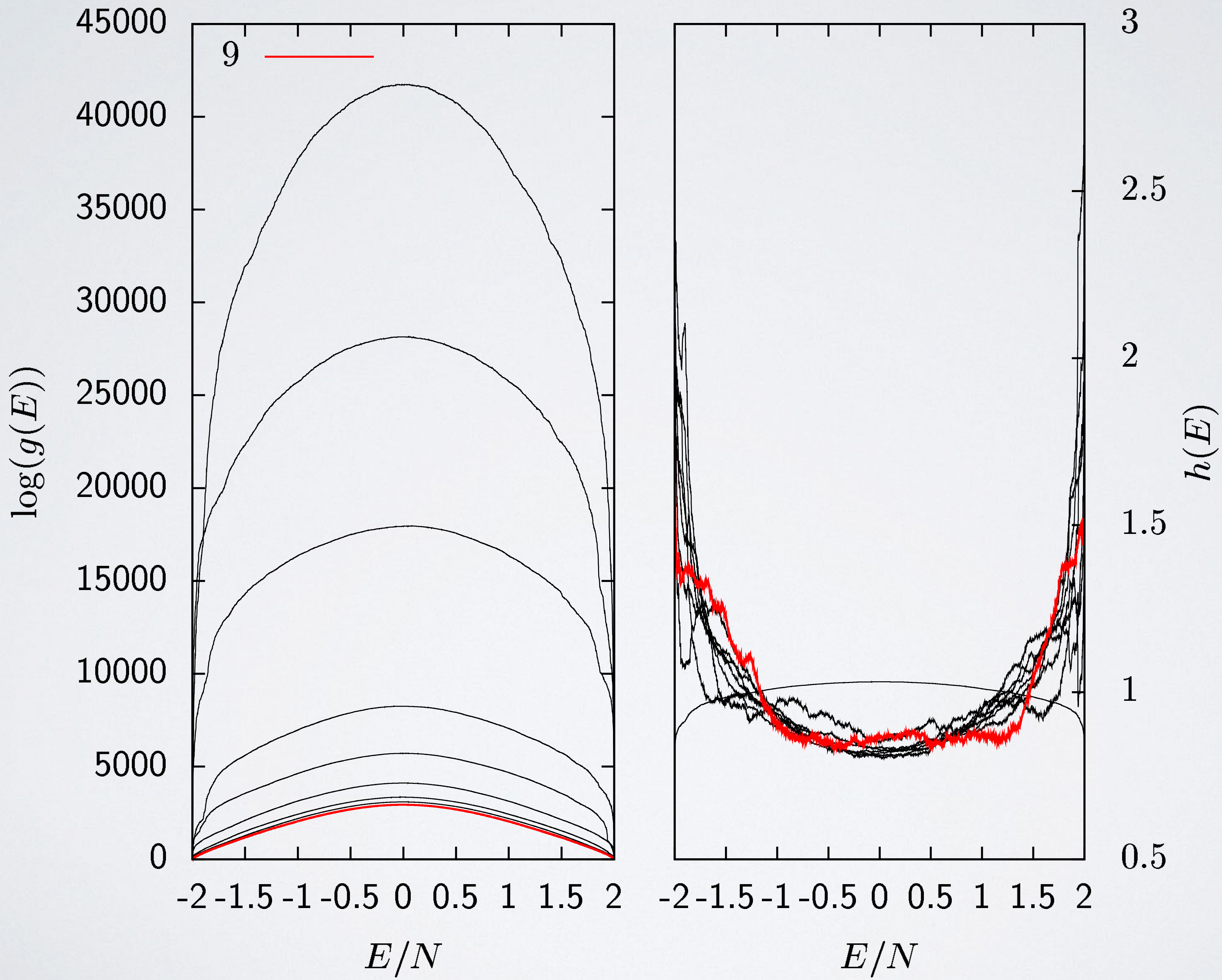
$L = 64$

Wang Landau Algorithm



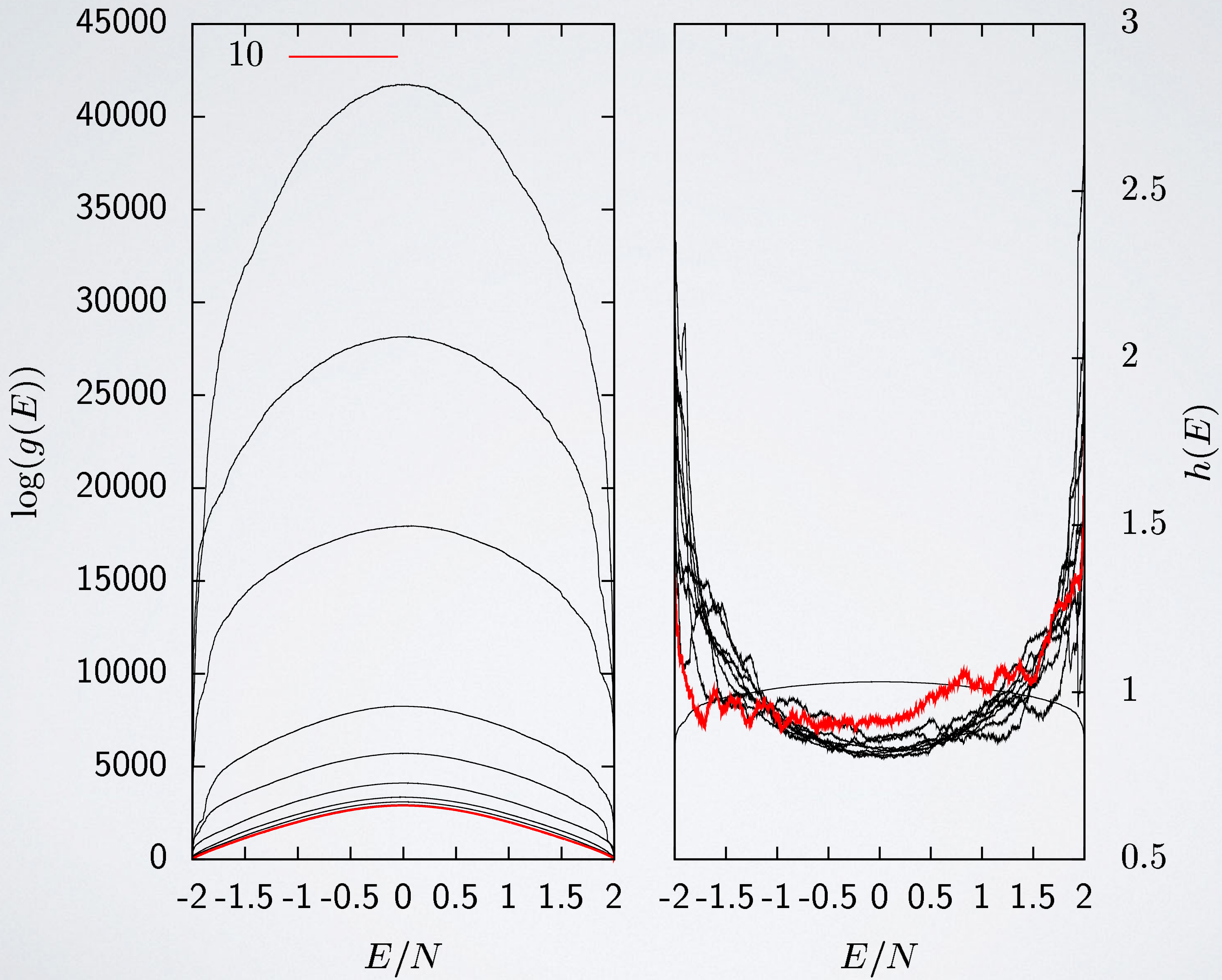
$L = 64$

Wang Landau Algorithm



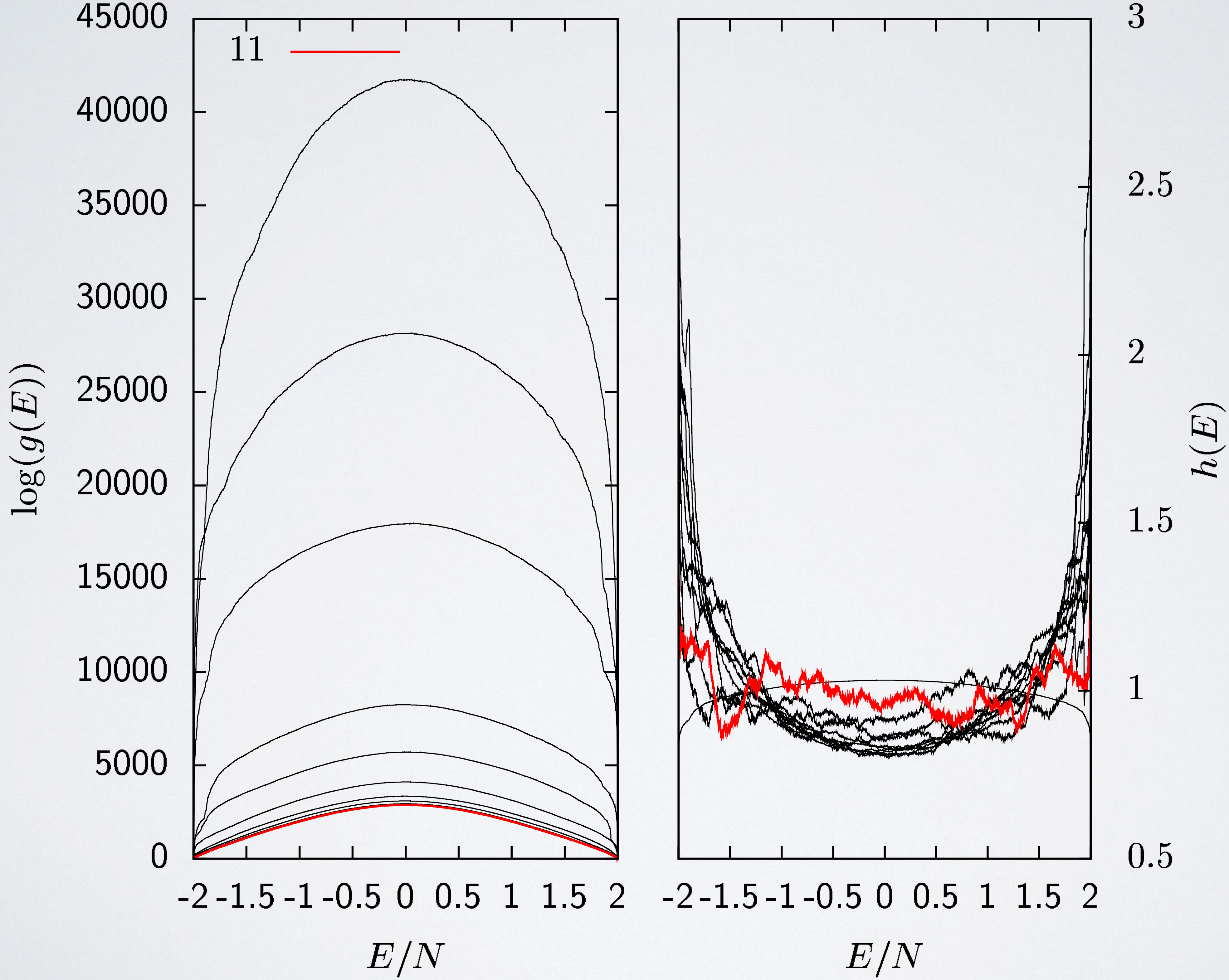
$L = 64$

Wang Landau Algorithm



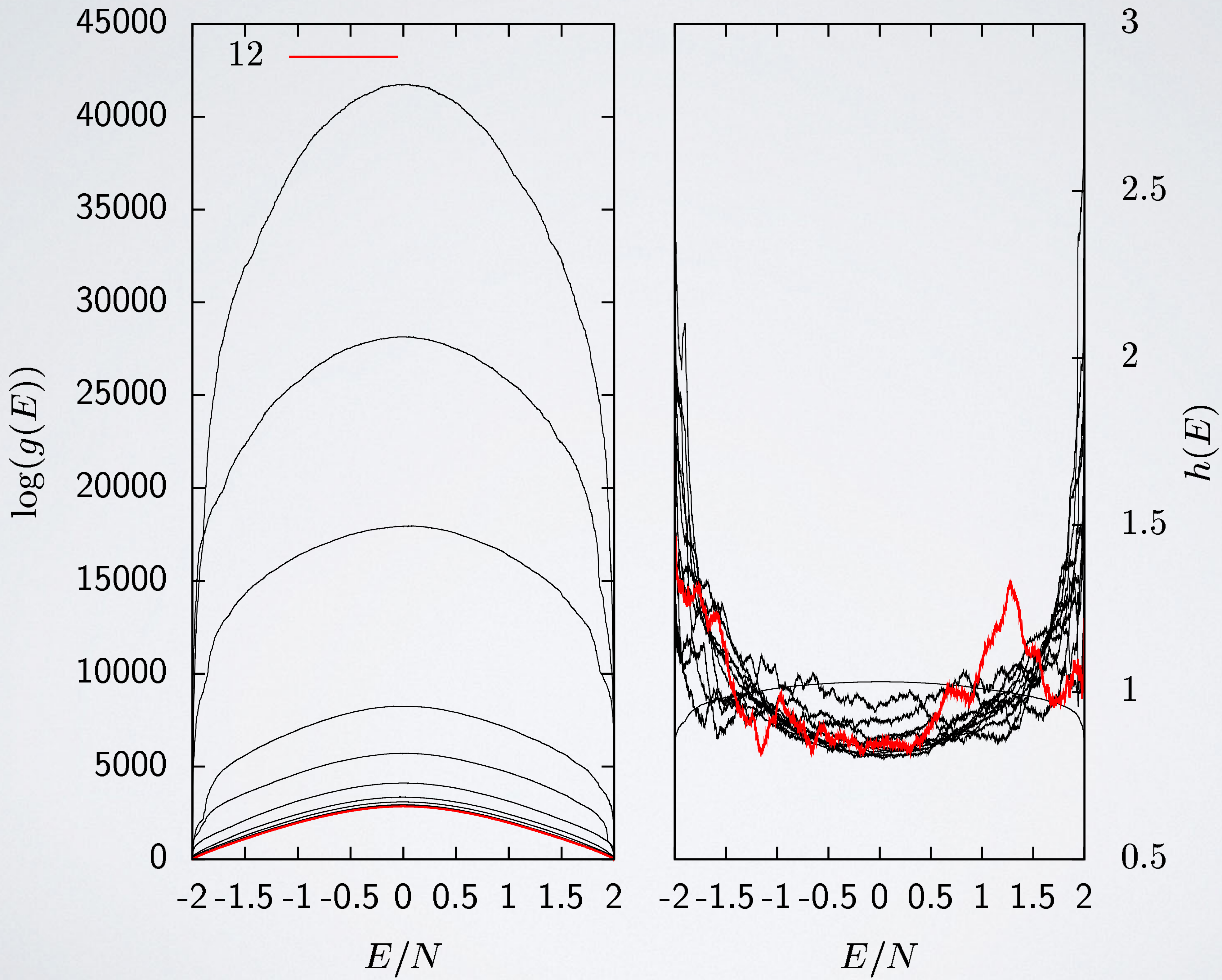
$L = 64$

Wang Landau Algorithm



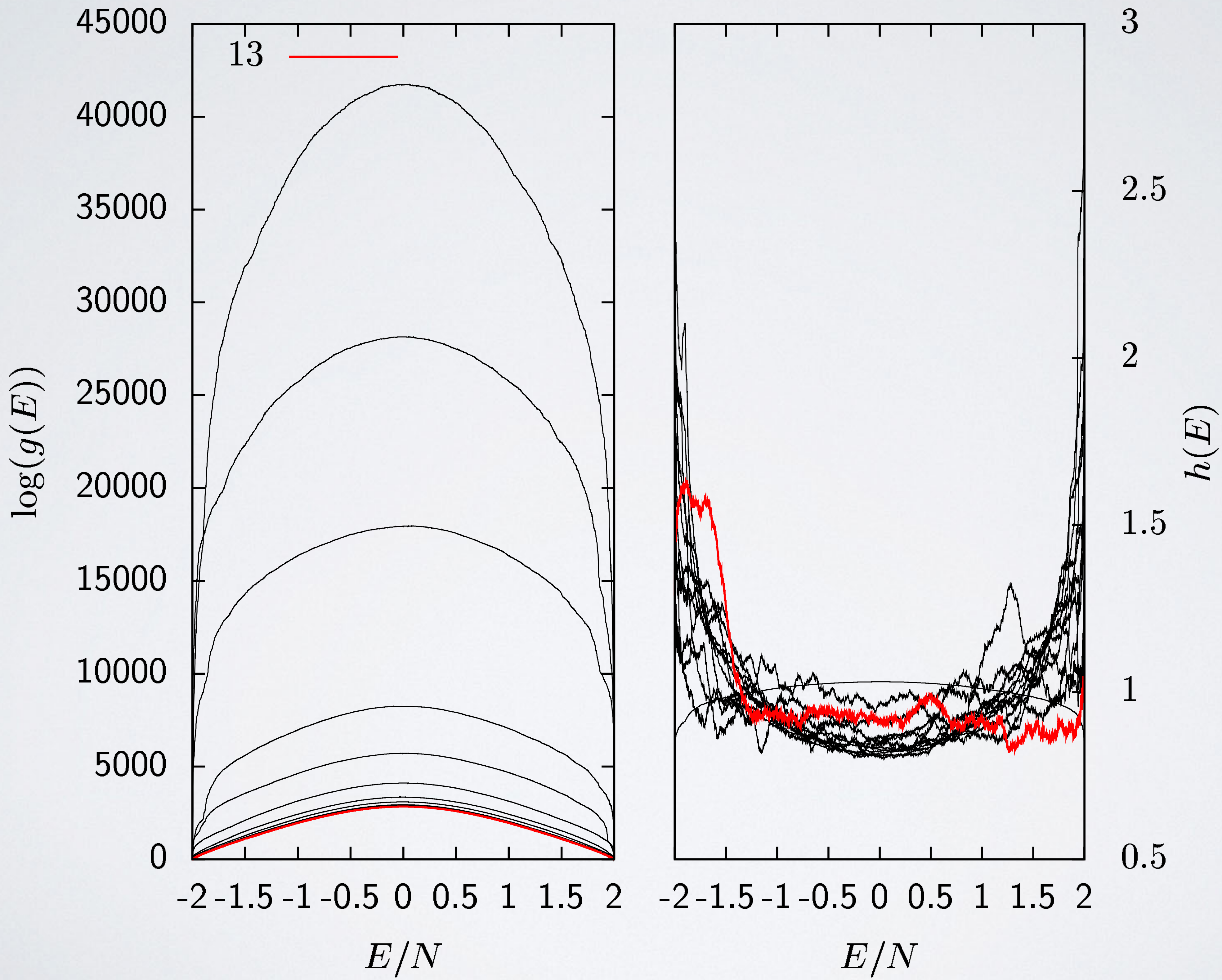
$L = 64$

Wang Landau Algorithm



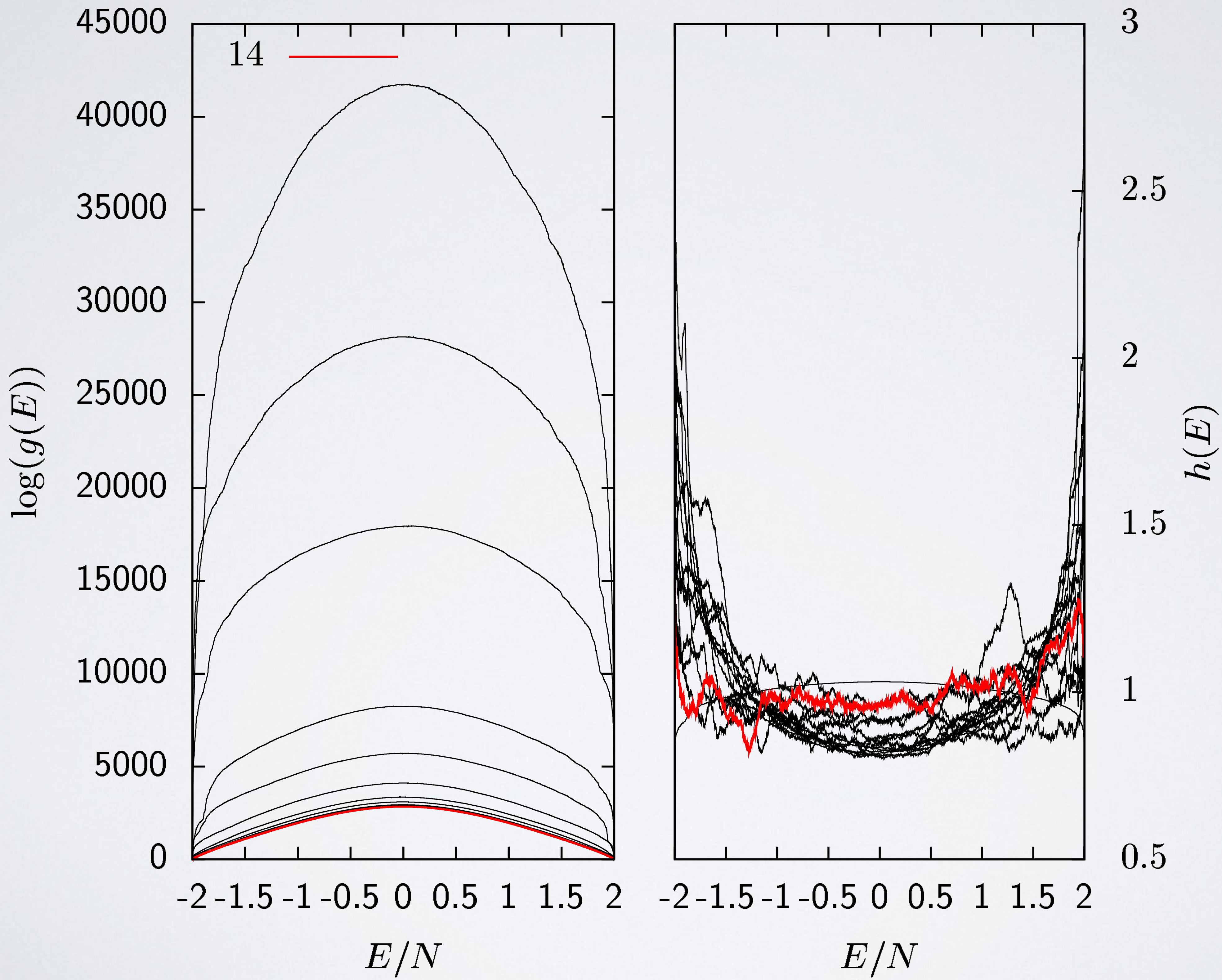
$L = 64$

Wang Landau Algorithm



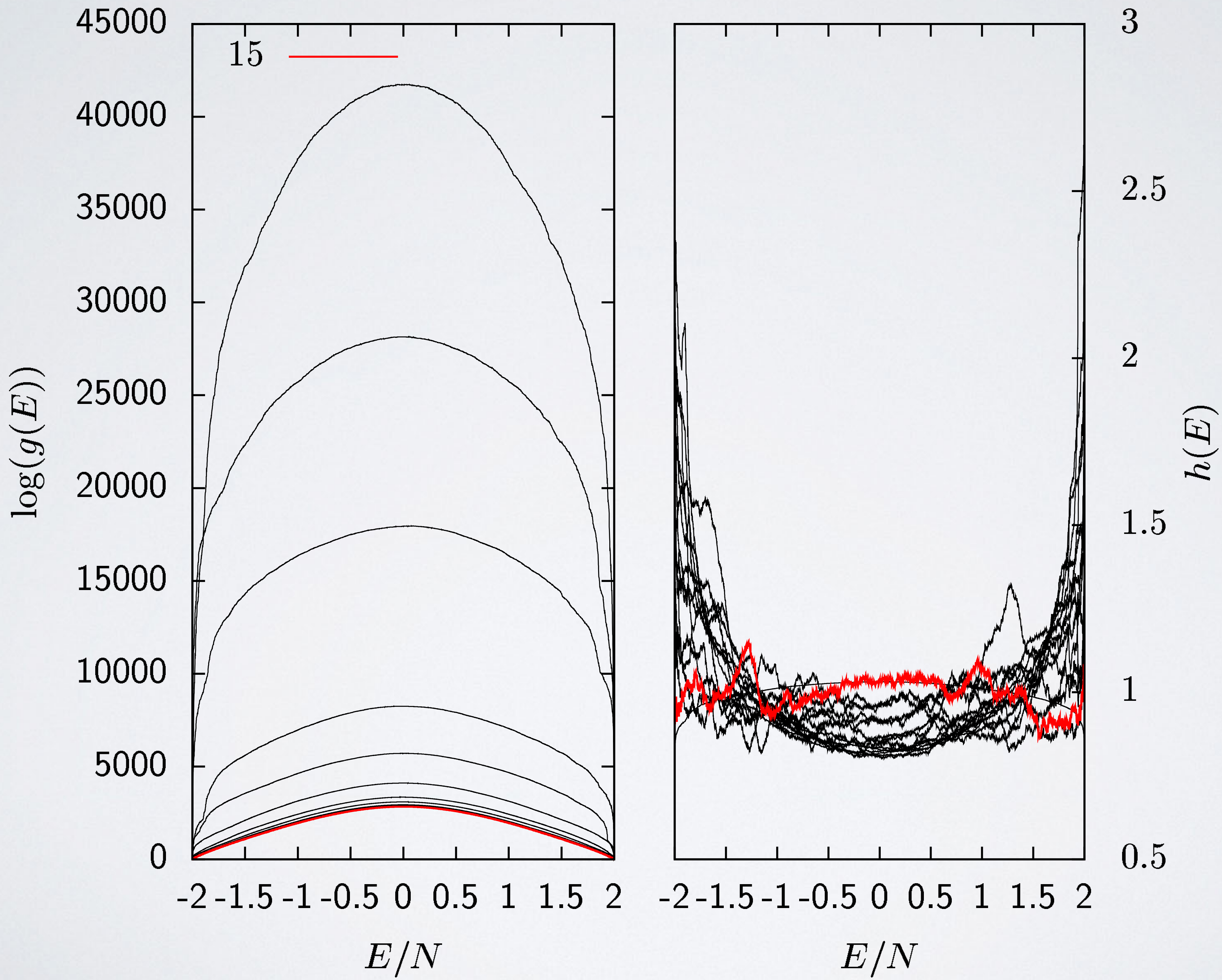
$L = 64$

Wang Landau Algorithm



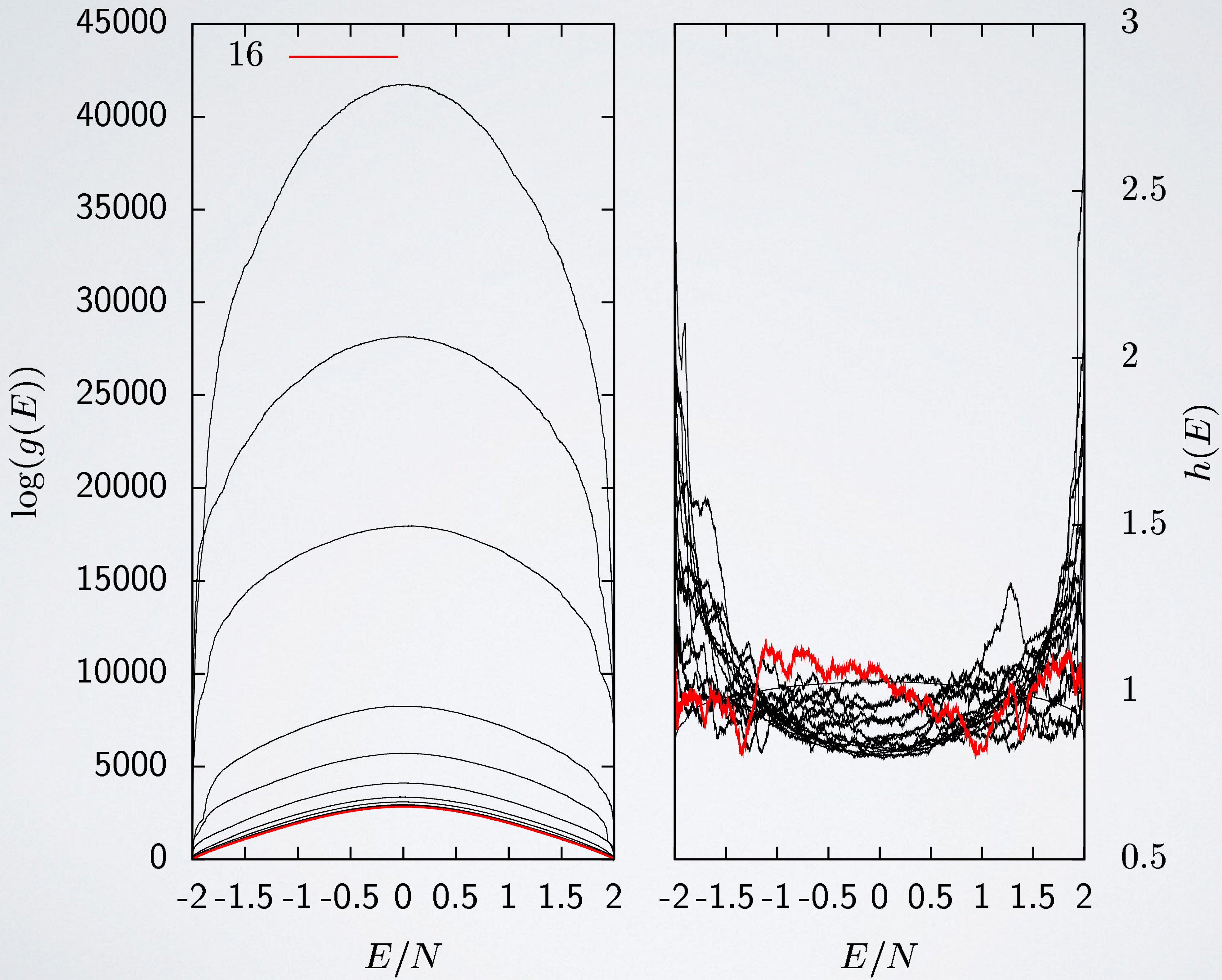
$L = 64$

Wang Landau Algorithm



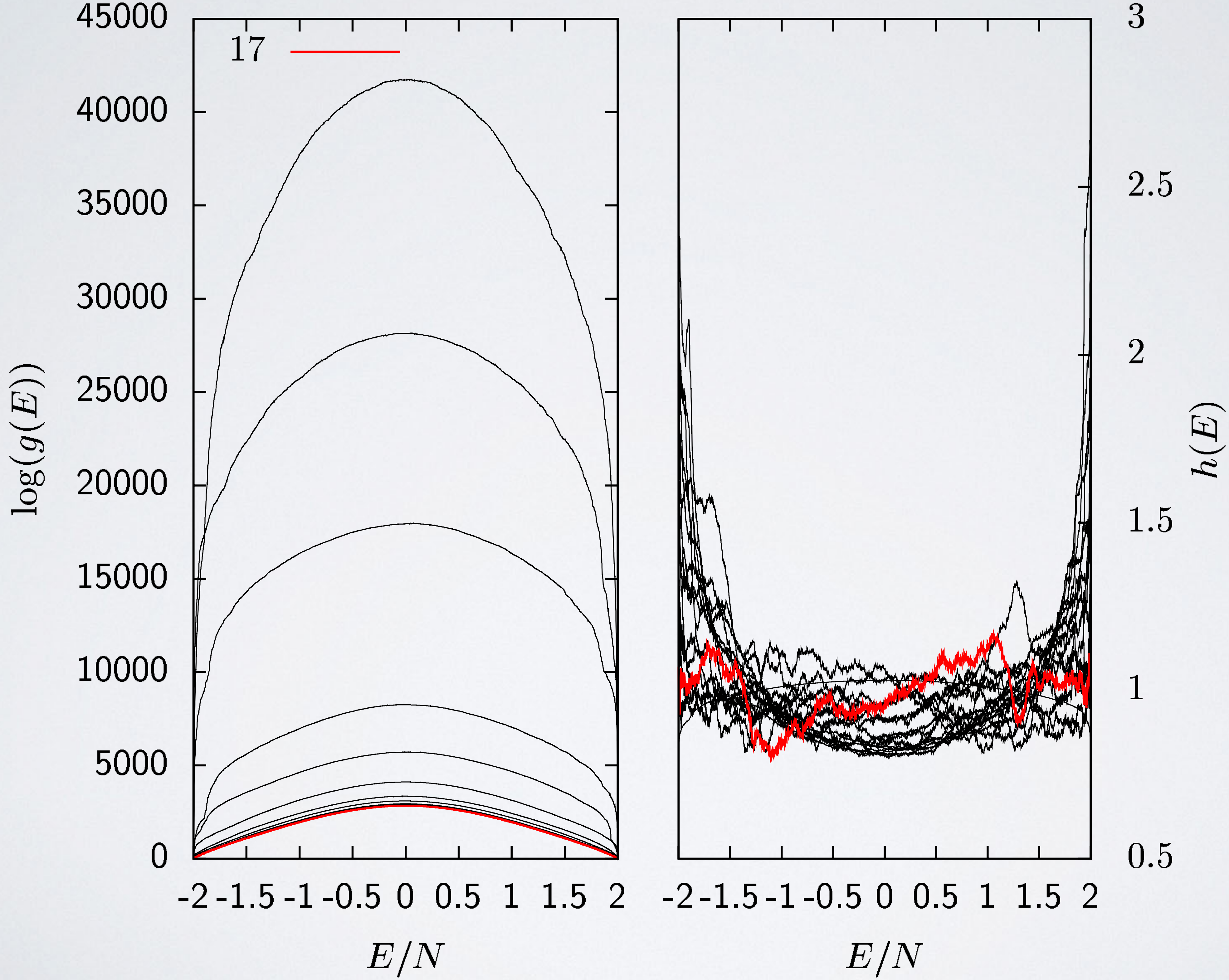
$L = 64$

Wang Landau Algorithm



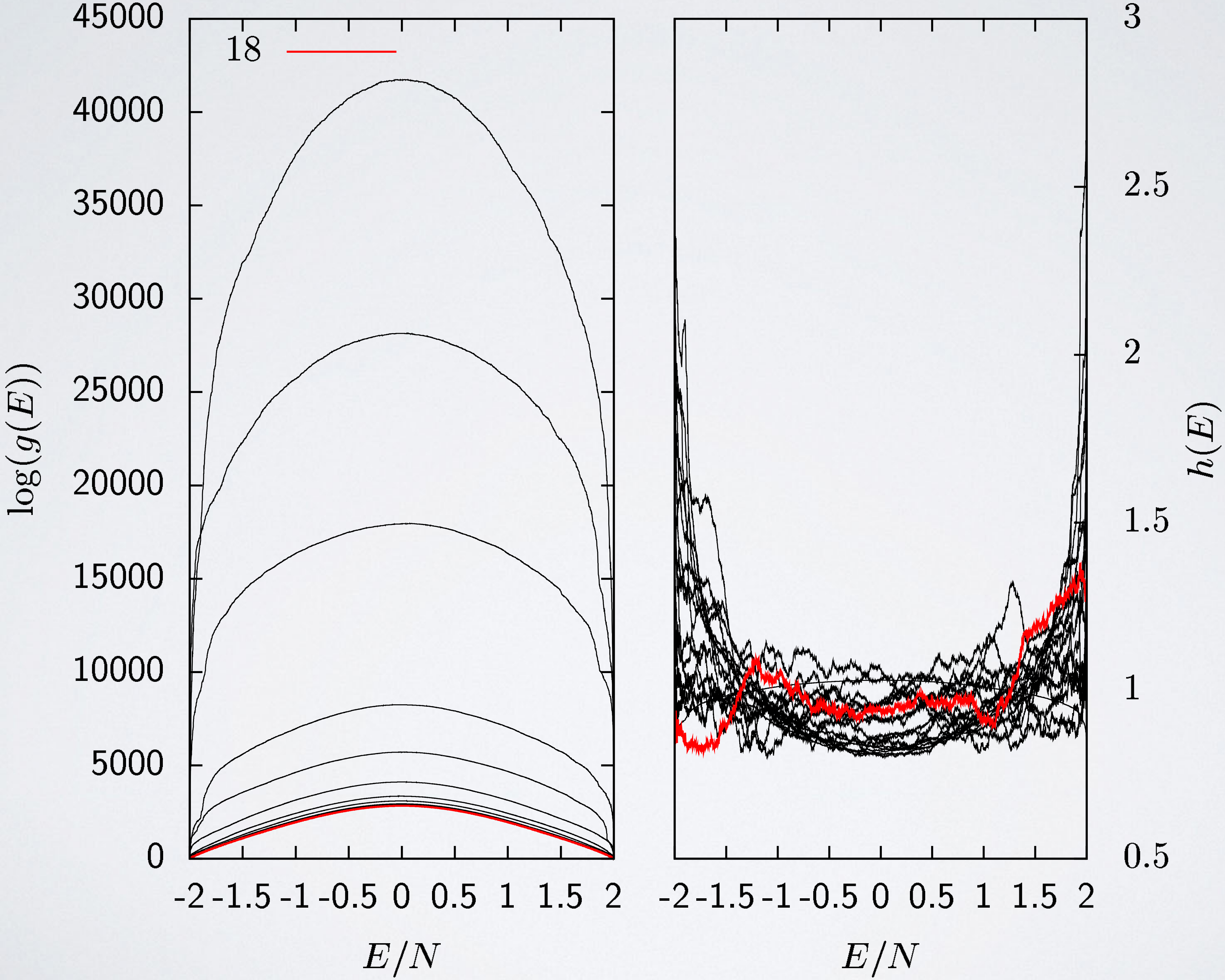
$L = 64$

Wang Landau Algorithm



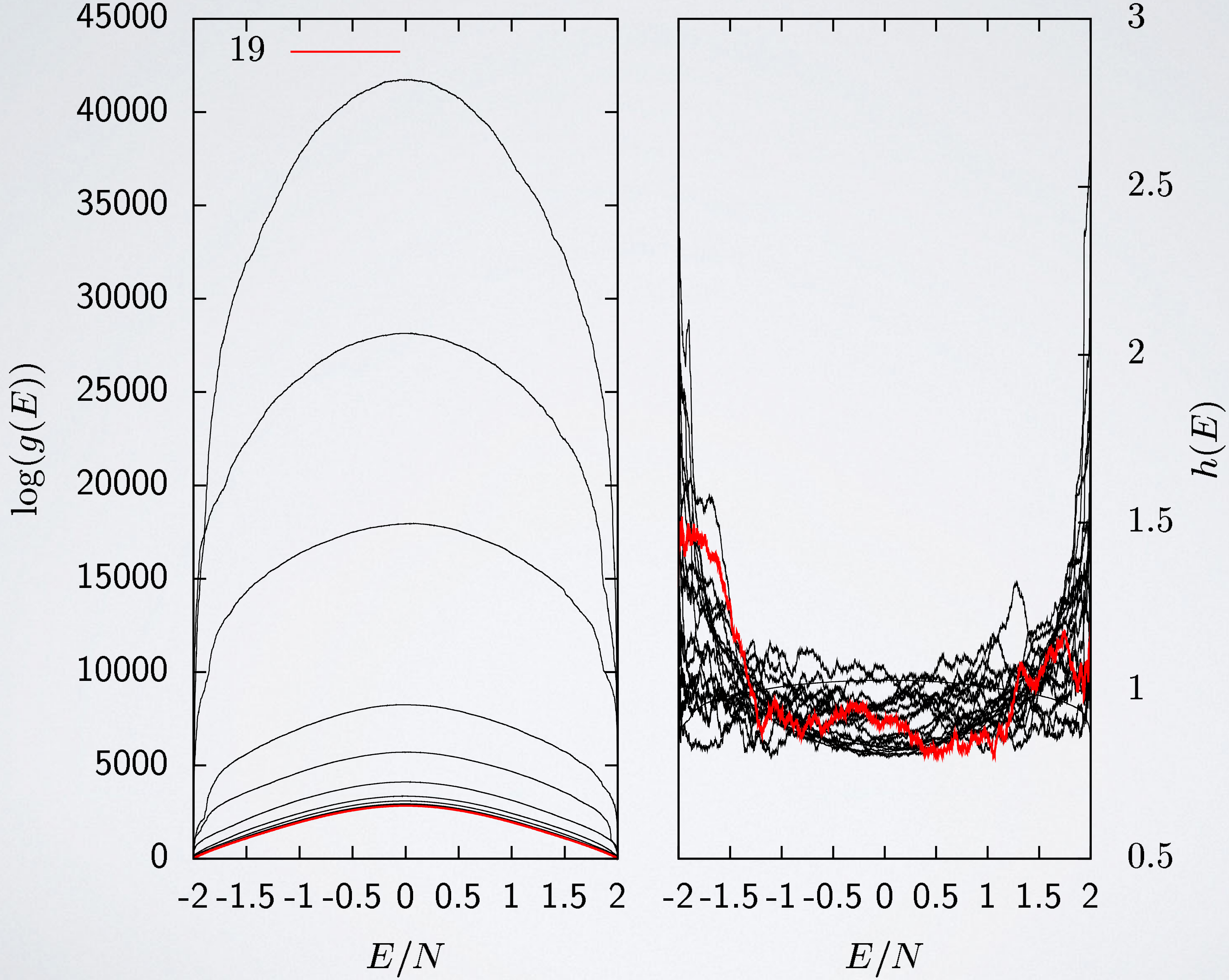
$L = 64$

Wang Landau Algorithm



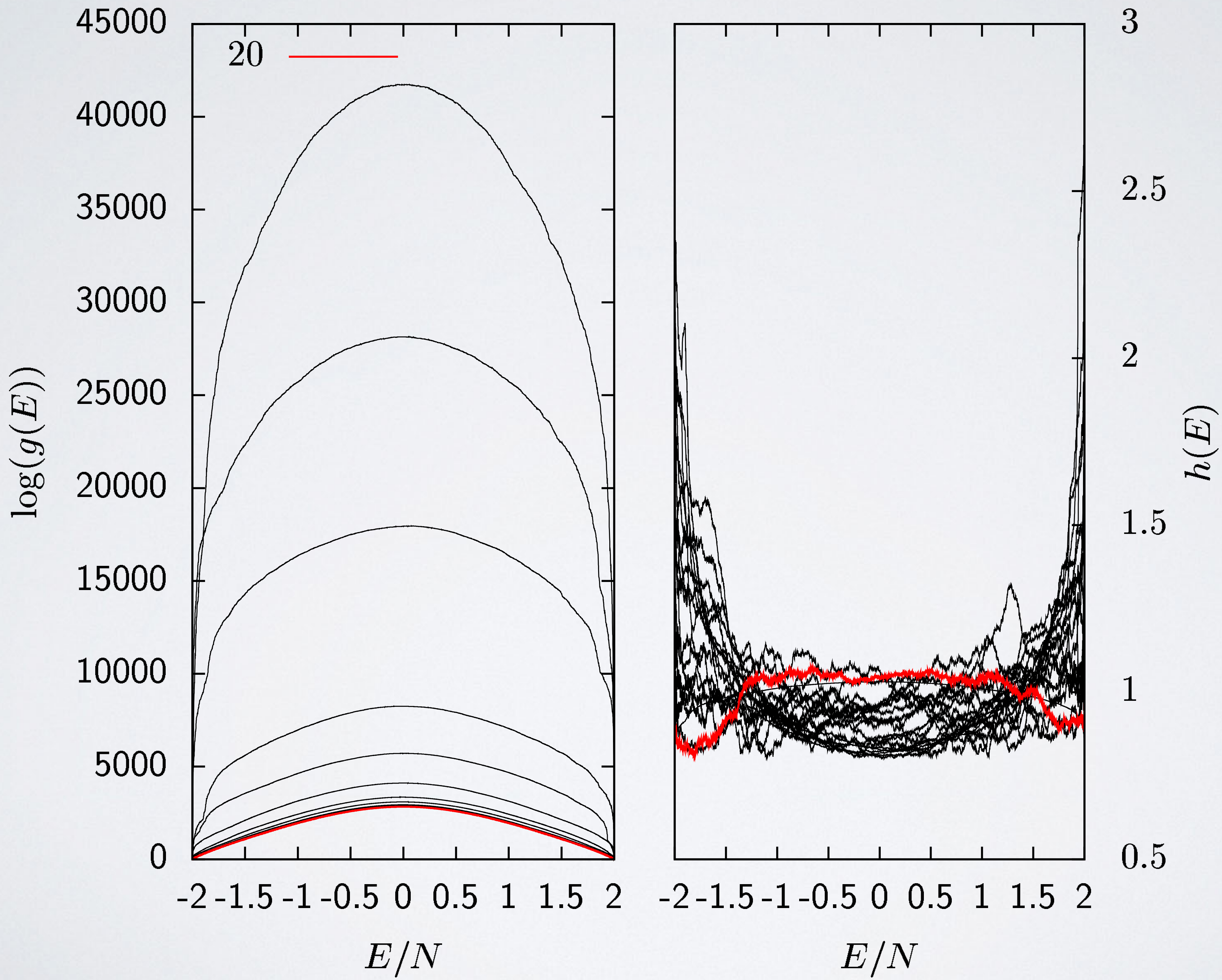
$L = 64$

Wang Landau Algorithm



$L = 64$

Wang Landau Algorithm



$L = 64$

Wang Landau Algorithm

1. Set $g(E) = 1$; choose a modification factor (e.g. $f_0 = e^1$)
2. Choose an initial state
3. Choose a site i
4. Calculate the ratio of the density of states
$$\eta = \frac{g(E_1)}{g(E_2)}$$
which results if the spin at the site i is overturned
5. Generate a random number r such that $0 < r < 1$
6. If $r < \eta$, flip the spin
7. Set $g(E_i) \rightarrow g(E_i) * f$
8. If the histogram is not *flat*, go to the next site and go to 4.
9. If the histogram is *flat*, decrease f , e.g. $f_{i+1} = f^{1/2}$
10. Repeat step 3-9 until $f = f_{\min} \sim \exp(10^{-8})$

Wang Landau Algorithm

11. ~~Calculate properties using final density of states $g(E)$~~

11. set $f = 0$, i.e. the measurement part is the same as for a multicanonical simulation
12. Calculate properties using the measurement run

$$\langle E \rangle_T = \frac{\sum_E E g(E) \exp(-\beta E)}{\sum_E g(E) \exp(-\beta E)}$$

$$C = \beta^2 (\langle E^2 \rangle - \langle E \rangle^2)/V$$

Comparison

1. run 128 independent Wang-Landau simulations

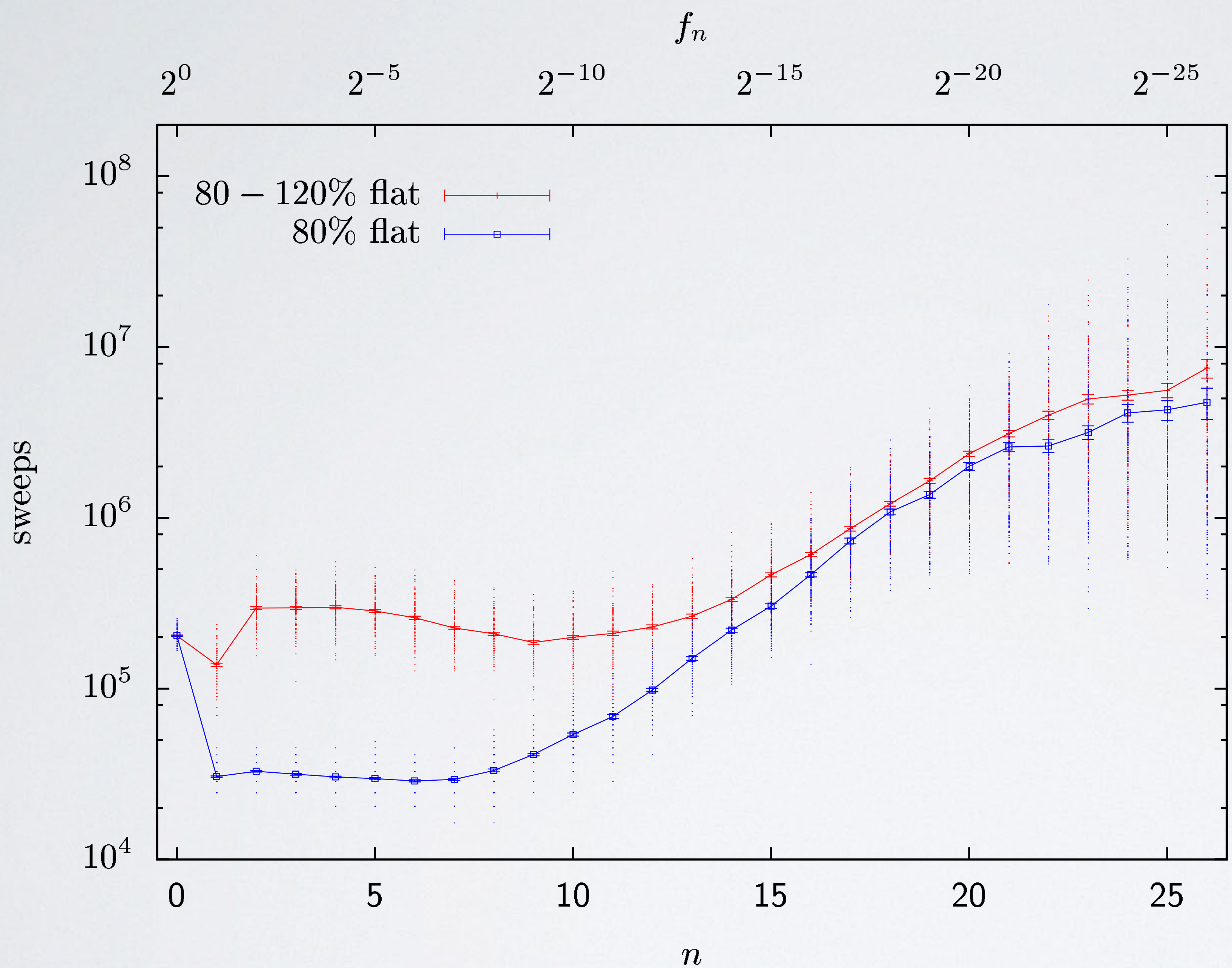
2. compute the average total number of sweeps

3. run 128 independent multicanonical simulations and stop them at the same total number of sweeps

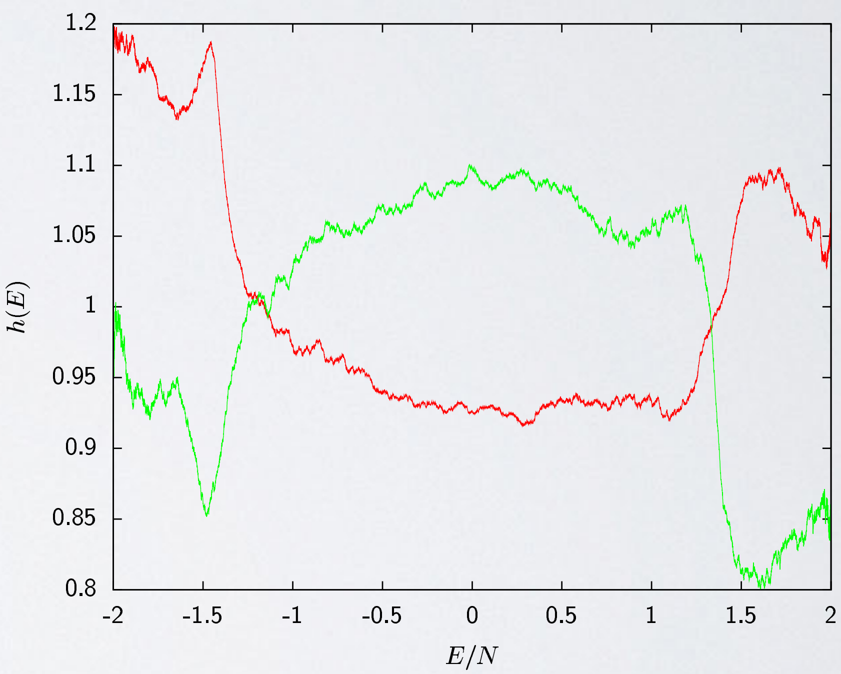
4. compute the relative errors and compete the results

$$\epsilon(X) \equiv |\frac{X_{\mathrm{sim}} - X_{\mathrm{exact}}}{X_{\mathrm{exact}}}|$$

Wang Landau Algorithm

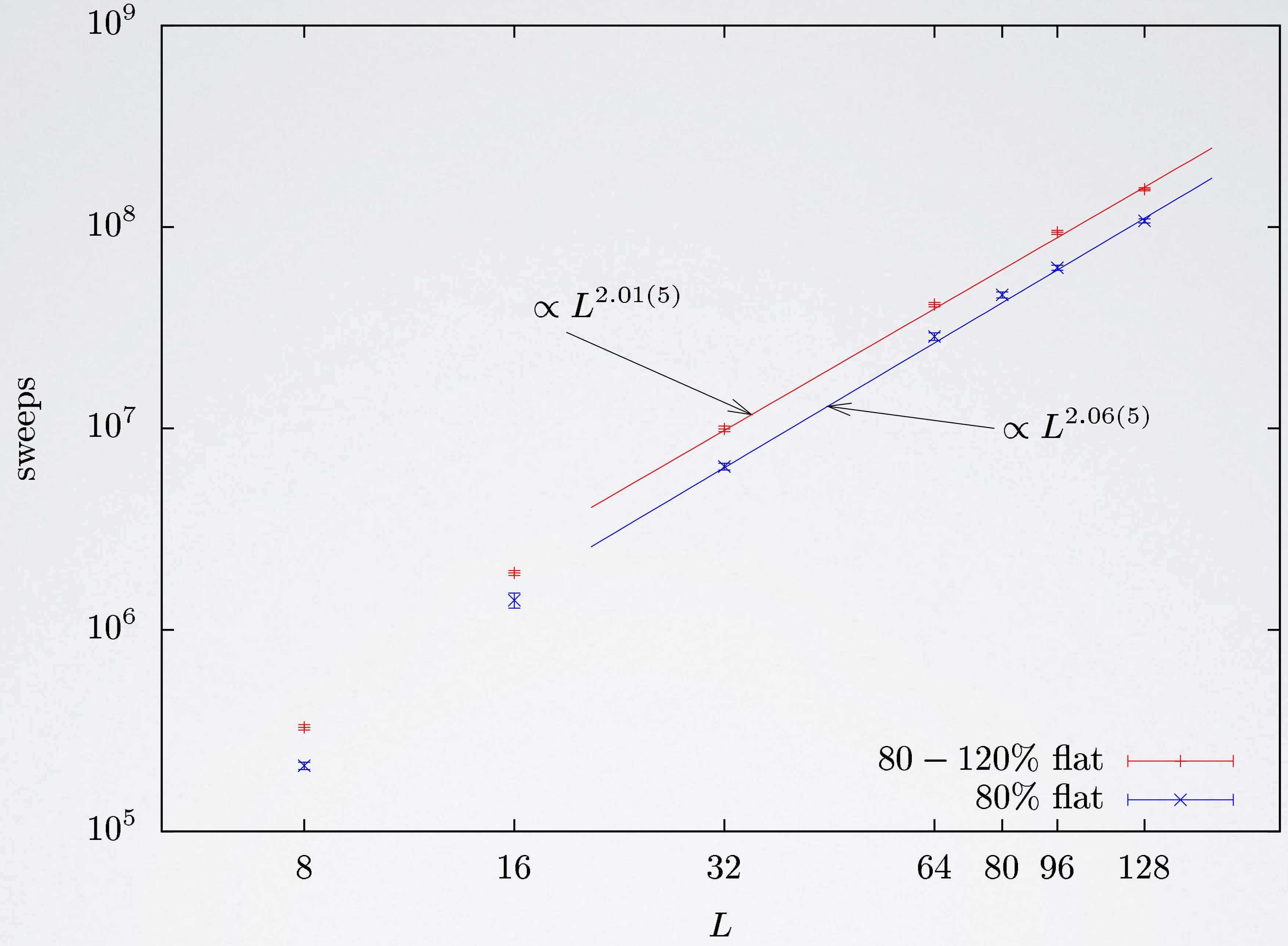


$L = 64$



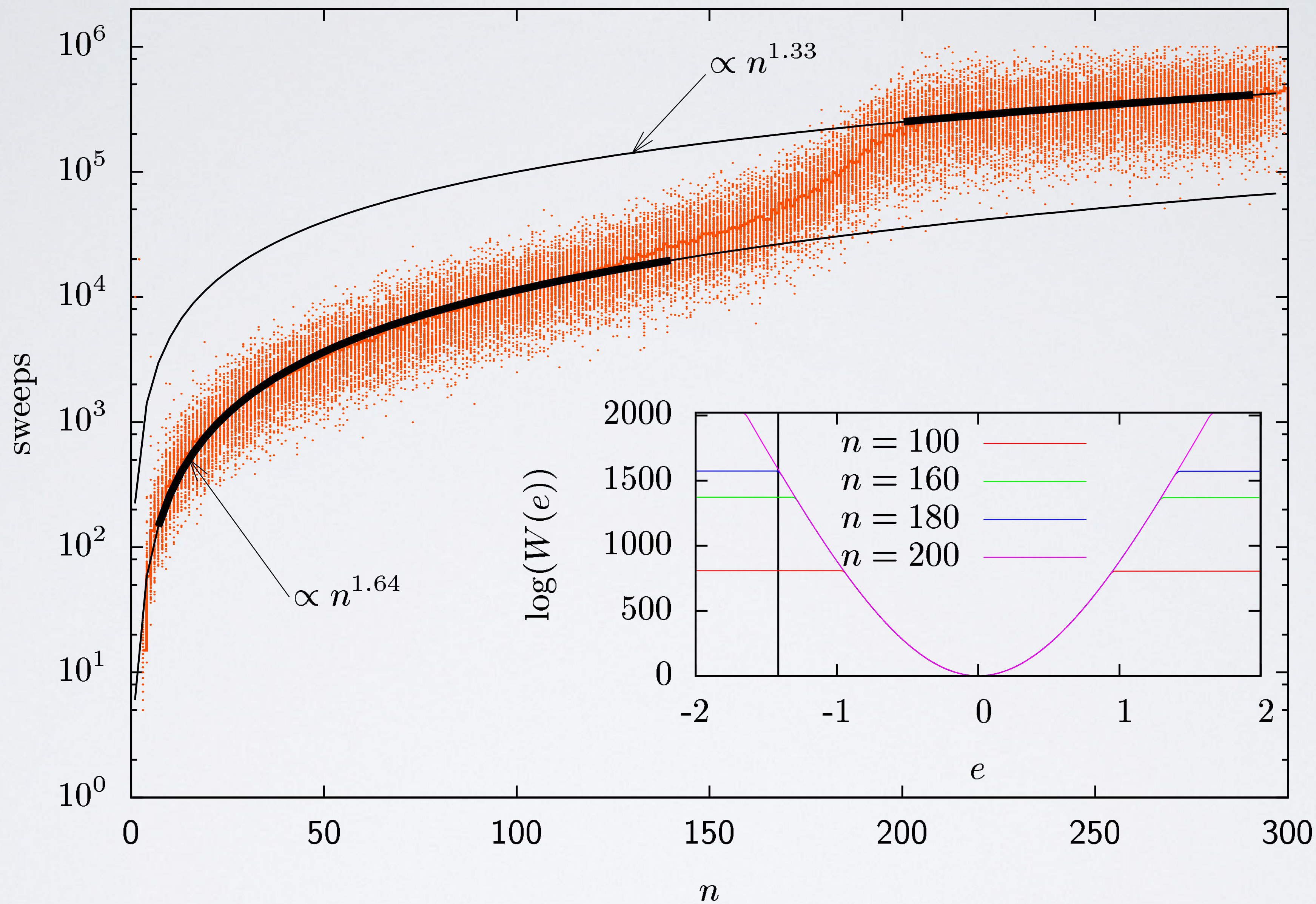
number of sweeps used in the individual iteration of 128 WL runs

Wang Landau Algorithm



Total number of sweeps used in iteration $n = 0$ to $n = 26$ of WL simulations as a function of the lattice size

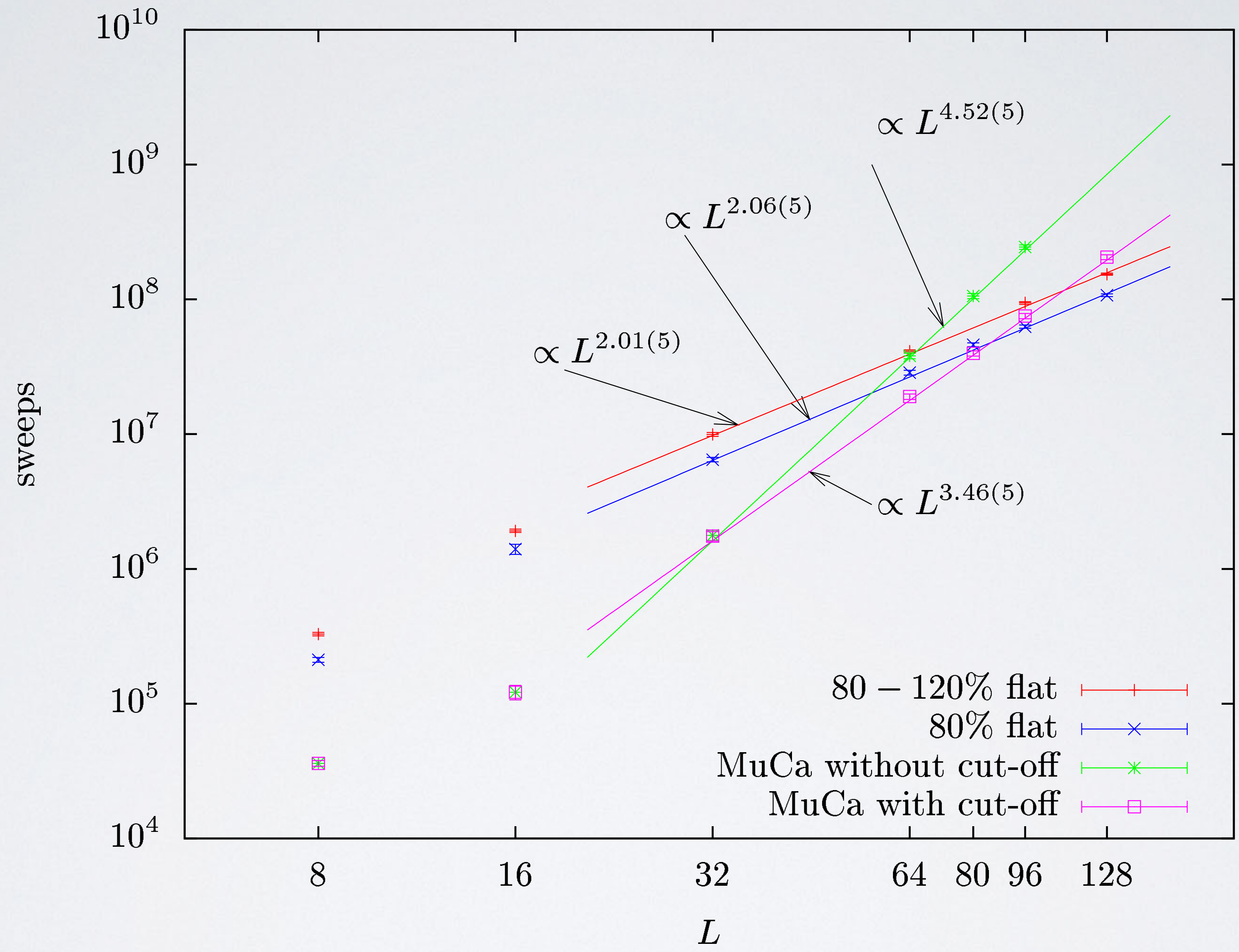
Multicanonical Monte Carlo Algorithm



$L = 64$

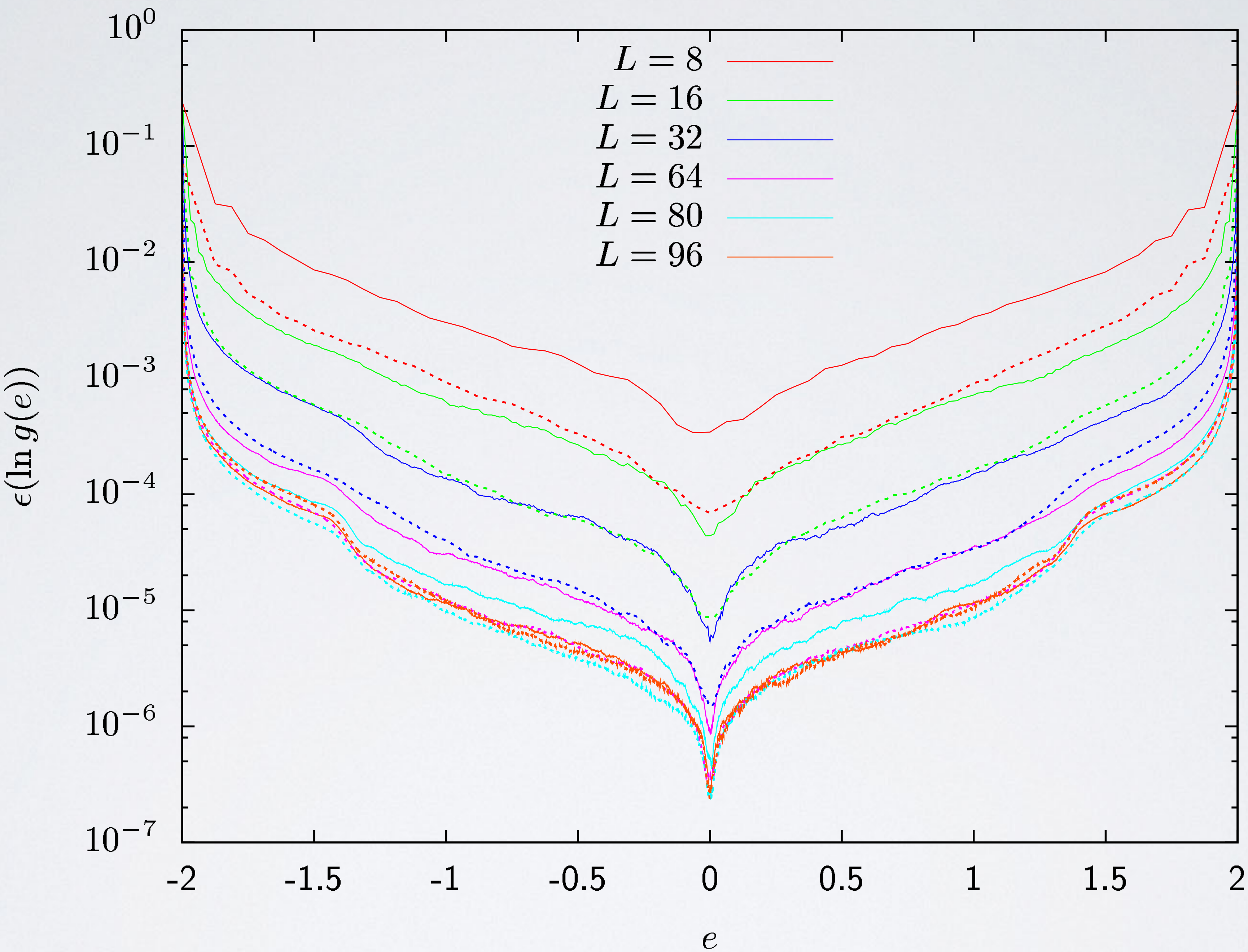
Number of sweeps used for the nth run of the accumulative recursion for 128 independent multicanonical simulations

Comparison



Total number of sweeps used to generate the final set of weights as a function of the lattice size

Comparison



The relative error using the total number of states to normalize the density of states using the 80% flatness for WL (continuous lines) and the cut-off criterion for MuCa (dotted lines) simulations.

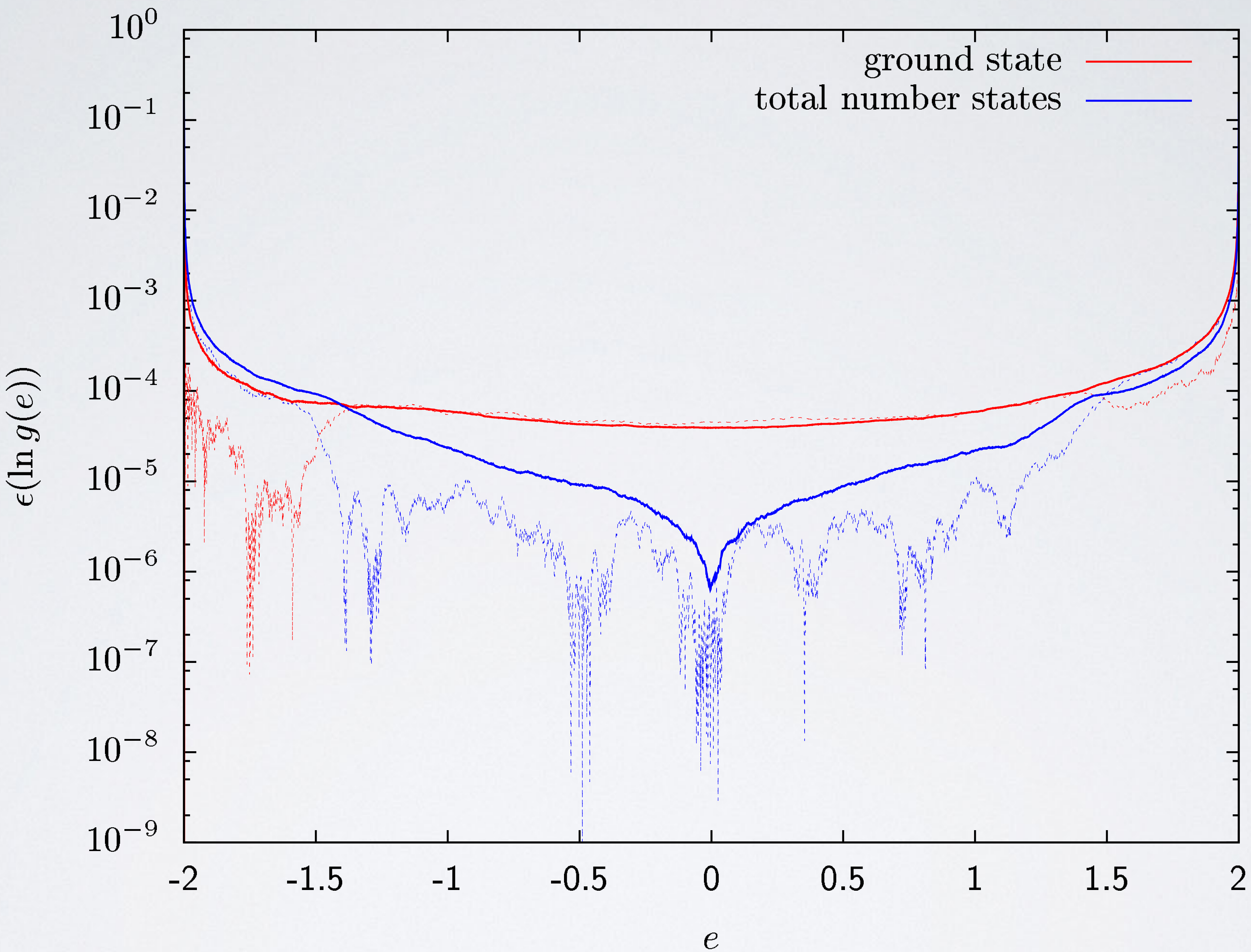
Summary

- quite a few control parameters to play with
- multicanonical algorithm faster for small system
- Wang-Landau algorithm shows better scaling

- for larger systems parallel versions are available

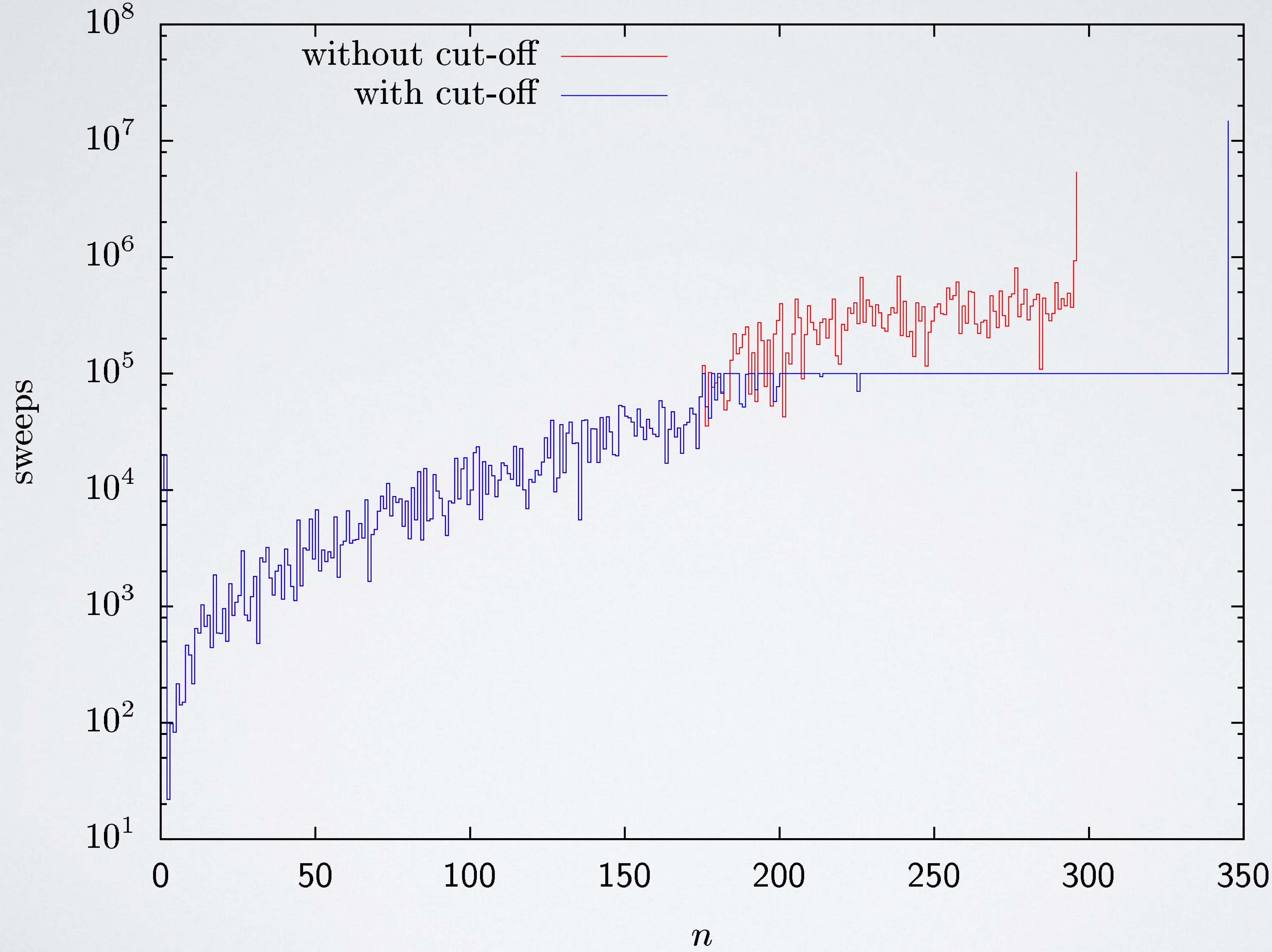
EB and W. Janke, in preparation

Discussion



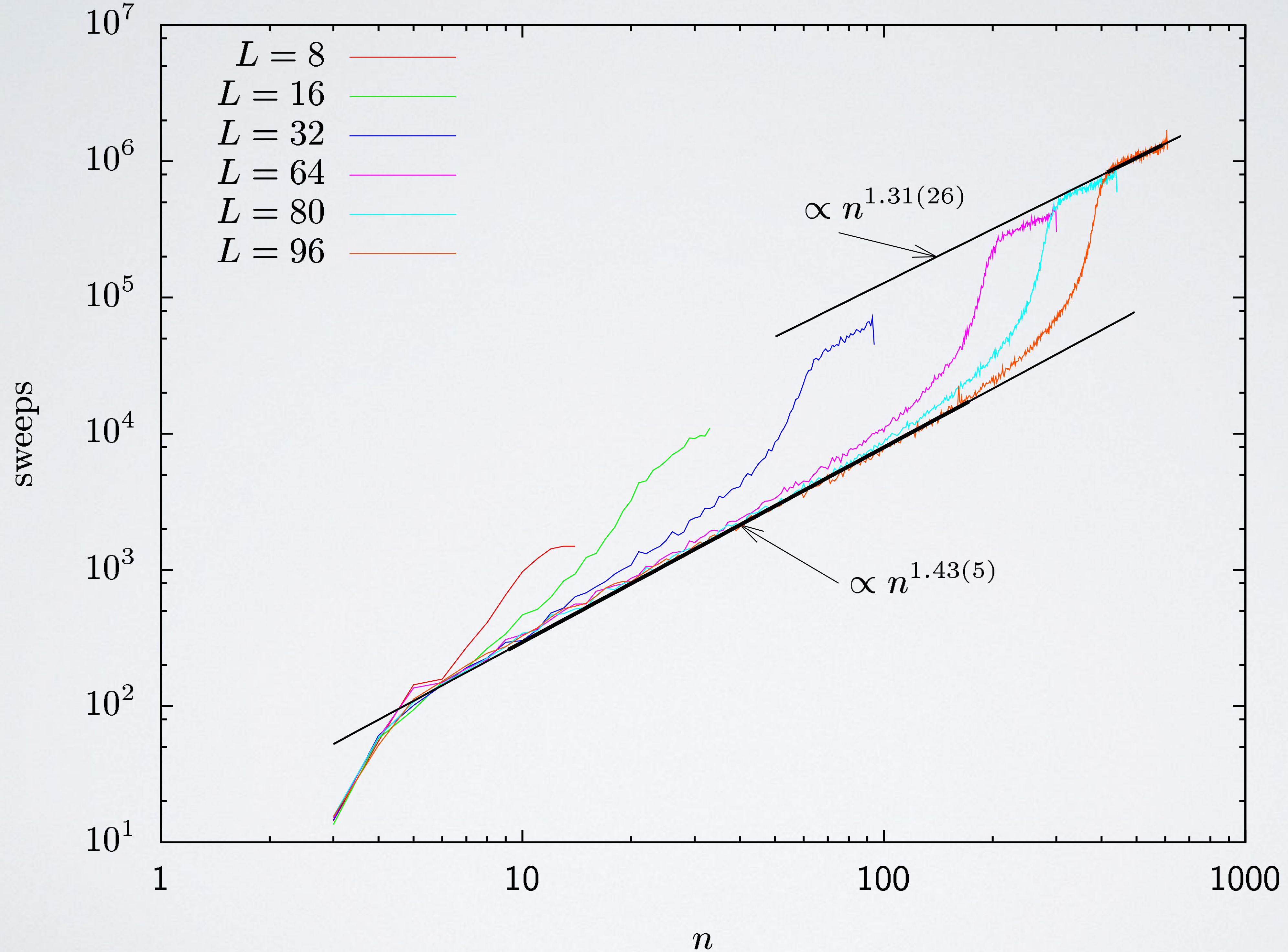
The relative error using the ground-state degeneracy or the total number of states to normalize the density of states, respectively.

Discussion



without cut-off
with cut-off
sweeps
10^8
10^7
10^6
10^5
10^4
10^3
10^2
10^1
0
50
100
150
200
250
300
350
n

Discussion



$L = 8$
$L = 16$
$L = 32$
$L = 64$
$L = 80$
$L = 96$
$\propto n^{1.31(26)}$
$\propto n^{1.43(5)}$
sweeps
n
10^1
10^2
10^3
10^4
10^5
10^6
10^7
1
10
100
1000