PY 722 Homework 1

Arvin Kushwaha

Dr. Lubos Mitas

September 24, 2024

1. Introduction

The Wang-Landau algorithm [1] is one of a collection of algorithms categorized as "flat-histogram algorithms." In contrast to the very commonly used Metropolis-Hastings Monte Carlo state sampling method that attempts to recreate the state distribution of a system at some given temperature T, we have an iteratively refined estimate of the **density of states** (DOS), which is independent of temperature. This provides the notable advantage of being able to capture critical phenomena and complex behavior without having to perform simulations across multiple T parameters. Suppose we have a system S, with configuration space Ω and energies $E : \Omega \to [E_{\min}, E_{\max}]$. We choose $E_{\min}, E_1, E_2, ..., E_{n-1}, E_{\max}$ such that $E_{\min} = E_0 < E_1 < E_2 < \cdots < E_{n-1} < E_n = E_{\max}$. This gives us n bins of energies: $[E_0, E_1], [E_1, E_2], ..., [E_{n-2}, E_{n-1}], [E_{n-1}, E_n]$, where we denote $B_i = [E_{i-1}, E_i]$. The algorithm generally proceeds as follows:

Wang-Landau (Vanilla)

---

1. Initialize the energy histogram, $\forall i, H_i = 0$, and the entropy (i.e., log(density of states)) $\forall i, S_i = 0$, where H_i records the number of times a visit to a state with energy in B_i has occurred and S_i is the current estimate of the entropy for the energy range B_i.

   Initialize a starting state to walk from: $x_0 \in \Omega$ and let our iteration $k = 0$. Initialize $f = 0$. Additionally, choose some tolerance ε. A common choice for this is $\varepsilon = 1 \times 10^{-8}$.

2. Generate a state transition $x_k \to x'_k$ (with some specific properties). If $E(x_k) \in B_a$ and $E(x'_k) \in B_b$, then let the acceptance probability $P = \min(1, \exp(S_a - S_b))$. Let r be a uniform random number in $[0, 1)$, if $r < P$, then let $x_{k+1} = x'_k$, otherwise, let $x_{k+1} = x_k$.

3. If $E(x_{k+1}) \in B_i$, $H_i \leftarrow H_i + 1$ and $S_i \leftarrow S_i + f$.

4. If the histogram H is "flat," then we set $f \leftarrow f/2$ and reset the histogram to zero: $\forall i, H_i = 0$. If $f < \varepsilon$, then terminate the algorithm, the S_i at the current state represents the log(density of states).

   For computing "flatness", the test

   $$\frac{\max(H_i) - \min(H_i)}{\frac{1}{n}\sum_i H_i} < p$$

   is often used, where a common value for p is 0.05. Adjusting p lets you trade convergence time with accuracy.

5. Let $k \leftarrow k + 1$, return to step 2.

There is some significant freedom in how this algorithm can be implemented to reduce the computational expense of walking in the state space. An arbitrary proposal algorithm can be used. Consider a proposal probability function $p(x_1 \to x_2)$. Then, we can adjust the Wang-Landau algorithm as follows, in step 2, let the acceptance probability instead be:

$$P = \min\left(1, \exp(S_a - S_b)\frac{p(x_2 \to x_1)}{p(x_1 \to x_2)}\right).$$

In the pleasant case of a symmetric proposal, like flipping a random spin for an Ising model, this additional factor can be neglected, as $p(x_1 \to x_2) = p(x_2 \to x_1)$, but in general this is not the case. Notice however, that this freedom of choice gives us the power to apply more "exotic" proposal distributions.

2. Implementation and Application

With the results of the Wang-Landau algorithm, the density of states: $g(E) = \exp(S(E))$, the construction of the partition function is rather straightforward and leads directly to the free energy, internal energy, and heat capacity:

$$\begin{aligned}
Z &= \sum_{s\in\Omega} e^{-\beta E(s)} \propto \sum_E e^{-\beta E} g(E) \\
F &= -\frac{1}{\beta}\ln Z = -\frac{1}{\beta}\ln\left(\sum_E e^{-\beta E} g(E)\right) \\
U &= -\frac{\partial \ln Z}{\partial \beta} = \sum_E E e^{-\beta E} g(E) \\
C_V &= \frac{\partial U}{\partial T} = -\beta^2 \frac{\partial U}{\partial \beta} = \beta^2 \frac{\partial^2 \ln Z}{\partial \beta^2} = \beta^2 \sum_E E^2 e^{-\beta E} g(E)
\end{aligned}$$

Because there are numerous derivatives here, we can take advantage of the JAX library [2], which provide excellent compositional transformations and JIT-ing utilities. In the following code, `log_dos` is the S (microcanonical entropy) and `energies` are the energies of the bins. The following are short samples of code, with some modifications for readability. The full code can be found on GitHub here: https://github.com/ArvinSKushwaha/wang-landau-analysis.

```
@jax.jit
def partition_function(beta: float, log_dos: Array, energies: Array):
    return np.sum(np.exp(log_dos - energies * beta))
```

The `partition_function` method is just a direct implementation of the equation for Z above.

```
@jax.jit
def internal_energy(beta: float, log_dos: Array, energies: Array):
    min_energy = energies.min()
    return -jax.grad(
        lambda beta: np.log(partition_function(
            beta,
            log_dos,
            energies - min_energy
        ))
    )(beta) + min_energy
```

Here, we used the `jax.grad` function that takes a function and returns another function that is the derivative of the function. In other words, `jax.grad` : $f \mapsto f'$. Notice the modification of `energies` to `energies - energies.min()`. This is for the purposes of numerical stability, to prevent ∞ from arising.

```
@jax.jit
def free_energy_beta(beta: float, log_dos: Array, energies: Array):
    min_energy
    return -np.log(partition_function(
        beta,
        log_dos,
        energies - min_energy
    )) + min_energy * beta
```

Unfortunately, since F diverges to $-\infty$ as $\beta \to 0$ (at least for models like the 2-D Ising Model), it's much nicer to plot βF, which is what the `free_energy_beta` method computes. Again, this is a direct implementation of the equation above.

```
def d_dT(func_of_beta: Callable[[float, Any, Any], Any]):
    return lambda beta, *args: -(beta**2) * jax.grad(func_of_beta)(beta, *args)

@jax.jit
def heat_capacity(beta: float, log_dos: Array, energies: Array):
    return d_dT(internal_energy_)(beta, log_dos - log_dos.min(), energies)
```

Here, the `d_dT` function takes the place of `jax.grad`, and effectively implements the following identity:

$$\frac{\partial f}{\partial T} = -\beta^2 \frac{\partial f}{\partial \beta}.$$

We implement it because heat capacity (C_V) is most cleanly represented as $\frac{\partial U}{\partial T}$. Here, we substitute `log_dos` with `log_dos - log_dos.min()` for numerical stability.

3. Analysis on the Periodic Ising Model

We analyze the periodic 2-dimensional Ising model with even side length (N) given by the following Hamiltonian and configuration space:

$$\Omega = \{-1, +1\}^{N^2} \Rightarrow |\Omega| = 2^{N^2}$$

$$\mathcal{H}(s) = -J \sum_{i=1}^{N} \sum_{j=1}^{N} s_{i,j}\left(s_{i+1,j} + s_{i,j+1}\right), \text{where } s \in \Omega, s_{i,j} = s_{i+N,j} = s_{i,j+N} \in \{-1, +1\}$$

For the state-space walker in the Wang-Landau algorithm, we use the naive step of flipping a spin at a random site on the lattice. This trivially satisfies detailed balance.

In Figure 1, we can confirm that the primary expected characteristics of the periodic Ising model density of states are demonstrated, for example the symmetry and approximately paraboloid shape of the curve. The correctness of the converged DOS can be further confirmed by the characteristics of the derived quantities in Figure 2, Figure 3, and Figure 4. We find that Figure 2 exactly matches the expected internal energy presented in [3] and Figure 3 does also, albeit with a vertical offset that is likely a consequence of the choice of normalization for the partition function. Finally, and most relevantly is the estimation of the critical temperature from the heat capacity (C_V) of the system. In Figure 4, we see that as the size of the lattice increases, the peak of the heat capacity, which is the estimator for the critical temperature

becomes a better approximation to the expected critical temperature from the Onsager solution: β_{ONS}. Thus, the Wang-Landau algorithm has done a fairly effective job of computing the thermodynamic properties of the 2D periodic Ising model, without having had to simulate the model at every temperature of relevance.

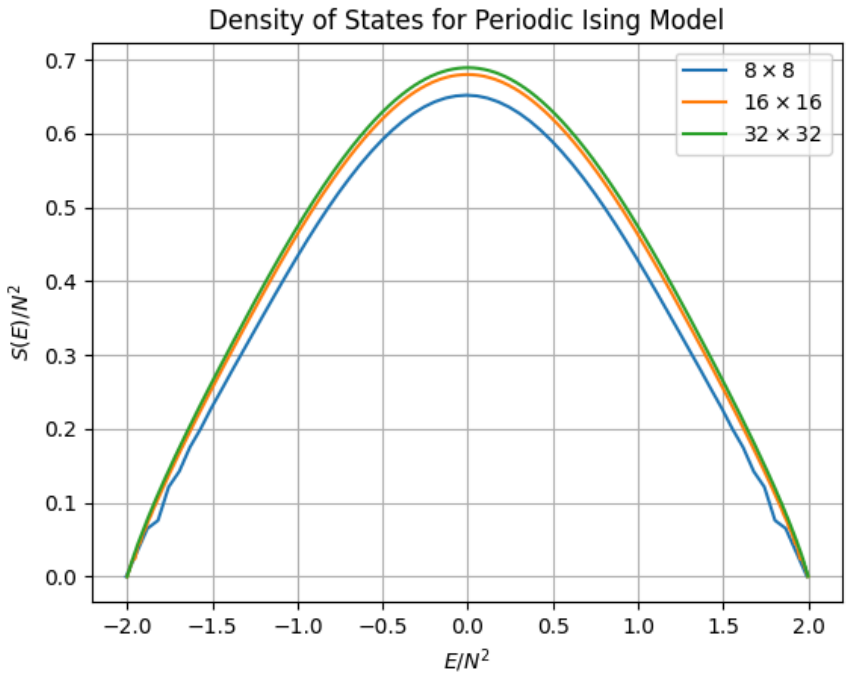

Figure 1: The rescaled density of states for a 2-dimensional Ising model with periodic boundary conditions and varying side lengths computed with the Wang-Landau algorithm.

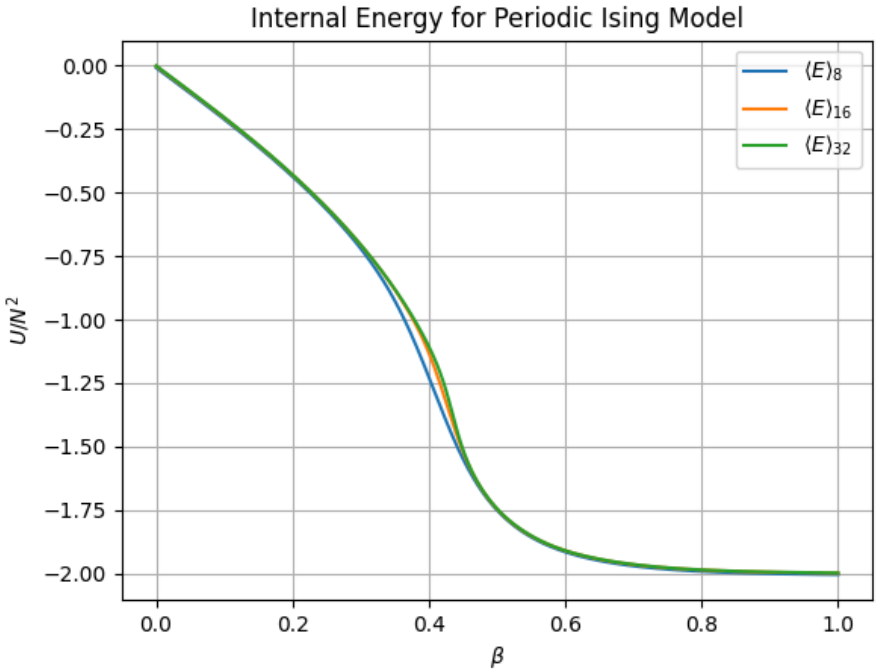

Figure 2: The rescaled internal energy for a 2-dimensional Ising model with periodic boundary conditions and varying side lengths computed with the Wang-Landau algorithm.

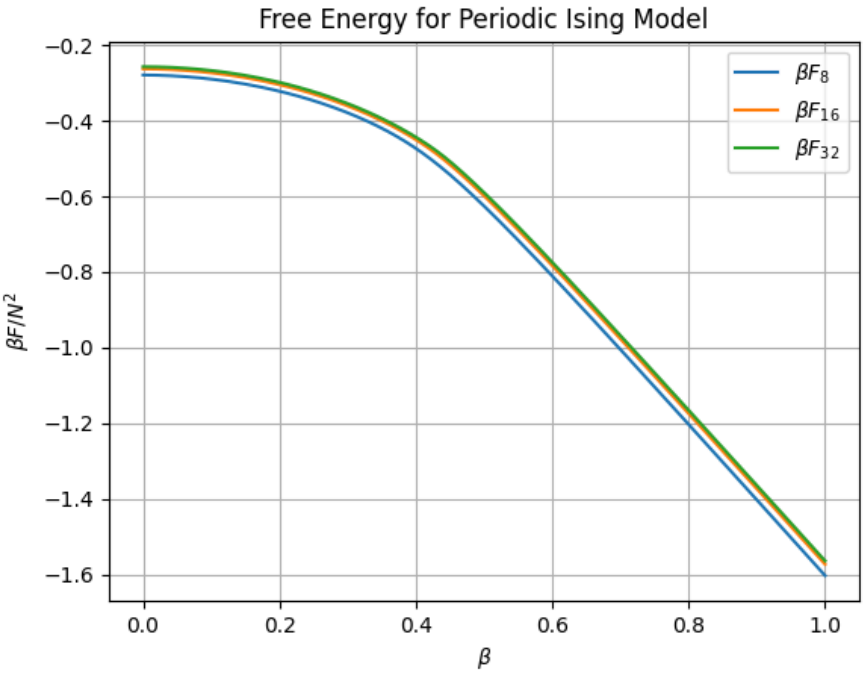

Figure 3: The rescaled free energy for a 2-dimensional Ising model with periodic boundary conditions and varying side lengths computed with the Wang-Landau algorithm.

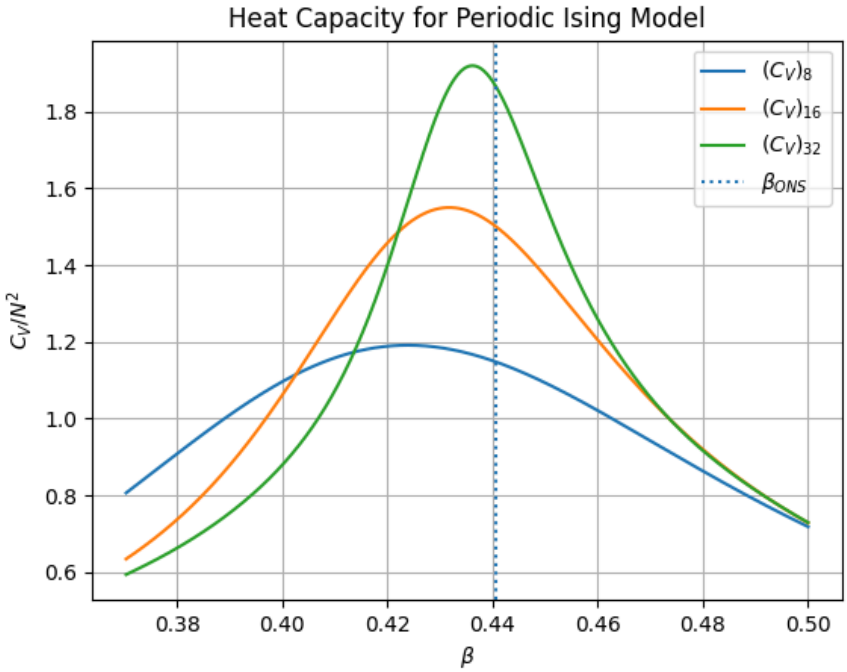

Figure 4: The rescaled heat capacity for a 2-dimensional Ising model with periodic boundary conditions and varying side lengths computed with the Wang-Landau algorithm. It can be seen that the peak of the heat capacity curve is converging towards β_{ONS}, the expected critical temperature.

Bibliography

[1] "Determining the density of states for classical statistical models: A random walk algorithm to produce a flat histogram," *Phys. Rev. E*, vol. 64, no. 5, Oct. 2001, doi: 10.1103/PhysRevE.64.056101.

[2] "JAX: composable transformations of Python+NumPy programs." 2018. [Online]. Available: http://github.com/jax-ml/jax

[3] "The Analytical Expressions for a Finite-Size 2D Ising Model." 2017. [Online]. Available: https://arxiv.org/abs/1706.02541