The Quantum Monte Carlo Simulation of Transverse Field Ising Model

Yuanxing Duan, Shixin Hu and Fangyu Xiong

School of Physics, Peking University

January 5, 2021

Outline

1 Introduction to Transverse Field Ising

2 Worm Algorithm

3 C++ Implement

4 Numerical Results

Classical Ising Model

Hamiltonian:

$$\mathcal{H} = -t \sum_{\langle ij \rangle} \sigma_i^z \sigma_j^z - h_z \sum_i \sigma_i^z$$

Thermal Fluctuation $\Longrightarrow$ Thermal(Classical) Phase Transition

Low T: Ordered state

High T: Paramagnetic state

Exponent	Definition	Ising Value
α	$C \propto (T - T_c)^{-\alpha}$	0
β	$M \propto (T_c - T)^{\beta}$	1/8
γ	$\chi \propto (T - T_c)^{-\gamma}$	7/4
δ	$M \propto h^{1/\delta}$	15
ν	$\xi \propto (T - T_c)^{-\nu}$	1
η	$\Gamma(n) \propto \lvert n \rvert^{2-d-\eta}$	1/4

Critical exponents of Classical Ising Model

Quantum Ising Model

Hamiltonian of Transverse Field Ising model (TFI):

$$\mathcal{H} = -t \sum_{\langle ij \rangle} \sigma_i^z \sigma_j^z - h \sum_i \sigma_i^x$$

$[\sigma_i^z, \sigma_i^x] \neq 0 \Longrightarrow$ Quantum Fluctuation!

Quantum Phase Transition:

$h \ll t$: Spins aligned in z-direction.

$h \gg t$: Spins aligned in x-direction.

Theoretical Calculation

Jordan-Wigner tranformation:

$$\sigma_i^x = 1 - 2c_i^\dagger c_i$$

$$\sigma_i^z = -\prod_{j<i}(1 - 2c_j^\dagger c_j)(c_i + c_i^\dagger)$$

c_is are fermion operators.

Hamiltonian becomes:

$$\mathcal{H} = -t\sum_i (c_i^\dagger c_{i+1} + c_{i+1}^\dagger c_i + c_i^\dagger c_{i+1}^\dagger + c_{i+1} c_i - 2g c_i^\dagger c_i + g)$$

with $g = h/t$

Theoretical Calculation

Solve it by Bogoliubov transformation:

$$\gamma_k = u_k c_k - v_k c^\dagger_{-k}$$

$$\gamma_k^\dagger = u_k c_k^\dagger - v_k c_{-k}$$

with c_k Fourier transformation of c_i. And

$$u_k = \cos\left(\frac{\theta_k}{2}\right), v_k = \sin\left(\frac{\theta_k}{2}\right)$$

$$\tan\left(\theta_k\right) = \frac{\sin(ka)}{g - \cos(ka)}$$

Theoretical Calculation

Switch to Majorana representation (continuous limit):

$$\psi(x_i) = -i(c_i - c_i^\dagger) = \frac{1}{2N}\sum_k (u_1(k)\gamma_k^\dagger e^{ikx_i} + h.c.)$$

$$\bar{\psi}(x_i) = c_i + c_i^\dagger = \frac{1}{2N}\sum_k (u_2(k)\gamma_k^\dagger e^{ikx_i} + h.c.)$$

$$\psi(x_i, t) = e^{i\mathcal{H}t}\psi(x_i)e^{-i\mathcal{H}t}, \bar{\psi}(x_i, t) = e^{i\mathcal{H}t}\bar{\psi}(x_i)e^{-i\mathcal{H}t}$$

$\Longrightarrow$ Solution of 1+1D Dirac equation

Theoretical Calculation

The action:

$$S = \frac{i}{2} \int \psi(\partial_0 - \partial_1)\psi + \bar{\psi}(\partial_0 + \partial_1)\bar{\psi} + (1-g)\psi\bar{\psi}$$

$g = 1 \Longrightarrow$ masless, scale invariance , i.e.

$$S' = S \text{ when } x \to \lambda x$$

$\Longrightarrow$ Critical Point at $h = t$!!

The two point correlation functions at criticality:

$$\langle \sigma_i^z \sigma_{i+n}^z \rangle \propto \frac{1}{|n|^{1/4}}$$

$$\langle \sigma_i^x \sigma_{i+n}^x \rangle \propto \frac{1}{|n|^2}$$

Outline

1 Introduction to Transverse Field Ising

2 Worm Algorithm

3 C++ Implement

4 Numerical Results

Holstein-Primakoff Transformation

TFI:

$$\mathcal{H} = -t\sum_{\langle ij\rangle}\sigma_i^x\sigma_j^x - h\sum_i \sigma_i^z = K + U$$

Hopping and pairing terms:

$$K = K_1 + K_2 = -t\sum_{\langle ij\rangle}(\sigma_i^+\sigma_j^- + \text{h.c.}) - t\sum_{\langle ij\rangle}(\sigma_i^+\sigma_j^+ + \text{h.c.})$$

HP transformation: $b_i(b_i^\dagger) = \sigma_i^+(\sigma_i^-), n_i = b_i^\dagger b_i = (\sigma_i^z + 1)/2$

$$\Rightarrow \mathcal{H} = -t\sum_{\langle ij\rangle}(b_i^\dagger b_j + \text{h.c.}) - t\sum_{\langle ij\rangle}(b_i^\dagger b_j^\dagger + \text{h.c.}) - \mu\sum_i n_i$$

where $\mu = 2h$.

$\mathcal{Z}$ Configurations

$$
\begin{aligned}
\mathcal{Z} &= \mathrm{Tr}\left(\mathrm{e}^{-\beta\mathcal{H}}\right) = \sum_{a_0} \langle\alpha_0|\,\mathrm{e}^{-\beta\mathcal{H}}\,|\alpha_0\rangle \\
&= \lim_{\mathrm{d}\tau=\frac{\beta}{n},n\to\infty} \sum_{\{\alpha_i\}} \langle\alpha_0|\,\mathrm{e}^{-\mathcal{H}\mathrm{d}\tau}\,|\alpha_{n-1}\rangle\cdots\langle\alpha_i|\,\mathrm{e}^{-\mathcal{H}\mathrm{d}\tau}\,|\alpha_0\rangle \\
&= \sum_{\alpha_0}\sum_{\mathcal{N}}^{\infty}\int_0^{\beta}\int_{\tau_1}^{\beta}\cdots\int_{\tau_{\mathcal{N}-1}}^{\beta}\prod_{k=1}^{\mathcal{N}}\mathrm{d}\tau_k\,F(t,h)
\end{aligned}
$$

$$
F(t,h) = t^{\mathcal{N}_h+\mathcal{N}_p}\mathrm{e}^{-\int_0^{\beta}U(\tau)\,\mathrm{d}\tau}
$$

$\mathcal{N}_h$ and $\mathcal{N}_p$: number of hopping and pairing kinks. $\mathcal{N}_h+\mathcal{N}_p=\mathcal{N}$.

$\mathcal{Z}$ configuration: several loops of spin-up (and kinks) in the $(d+1)$D space-time.
Statistical weight of a $\mathcal{Z}$ configuration:

$$
W_{\mathcal{Z}}(t,h) = \prod_{k=1}^{\mathcal{N}}\mathrm{d}\tau\,F(t,h)
$$

$\mathcal{G}$ Configurations

$$\mathcal{G}(\boldsymbol{x}_I, \tau_I; \boldsymbol{x}_M, \tau_M) = \mathrm{Tr}\left[T_\tau\left(\sigma_I^x(\tau_I)\sigma_M^x(\tau_M)\mathrm{e}^{-\beta\mathcal{H}}\right)\right]$$

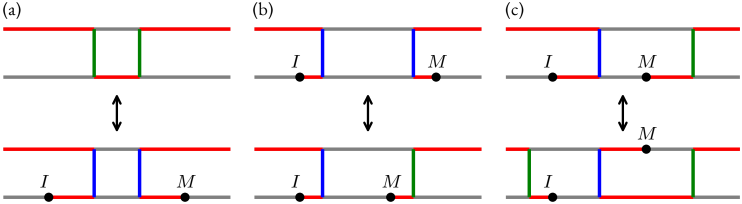
$\mathcal{G}$ configuration: $\mathcal{Z}$ configuration + an open path of spin-up with two ending points "Ira" (I) and "Masha" (M).

Statistical weight of a $\mathcal{G}$ configuration:

$$W_{\mathcal{G}} = \frac{\mathrm{d}\tau_I \mathrm{d}\tau_M}{\omega_G} \prod_{k=1}^{\mathcal{N}} \mathrm{d}\tau_k \, F(t, h)$$

- Key idea: move the defect M of the open path to produce different $\mathcal{G}$ and $\mathcal{Z}$ configurations for sampling. (Like a worm wriggling!)

Updates



(a) Create/annihilate defects I and M.

$$\frac{d\tau_I}{\beta N}\frac{d\tau_M}{\tau_a}\cdot W_{\mathcal{Z}}\cdot\mathcal{P}_{\text{crea}} = \mathcal{A}_a\cdot W_{\mathcal{G}}\cdot\mathcal{P}_{\text{anni}} \Rightarrow \begin{cases} P_{\text{crea}} = \min\left[1, \mathcal{A}_a\tau_a\frac{\beta N}{\omega_G}\frac{F_{\text{new}}}{F_{\text{old}}}\right] \\ P_{\text{anni}} = \min\left[1, \frac{1}{\mathcal{A}_a\tau_a}\frac{\omega_G}{\beta N}\frac{F_{\text{new}}}{F_{\text{old}}}\right] \end{cases}$$

(b) Move imaginary time of defect M.

$$P_{\text{move}} = \min\left\{1, \frac{F_{\text{new}}}{F_{\text{old}}}\right\}$$

(c) Insert/delete a kink.

$$\mathcal{A}_c\cdot\frac{1}{z_d}\cdot\frac{d\tau}{\tau_c}\cdot W\cdot\mathcal{P}_{\text{inse}} = \mathcal{A}_c\cdot\frac{1}{z_d}\cdot\frac{1}{n_k}\cdot W_+\cdot\mathcal{P}_{\text{dele}} \Rightarrow \begin{cases} P_{\text{inse}} = \min\left[1, \frac{\tau_c}{n_k+1}\frac{F_{\text{new}}}{F_{\text{old}}}\right] \\ P_{\text{dele}} = \min\left[1, \frac{n_k}{\tau_c}\frac{F_{\text{new}}}{F_{\text{old}}}\right] \end{cases}$$

τ_a, τ_b, τ_c: ranges for choosing random time displacement.
A priori probabilities: $\mathcal{A}_a + \mathcal{A}_b + 2\mathcal{A}_c = 1$.

Outline

1 Introduction to Transverse Field Ising

2 Worm Algorithm

3 C++ Implement

4 Numerical Results

The Grid

- Use one bit to represent one spin
- Use one bit to represent one kink between two world lines
- Use bitwise **XOR** with mask to flip spins and calculate the correlation
- Use **OR** to get the state of one spin
- Use bitwise **OR** to get the positions without kinks
- Use **__popcnt** to count spin-up number
- Use **_tzcnt_u32** to find the first set bit to find the starting point of one loop in the grid

Warpping Number of Loops

- Depth first search
- Must traverse the whole grid to find all loops and calculate the warpping number of each loop
- This doesn't affect the performance since its time consumption is about 1% of creating and annihilating once
- Use a direction to make sure that the current point donsn't goes it way back
- Mark the spins "walked" to avoid traversing one loop twice

Miscellaneous

- Use OpenGL to implement visualization
- Reached 1000 times creating and annihilating per second under grid size N=64 with single thread
- For $R_\downarrow$, we can simply change the sign of h

Outline

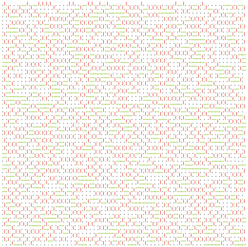
1 Introduction to Transverse Field Ising

2 Worm Algorithm

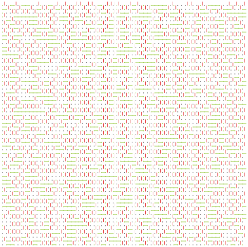
3 C++ Implement

4 Numerical Results

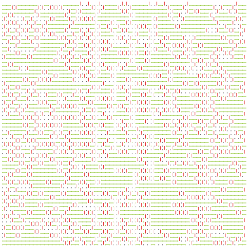
Visualization



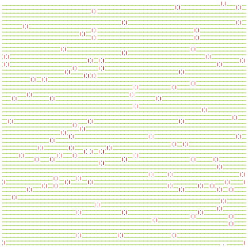
(a) $h = 0.01$.



(b) $h = 0.5$.



(c) $h = 1.0$.



(d) $h = 1.5$.

Calculated Quantities

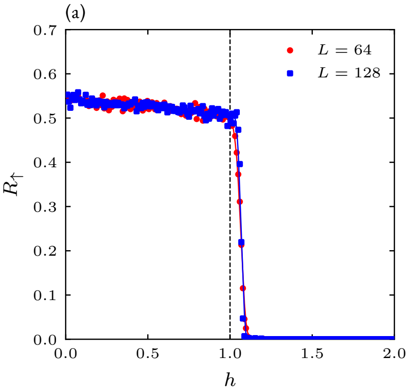
- Topological quantities: winding number $W^l \Rightarrow$ total winding number $W \Rightarrow$ wrapping probability R.
- Physical quantities: correlation functions.

$$\overline{\sigma_i^z} \equiv \frac{1}{L^2} \sum_{j,k} \sigma_{j,k}^z$$

$$\overline{\sigma_i^z \sigma_{i+n}^z} \equiv \frac{1}{L^2} \sum_{j,k} \sigma_{j,k}^z \sigma_{j+n,k}^z$$

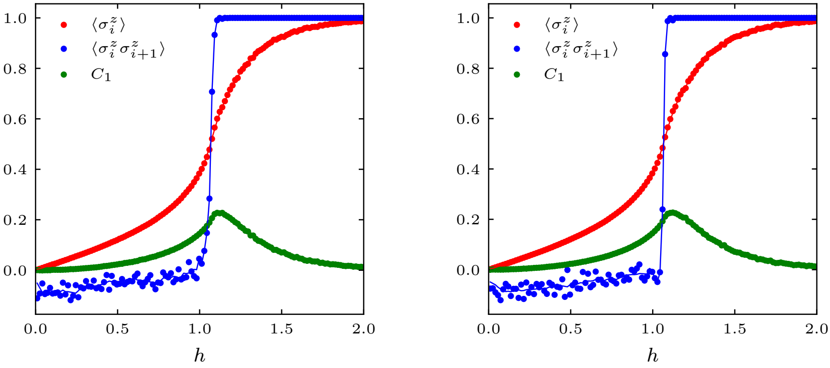
$$C_n \equiv \overline{\sigma_i^z \sigma_{i+n}^z} - \overline{\sigma_i^z}^2$$

Wrapping Probabilities



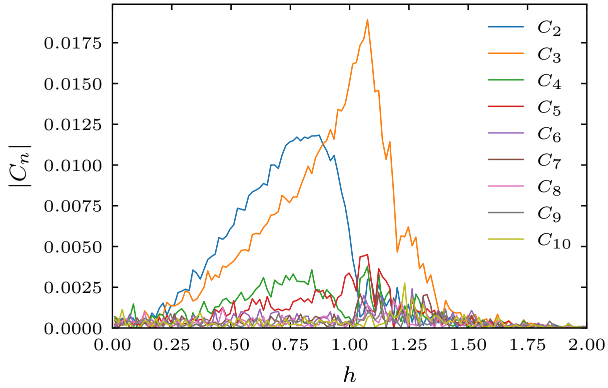
Wrapping probabilities $R_\uparrow$ and $R_\downarrow$ versus the transverse field h. (a) Spin-up. (b) Spin-down.

Correlation Functions



Expectation values of spin states $\overline{\sigma_i^z}$, product of NN spins $\overline{\sigma_i^z\sigma_{i+1}^z}$ and correlation function C_1 versus transverse field h. (a) $L = 64$. (b) $L = 128$.

Correlation Functions



Correlation functions $|C_n|$ versus the transverse field h.

Thank you for listening!