Matrix Product State Simulation of Quantum Phase Transitions

A Thesis

submitted to

Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
Master of Science in Quantum Technology

by

Devang Dimri

Indian Institute of Science Education and Research Pune

April, 2026

Supervisor: Mr. Lakshya Priyadarshi

Certificate

This is to certify that this dissertation entitled "Matrix Product State Simulation of Quantum Phase Transitions" submitted towards the partial fulfilment of the Master of Science in Quantum Technology represents work carried out by Devang Dimri from Indian Institute of Science Education and Research Pune under the supervision of Mr. Lakshya Priyadarshi (QpiAI India Pvt. Ltd.) during the academic year 2025–2026.

Mr. Lakshya Priyadarshi (Advisor)

Prof. Sreejith G. J. (TAC Member)

Declaration

I hereby declare that the matter embodied in the report entitled "Matrix Product State Simulation of Quantum Phase Transitions" are the results of the work carried out by me at the QpiAI India Pvt. Ltd. under the supervision of Mr. Lakshya Priyadarshi and the same has not been submitted elsewhere for any other degree.

Devang Dimri

Acknowledgements

I would like to convey my gratitude to my supervisor, Mr. Lakshya Priyadarshi, for his valuable constant support and insights in shaping the direction and execution of the project. His guidance has been instrumental in my journey.

I also extend my appreciation to Prof. Sreejith G.J., it was under him that I had my first project ever and greatly benefited from his guidance. His mentorship had a great impact on developing my interest in this field.

Deeply thankful to my friends and seniors for all the motivation, fun and beautiful experiences, and for just being who they are.

I am highly grateful to my parents, family members especially my sisters, Prerna, Pragya and Avani for their love, faith and encouragement through thick and thin and for shaping me into who I am today.

I would also like to thank Claude for the assistance in code debugging, the work is based out of my understanding and the tool helped in giving the final touches.

Abstract

The fundamental computational challenge in Quantum many body physics is about the exponential growth of the Hilbert space with the system size, therefore making it impossible to do exact simulation for more than a few tens of qubits. Matrix Product States (MPS) is a efficient way used to represent quantum states that obey the entanglement area law, using which one can accurately simulate one-dimensional quantum systems in polynomial computational cost.

In this work, we develop a GPU-based MPS simulator in PyTorch and use it to study phase transitions in the Transverse Field Ising Model (TFIM) and its non-integrable version, the Mixed-Field Ising Model (MFIM). We also implement imaginary time evolution using the Time-Evolving Block Decimation (TEBD) algorithm with a second-order Suzuki-Trotter decomposition and SVD-based bond dimension truncation.

We check the accuracy against exact diagonalisation for system sizes $N = 6$ to $N = 12$, giving errors of order 10^{-6} and 10^{-4} respectively at the critical point. The ground state phase diagram of the TFIM is described by computing six observables, energy density, longitudinal magnetisation $\langle Z \rangle$, transverse magnetisation $\langle X \rangle$, entanglement entropy, maximum bond dimension and a susceptibility proxy. The finite-size scaling of the half-chain entanglement entropy over system sizes $N = 6$ to $N = 20$ gives a central charge $c \approx 0.530$, which is nearly in accordance with the exact value $c = 0.5$ for the 2D Ising universality class. Moreover the power-law decay of ZZ spin correlations at criticality is also validated along with computing the entanglement spectrum and showing Rényi entropies for multiple indices.

Finally, we study the MFIM in the ordered phase (h/J = 0.5), where the longitudinal field forces the existing ferromagnetic order, causing the ground state to move toward a classical product state and thereby reducing entanglement entropy and bond dimension requirements as g increases. This demonstrates that the effect of non-integrability on entanglement is strongly phase-dependent, near the critical point, a longitudinal field would instead compete with quantum fluctuations and increase entanglement, a direction left for future work.

Contents

1 Introduction **1**

1.1 The Quantum Many-Body Problem 1
1.2 Tensor Networks and Matrix Product States 1
1.3 The Transverse Field Ising Model 2
1.4 Non-Integrable Extension: The Mixed-Field Ising Model 2
1.5 Objectives 3
1.6 Thesis Structure 3

2 Theoretical Background **4**

2.1 Quantum Entanglement and the Area Law 4
2.1.1 Entanglement Entropy 4
2.1.2 The Area Law 4
2.1.3 Logarithmic Violation at Criticality 5
2.1.4 Rényi Entropies 7
2.2 Matrix Product States 7
2.2.1 The Exponential Scaling and Tensor Decompositions 7
2.2.2 Construction via Sequential SVD 8
2.2.3 Definition and Tensor Convention 8
2.2.4 Schmidt Decomposition and SVD 9
2.2.5 Optimality of SVD Truncation 10
2.2.6 Canonical Forms 10
2.2.7 Matrix Product Operators 11
2.3 The TEBD Algorithm 13
2.3.1 Imaginary Time Evolution 13
2.3.2 Suzuki-Trotter Decomposition 13
2.3.3 Gate Application and Bond Truncation 14
2.4 The Transverse Field Ising Model 15
2.4.1 Phase Diagram 15
2.4.2 Exact Solution via Jordan-Wigner Transformation 16
2.5 Relation to DMRG 16

3 Methods **18**
3.1 Simulator Architecture . . . 18
3.1.1 Tensor Convention . . . 18
3.1.2 Initialisation . . . 18
3.1.3 Key Parameters . . . 19
3.2 Canonical Form Implementation . . . 19
3.2.1 Left Canonicalisation . . . 19
3.2.2 Right Canonicalisation . . . 20
3.2.3 Mixed-Canonical Form . . . 20
3.3 Gate Construction . . . 20
3.3.1 Transverse Field Gate . . . 20
3.3.2 ZZ Interaction Gate . . . 20
3.3.3 Longitudinal Field Gate (MFIM) . . . 21
3.4 Gate Application . . . 21
3.4.1 Single-Qubit Gates . . . 21
3.4.2 Two-Qubit Gates: Compressed and Deferred Variants . . . 21
3.4.3 Compression Sweep . . . 22
3.5 Expectation Value Computation . . . 22
3.5.1 Single-Site Operators . . . 22
3.5.2 Two-Site Expectation Values . . . 22
3.5.3 Correlation Functions . . . 22
3.5.4 Entanglement Entropy and Spectrum . . . 23
3.6 Energy Computation and Convergence . . . 23
3.7 The TEBD Step . . . 23
3.7.1 TFIM Step . . . 23
3.7.2 MFIM Step . . . 24
3.8 Numerical Stability . . . 24
3.9 Computational Complexity . . . 25
3.10 Data Collection Pipeline . . . 25

4 Results and Discussion **27**
4.1 Validation Against Exact Diagonalisation . . . 27
4.2 Phase Diagram of the TFIM . . . 28
4.2.1 Order Parameter and Magnetisation . . . 28
4.2.2 Entanglement Entropy and Bond Dimension . . . 28
4.2.3 Susceptibility Proxy . . . 29
4.3 Finite-Size Scaling and Central Charge . . . 30
4.4 Correlation Functions . . . 31
4.5 Entanglement Spectrum . . . 32

4.6 Extension to Non-Integrable Systems: The Mixed-Field Ising Model . 32

5 Conclusion and Future Work **35**

5.1 Summary of Achievements 35

5.2 Limitations 36

5.3 Future Directions 36

A Relevant Code **39**

A.1 TEBD Imaginary-Time Gates 39

A.2 SVD Truncation in Two-Site Gate Application 41

A.3 Connected Correlation Function via Transfer Matrix 42

A.4 Mixed-Field Ising Model (MFIM) Extension 43

A.5 Central Charge Fitting 44

List of Figures

2.1 Tensor network diagram of a Matrix Product State for N sites. Each circle represents a three-index tensor $A^{[k]}$ with a physical index (vertical leg, dimension $d = 2$) and two virtual bond indices (horizontal legs, dimension χ_k). 9

2.2 Diagrammatic representation of a single TEBD gate application on sites k and $k+1$. 14

4.1 MPS-TEBD vs exact diagonalisation at $h = J = 1$. *Left*: ground state energy E_0 vs system size N. *Right*: relative energy error on a logarithmic scale. Bond dimension $\chi_{\max} = 32$, Trotter step $\delta\tau = 0.005$. 28

4.2 TFIM phase diagram, $N = 20$, $J = 1$, $\chi_{\max} = 64$. Dashed red line: critical point $h/J = 1$. Panels show (top) energy density, $|\langle Z\rangle|$, $\langle X\rangle$; (bottom) entanglement entropy, max bond dimension, susceptibility proxy. 29

4.3 Finite-size scaling of the half-chain entanglement entropy at $h = J = 1$. *Left*: S vs $\ln N$ with least-squares fit to $S = \frac{c}{6}\ln N + \text{const}$, giving $c \approx 0.530$ ($R^2 = 0.960$). *Right*: fit residuals showing a systematic U-shaped pattern from subleading corrections. 30

4.4 ZZ connected correlations $C(0, j)$ at $h = J = 1$, $N = 20$. *Left*: decay with distance. *Right*: log-log plot with power-law fit $C \propto r^{-0.68}$ (exact CFT prediction for infinite PBC system: $r^{-0.25}$). 31

4.5 Entanglement properties at the centre bond, $N = 20$, $h = J = 1$. *Left*: Schmidt eigenvalues λ_i on a logarithmic scale. *Right*: Rényi entropies S_α for $\alpha \in \{0.5, 1.0, 2.0\}$. 32

List of Tables

4.1 Ground state energy comparison between MPS-TEBD and exact diagonalisation at the critical point $h = J = 1$, with bond dimension $\chi_{\max} = 32$ and Trotter step $\delta\tau = 0.005$. 27
4.2 Half-chain von Neumann entanglement entropy at the critical point $h = J = 1$ for system sizes $N = 6$ to $N = 20$. 30
4.3 Ground state properties of the MFIM with $J = 1$, $h = 0.5$, $N = 16$, $\chi_{\max} = 64$, as a function of longitudinal field g. At $g = 0$ the model is the integrable TFIM. 33

Chapter 1

Introduction

1.1 The Quantum Many-Body Problem

One of the major challenge in condensed matter physics and quantum information science is the simulation of quantum many-body systems. A system having N quantum spins each $\frac{1}{2}$ is described using a state vector in a Hilbert space of dimension 2^N. For $N = 50$ spins, this has more than 10^{15} complex amplitudes, which is much more than the memory of any classical computer. Storing such a state of a simple 50-qubit system would require petabytes of memory and its dynamics simulations would require large computational resources [1].

This scaling is not just because of any limitation, it is due to the quantum entanglement. When two subsystems are entangled, the state of the composite system cannot be written as a product of individual subsystem states, thus the information content of the joint state grows exponentially with the number of subsystems. Classical computers store information in bits rather than in qubits and have no natural way to represent this entanglement efficiently.

Despite this problem, many physically relevant states, like ground states of local Hamiltonians having an energy gap have much less entanglement than any arbitrary quantum state. For one-dimensional gapped systems, the entanglement entropy of any bipartition follows the *area law* and is bounded by a constant which is independent of system size [2]. This brings up the concept of tensor network methods, which exploit the limited entanglement structure of relevant states to give polynomial-scaling representations.

1.2 Tensor Networks and Matrix Product States

Tensor network methods provide an easy way for representing and manipulating quantum states whose entanglement is constrained. This works by decomposing the am-

plitudes of a quantum state into a network of tensors. Each tensor describes some local degrees of freedom, connected by indices that tell us about the entanglement information.

Matrix Product States (MPS) are the simplest class of tensor network ansatz for one-dimensional systems. Here we represents a quantum state as a product of matrices, one per lattice site, whose dimensions are bounded by a parameter χ called the bond dimension. χ controls both the accuracy of the representation and the computational cost. If we have larger χ we can capture more entanglement, which however comes at a correspondingly higher computational cost. For ground states satisfying the area law, a finite bond dimension is more than sufficient to represent the state upto a certain accuracy [3].

At quantum critical points, the area law is violated, the entanglement entropy grows and larger bond dimensions are needed.

1.3 The Transverse Field Ising Model

The Transverse Field Ising Model (TFIM) is one of the basic models of quantum phase transitions. The Hamiltonian is

$$\hat{H} = -J \sum_{i=1}^{N-1} Z_i Z_{i+1} - h \sum_{i=1}^{N} X_i, \tag{1.1}$$

where Z_i and X_i are Pauli operators on site i, J is the ferromagnetic coupling strength and h is the transverse field strength.

This model shows a quantum phase transition at $h/J = 1$. For $h/J < 1$ the system is in a ferromagnetic ordered phase with $\langle Z \rangle \neq 0$ as the order parameter. For $h/J > 1$ the system is in a disordered (paramagnetic) phase with $\langle Z \rangle = 0$. At the critical point $h/J = 1$ the system is described by conformal field theory with central charge $c = 1/2$, belonging to the 2D Ising universality class.

1.4 Non-Integrable Extension: The Mixed-Field Ising Model

TFIM is exactly solvable due to its $\mathbb{Z}_2$ symmetry and the existence of a Jordan-Wigner mapping. If we add a longitudinal field, that breaks the symmetry and integrability. The Mixed-Field Ising Model (MFIM) is given as

$$\hat{H}_{\text{MFIM}} = -J \sum_{i=1}^{N-1} Z_i Z_{i+1} - h \sum_{i=1}^{N} X_i - g \sum_{i=1}^{N} Z_i, \tag{1.2}$$

where g is the longitudinal field strength. For $g = 0$ this is just the TFIM. For $g \neq 0$ and $h \neq 0$, the model is non-integrable and must be treated numerically [4].

1.5 Objectives

The main objectives of the thesis are:

1. Implementing a GPU-based MPS simulator in PyTorch, without relying on any existing tensor network libraries such as iTensor or TeNPy.
2. Verifying the simulator rigorously against exact diagonalisation for small system sizes.
3. Characterising the quantum phase transition of the TFIM through multiple physical observables and extract the central charge via finite-size scaling.
4. Extending the simulator to the non-integrable MFIM and to show how non integrability shows up in different ground state behaviour compared to the integrable TFIM.

1.6 Thesis Structure

The thesis is organised as follows. Chapter 2 is about the theoretical background on area law, the MPS formalism and canonical forms, the TEBD algorithm, the physics of the TFIM, and the Calabrese-Cardy formula for entanglement entropy at criticality. Chapter 3 is about the implementation of the MPS simulator, including design choices, and numerical stability considerations. Chapter 4 presents all numerical results, benchmarking against exact diagonalisation, the TFIM phase diagram, finite-size scaling and central charge extraction, correlation functions, the entanglement spectrum, and the MFIM extension. Chapter 5 summarises the findings, discusses the limitations and future directions.

Chapter 2

Theoretical Background

2.1 Quantum Entanglement and the Area Law

2.1.1 Entanglement Entropy

If we have a quantum system described by a pure state $|\psi\rangle$ and it is bipartitioned into subsystems A and B then the reduced density matrix of subsystem A is obtained by tracing out B:

$$\rho_A = \mathrm{Tr}_B \, |\psi\rangle \langle\psi| \, . \tag{2.1}$$

The von Neumann entanglement entropy of this subsystem A is

$$S(A) = -\mathrm{Tr}(\rho_A \log \rho_A) = -\sum_i \lambda_i \log \lambda_i, \tag{2.2}$$

where $\{\lambda_i\}$ are the eigenvalues of ρ_A, also called as the Schmidt eigenvalues. The entropy $S(A)$ tells about the amount of quantum entanglement between A and B.

An arbitrary quantum state in a Hilbert space of dimension 2^N would have an entanglement entropy that scales as $S \sim N$ (as per volume law). The state itself requires 2^N complex amplitudes to specify, but most of these amplitudes participate non-trivially in the entanglement. In the case of local Hamiltonians, the physically relevant ground states have much less entanglement, and this structure is the only reason that makes efficient numerical simulation possible.

2.1.2 The Area Law

For one-dimensional systems with a gapped local Hamiltonian, the ground state entropy satisfy the area law, this was shown by Hastings [2]:

$$S(A) \leq \text{const} \tag{2.3}$$

independent of the subsystem size A. In one dimension, the area of a bipartition is simply a point or more precisely the boundary between A and B, hence the entropy is bounded by a constant independent of N.

Local interactions in a one-dimensional system generate entanglement only across the boundary between two subsystems A and B. Hence terms in the Hamiltonian that act completely within A or B have no effect on the entanglement between them, this can also be understood as since entanglement entropy does not vary under local unitaries acting within a single subsystem, only the interaction terms that cross over the bipartition generate entanglement between A and B.

For gapped systems, because of the presence of a finite energy gap $\Delta > 0$, there exists a finite correlation length, $\xi \sim 1/\Delta$, leading to exponentially decaying correlations $\langle O_i O_j \rangle - \langle O_i \rangle \langle O_j \rangle \sim e^{-|i-j|/\xi}$. Hence, only degrees of freedom within distance ξ of the boundary between A and B contribute primarily to the entanglement, and thus the entanglement entropy is bounded with system size,

$$S(A) = O(1). \tag{2.4}$$

Now if we consider the Schmidt decomposition across a bipartition:

$$|\psi\rangle = \sum_{i=1}^{\chi} s_i \; |i\rangle_A \otimes |i\rangle_B \,, \tag{2.5}$$

where χ is the Schmidt rank and $\{s_i\}$ are the Schmidt values. Then the entanglement entropy satisfies

$$S \leq \log \chi, \tag{2.6}$$

which gives the lower bound $\chi \geq e^S$. If $S = O(1)$, then $\chi = O(1)$, meaning the Schmidt rank does not grow with system size.

A general Matrix Product State with bond dimension χ requires $O(N\chi^2 d)$ parameters, where d is the local Hilbert space dimension. Since $\chi = O(1)$ for gapped one-dimensional systems, this scales as $O(N)$, which is polynomial in N, compared to the exponential $O(d^N)$ required for any arbitrary quantum state. This polynomial scaling makes MPS a computationally easy ansatz for gapped 1D systems, and the area law is therefore the main justification for the entire tensor network approach.

2.1.3 Logarithmic Violation at Criticality

At a quantum critical point, the energy gap decreases ($\Delta \to 0$), and the system becomes scale invariant. Now at this scale, the low-energy physics is governed by a one-dimensional conformal field theory (CFT).

For a subsystem of length l existing as a chain of total size N with periodic bound-

ary conditions, the entanglement entropy is given by the Calabrese–Cardy formula [5]:

$$S(l) = \frac{c}{3} \log\left(\frac{N}{\pi} \sin \frac{\pi l}{N}\right) + \text{const.} \tag{2.7}$$

This expression shows both the finite size of the system and its geometry. In the thermodynamic limit ($N \to \infty$), it reduces to:

$$S(l) \sim \frac{c}{3} \log l, \tag{2.8}$$

showing a logarithmic violation of the area law.

In case of open boundary conditions, the boundary effects reduce the coefficient by a factor of two. More precisely, for a half-chain bipartition ($l = N/2$), we get:

$$S(N/2) = \frac{c}{6} \log N + \text{const.} \tag{2.9}$$

The central charge c is a parameter of the CFT that tells about the universality class of the critical point and represents the number and nature of gapless degrees of freedom.

The logarithmic scaling of entanglement has some direct consequences in the tensor network representations. Using the bound $S \leq \log \chi$, the Schmidt rank must satisfy

$$\chi \gtrsim e^S. \tag{2.10}$$

Substituting the critical scaling $S \sim \frac{c}{6} \log N$ and simplifying,

$$\chi \gtrsim e^{\frac{c}{6} \log N} = N^{c/6}, \tag{2.11}$$

so the required bond dimension grows polynomially with system size. For the TFIM with $c = \frac{1}{2}$, this gives $\chi \sim N^{1/12}$, a slow yet non-trivial growth. Even though this is far more efficient than the $\chi \sim d^N$ required for a generic quantum state, it hints that MPS become progressively less efficient at criticality. In reality, this shows up as a need for increasing bond dimension to maintain accuracy as N grows, thus limiting the system sizes accessible at fixed computational resources. This is directly observed in the results of Chapter 4, where the relative energy error between MPS and exact diagonalisation grows with N at the critical point $h/J = 1$.

2.1.4 Rényi Entropies

The R'enyi entropy is a quantity for entanglement measure based on one parameter that generalises for different types of entropies:

$$S_\alpha = \frac{1}{1-\alpha} \log \sum_i \lambda_i^\alpha, \tag{2.12}$$

where λ_i are the eigenvalues of the reduced density matrix ρ_A . In the limit $\alpha \to 1$, we can get the von Neumann entropy:

$$S_1 = -\mathrm{Tr}(\rho_A \log \rho_A) \tag{2.13}$$

Different values of α signify the different regions of the entanglement spectrum. For $\alpha < 1$, the smaller eigenvalues are dominating, and for $\alpha > 1$, the larger eigenvalues are the ones dominating, so S_α shows the Schmidt coefficients accordingly.

This gives a way to pull out more detailed information about the entanglement structure than what is captured by the von Neumann entropy. At criticality , the R'enyi entropies follow a universal scaling. For a subsystem of length l in system of size N, the entropy is:

$$S_\alpha(l) = \frac{c}{6}\left(1 + \frac{1}{\alpha}\right) \log\left[\frac{N}{\pi} \sin\left(\frac{\pi l}{N}\right)\right] + \text{const}, \tag{2.14}$$

where c is the central charge of the CFT. For the half-chain ($l = N/2$), this changes to

$$S_\alpha(N/2) = \frac{c}{6}\left(1 + \frac{1}{\alpha}\right) \log N + \text{const}, \tag{2.15}$$

which gives the von Neumann result in the $\alpha \to 1$ limit. This scaling is important as it shows a method to calculate the central charge, by plotting numerical data for S_α as a function of system size and using multiple values of α as that improves the robustness.

2.2 Matrix Product States

2.2.1 The Exponential Scaling and Tensor Decompositions

The major obstacle in simulating quantum many-body systems on classical computers is about the exponential growth of the Hilbert space. The statevector for a system of N spin-$\frac{1}{2}$ particles is

$$|\psi\rangle = \sum_{\sigma_1,\ldots,\sigma_N \in \{0,1\}} \psi_{\sigma_1\sigma_2\cdots\sigma_N} |\sigma_1\sigma_2\cdots\sigma_N\rangle \tag{2.16}$$

where the coefficient tensor $\psi_{\sigma_1 \cdots \sigma_N}$ has 2^N complex entries. For $N = 50$, this tensor has $\sim 10^{15}$ entries, which exceeds the memory of any classical computer.

The main thing about tensor network methods is that the coefficient tensor of physically relevant states is not arbitrary, it has some hidden low-rank structure that allows it to be approximated by a network of much smaller tensors[6],[3]. One of such decompositions is the Matrix Product State, which forms the base of all the work done ahead.

2.2.2 Construction via Sequential SVD

For any given state we get the MPS representation arises by a sequence of singular value decompositions applied to the reshaped coefficient tensor. Starting from $\psi_{\sigma_1 \sigma_2 \cdots \sigma_N}$, the very first step in the construction is reshaping the coefficient tensor into a matrix $M^{(1)}$ by grouping the first index σ_1 on the left and all remaining indices $\sigma_2 \cdots \sigma_N$ on the right

$$M^{(1)}_{\sigma_1, (\sigma_2 \cdots \sigma_N)} \quad \text{of size } 2 \times 2^{N-1}. \tag{2.17}$$

we then perform SVD to get $M^{(1)} = U^{(1)} \Sigma^{(1)} V^{(1)\dagger}$, where $U^{(1)}$ has atmost $\chi_1 = \min(2, 2^{N-1})$ columns. We then define $A^{\sigma_1}[1] \equiv U^{(1)}_{\sigma_1, :}$ which is a row vector of dimension χ_1, and pass $\Sigma^{(1)} V^{(1)\dagger}$ further. Reshape $\Sigma^{(1)} V^{(1)\dagger}$ into a matrix of size $(\chi_1 \cdot 2) \times 2^{N-2}$, perform SVD, and extract $A^{\sigma_2}[2]$. This whole process is repeated until all N required tensors have been formed.

The resulting MPS form is exact when no truncation is applied, since the bond dimensions grow as $\chi_k = \min(2^k, 2^{N-k})$. We can also approximate the MPS form, approximation is done by truncating the singular values at each step, keeping only the $\chi_{\max}$ largest. From this sequential-SVD derivation we can see that an MPS is basically a systematic low-rank decomposition of the coefficient tensor. It also shows that the MPS representation is exact for any state, the question is only whether the required bond dimension χ grows polynomially or exponentially with N.

2.2.3 Definition and Tensor Convention

A Matrix Product State for N qubits (spin-$\frac{1}{2}$ sites) is written as [6]

$$|\psi\rangle = \sum_{\sigma_1, \ldots, \sigma_N} A^{\sigma_1}[1] A^{\sigma_2}[2] \cdots A^{\sigma_N}[N] \, |\sigma_1 \sigma_2 \cdots \sigma_N\rangle \,, \tag{2.18}$$

where each $A^{\sigma_k}[k]$ is a $\chi_{k-1} \times \chi_k$ matrix, $\sigma_k \in \{0, 1\}$ is the physical index (spin state), and χ_k is the bond dimension at bond k. The boundary conditions require $\chi_0 = \chi_N = 1$ so that the matrix product yields a scalar.

Figure 2.1 shows the tensor network diagram of this representation, where each

site tensor carries one physical leg and two virtual bond legs.

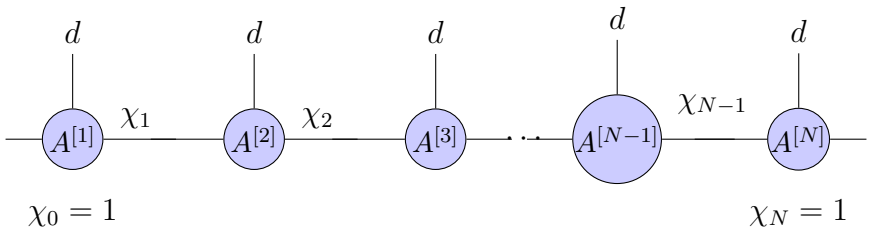

Figure 2.1: Tensor network diagram of a Matrix Product State for N sites. Each circle represents a three-index tensor $A^{[k]}$ with a physical index (vertical leg, dimension $d = 2$) and two virtual bond indices (horizontal legs, dimension χ_k).

In this work, we use three-index tensors $A[k]$ with shape (χ_L, d, χ_R), where $d = 2$ is the physical dimension and χ_L, χ_R are the left and right bond dimensions. Explicitly, $A^{\sigma_k}[k] = A[k][:, \sigma_k, :]$.

A product state, zero entanglement state, has all bond dimensions equal to 1, so each tensor is basically just a vector with two components. Entanglement is captured by increasing the bond dimension, a bond dimension of χ can represent states with up to $\log_2 \chi$ bits of entanglement across that bond. The computational cost of all MPS operations (gate application, expectation values, canonical form sweeps) scales as $O(N\chi^3)$ or $O(N\chi^2 d)$ depending on the operation, replacing the exponential cost of working with the full state vector.

2.2.4 Schmidt Decomposition and SVD

The relation between MPS and entanglement is made more precise by the Schmidt decomposition. For any bipartition of $|\psi\rangle$ into sites $1, \ldots, k$ (left) and $k+1, \ldots, N$ (right), the state can be written as

$$|\psi\rangle = \sum_{\alpha=1}^{\chi_k} s_\alpha \, |\phi_\alpha^L\rangle \, |\phi_\alpha^R\rangle \,, \tag{2.19}$$

where $\{s_\alpha\}$ are the Schmidt values (singular values), $|\phi_\alpha^L\rangle$ and $|\phi_\alpha^R\rangle$ are orthonormal basis for the left and right subsystems, and χ_k is the Schmidt rank. The Schmidt eigenvalues are $\lambda_\alpha = s_\alpha^2$, and have normalisation such that $\sum_\alpha \lambda_\alpha = 1$. The entanglement entropy then is, $S = -\sum_\alpha \lambda_\alpha \log \lambda_\alpha$. The Schmidt decomposition is obtained numerically via SVD by reshaping the coefficient tensor as a matrix of dimensions $(2^k) \times (2^{N-k})$ and doing SVD to get singular values and the Schmidt states directly. In the MPS representation, the bond dimension χ_k at bond k equals the Schmidt rank of the bipartition at that bond. If the true Schmidt rank exceeds $\chi_{\max}$, the smallest singular values are discarded and the MPS approximates the true state, with truncation error as given by the discarded weight.

2.2.5 Optimality of SVD Truncation

The only question in the tensor network computation is that given a bond dimension budget of $\chi_{\max}$, which singular values should we keep so that we minimise the approximation error. The answer is given by the Eckart-Young theorem [7]:

Theorem 2.1 (Eckart–Young, 1936)**.** *Let M be a matrix having singular value decomposition $M = U\Sigma V^\dagger$, where the singular values $s_1 \geq s_2 \geq \cdots \geq s_r > 0$ are ordered by decreasing magnitude. The best rank-χ approximation to M in the Frobenius norm is obtained by keeping only the χ largest singular values:*

$$\|M - M_\chi\|_F = \sqrt{\sum_{i>\chi} s_i^2} \equiv \sqrt{\varepsilon_{\text{disc}}}, \tag{2.20}$$

where $M_\chi = U_\chi \Sigma_\chi V_\chi^\dagger$ retains only the top χ singular triplets.

When used for the two-site tensor $\tilde{\Theta}$ during a TEBD gate application, this theorem guarantees that keeping the $\chi_{\max}$ largest singular values is the *optimal* rank-$\chi_{\max}$ approximation. The discarded weight keeps all singular values whose squared contribution to the total norm exceeds a threshold $\varepsilon_{\text{trunc}}$:

$$\text{keep index } i \iff s_i^2 > \varepsilon_{\text{trunc}} \sum_j s_j^2, \tag{2.21}$$

subject to the hard cap $i \leq \chi_{\max}$. This is the truncation limit implemented in the simulator as well. The Eckart–Young theorem guarantees that this strategy is globally optimal and no other choice of χ singular values would have a smaller Frobenius-norm error for the same bond dimension budget.

2.2.6 Canonical Forms

One important thing about the MPS representation is that it is not unique, we can insert a pair of inverse matrices X and X^{-1} at any bond without changing the physical state, $A[k] \to A[k]X$ and $A[k+1] \to X^{-1}A[k+1]$. This freedom has $O(\chi^2)$ continuous parameters per bond, and fixing it to a particular canonical form is essential for numerical stability and efficient computation [6].

Left-canonical form : A tensor $A[k]$ is left-orthonormal if

$$\sum_\sigma A^{\sigma\dagger}[k]A^\sigma[k] = \mathbb{I}, \quad \text{i.e.,} \quad \sum_{\sigma,\alpha_L} A^*_{\alpha_L,\sigma,\alpha_R}[k]\, A_{\alpha_L,\sigma,\alpha'_R}[k] = \delta_{\alpha_R\alpha'_R}. \tag{2.22}$$

An MPS in left-canonical form has all tensors left-orthonormal. It is computed by sweeping left to right, performing QR decompositions and absorbing the upper tri-

angular R factor into the next tensor. After left canonicalisation, the norm satisfies $\langle\psi|\psi\rangle = \mathrm{Tr}(A^\dagger[N]A[N])$, which is a purely local computation.

Right-canonical form : Similar to left canonical form we can have a right canonical form as well which is Computed by sweeping right to left with QR decompositions. Any tensor $B[k]$ is right-orthonormal if $\sum_\sigma B^\sigma[k]B^{\sigma\dagger}[k] = \mathbb{I}$.

Mixed-canonical form : In case of a fixed orthogonality centre c, all tensors to the left of c are left-orthonormal and all tensors to the right are right-orthonormal. We get this by performing a left sweep from site 0 to $c-1$ and a right sweep from site $N-1$ to $c+1$. The non-unitary factors pille up at site c, so

$$\langle\psi|\psi\rangle = \mathrm{Tr}(A^\dagger[c]A[c]), \tag{2.23}$$

which again becomes a purely local computation. More importantly, when the orthogonality centre is at site c, the left and right environments ,all tensors except at c contract to identity matrices due to the orthonormality conditions (2.22). Hence expectation value of a local operator O_c at site c therefore reduces to

$$\langle O_c\rangle = \sum_{\alpha_L,\sigma,\sigma',\alpha_R} A^*_{\alpha_L,\sigma,\alpha_R}[c]\, O_{\sigma,\sigma'}\, A_{\alpha_L,\sigma',\alpha_R}[c], \tag{2.24}$$

which is a contraction of cost $O(\chi^2 d^2)$ rather than the $O(N\chi^3)$ required without canonical form. This reduction from a global to a local computation is the main practical advantage of canonical forms and is used throughout in the implementation as well.

2.2.7 Matrix Product Operators

A Matrix Product Operator (MPO) is the operator version of an MPS. It represents a linear operator $\hat{O}$ acting on the full Hilbert space as a product of local tensors [6]:

$$\hat{O} = \sum_{\{\sigma_k\},\{\sigma'_k\}} W^{\sigma_1\sigma'_1}[1]\, W^{\sigma_2\sigma'_2}[2]\cdots W^{\sigma_N\sigma'_N}[N]\, |\sigma_1\cdots\sigma_N\rangle\, \langle\sigma'_1\cdots\sigma'_N|\,, \tag{2.25}$$

where each local tensor $W^{\sigma_k\sigma'_k}[k]$ carries two physical indices $\sigma_k, \sigma'_k \in \{1,\ldots,d\}$ and two virtual bond indices, making it a rank-4 tensor of dimension $D_{k-1}\times D_k \times d \times d$. The boundary conditions $D_0 = D_N = 1$ ensure $W[1]$ is a row vector and $W[N]$ a column vector of local operators, so contracting all bond indices yields a scalar in the virtual space the full operator.

In the same way that the MPS bond dimension χ controls how expressive the state is, the MPO bond dimension D also controls how complex the operator is. For

a k-local Hamiltonian, the bond dimension D is bounded by the number of different operator strings that can be there across any bond those beginning left of the cut and terminating to the right. Since this number depends only on the interaction range k and local dimension d and not on system size N, the local Hamiltonians admit exact MPO representations with D growing at most polynomially in k. Applying an MPO $\hat{O}$ to an MPS $|\psi\rangle$ produces a new MPS,

$$|\phi\rangle = \hat{O}\,|\psi\rangle\,, \quad B^{\sigma_k}[k]_{(\alpha_L,\beta_L),(\alpha_R,\beta_R)} = \sum_{\sigma'_k} W^{\sigma_k\sigma'_k}[k]_{\beta_L,\beta_R}\, A^{\sigma'_k}[k]_{\alpha_L,\alpha_R}, \tag{2.26}$$

and the bond dimension of $|\phi\rangle$ is at most χD. When applying an MPO to an MPS or composing two MPOs (e.g., computing $\hat{H}^2$ as an MPO), the bond dimension grows multiplicatively hence we can again compress the resulting MPO or MPS back to a manageable bond dimension using singular value decomposition (SVD)-based truncation, discarding singular values below a threshold ϵ. For states and operators with limited entanglement, the singular values at each bond decay rapidly, and an accurate approximation is maintained at low bond dimension D. The TFIM Hamiltonian $\hat{H} = -J\sum_i Z_i Z_{i+1} - h\sum_i X_i$ can be written exactly as an MPO with bond dimension $D = 3$. The bulk tensor is given as,

$$W[k] = \begin{pmatrix} \mathbb{I} & 0 & 0 \\ -JZ & 0 & 0 \\ -hX & Z & \mathbb{I} \end{pmatrix}, \tag{2.27}$$

where each entry is a 2×2 operator in the physical indices (σ_k, σ'_k), and the 3×3 block structure acts on the MPO virtual indices. The boundary tensors fix the edges, $W[1] = (-hX,\ Z,\ \mathbb{I})$ selects the last row, and $W[N] = (\mathbb{I},\ -JZ,\ -hX)^\top$ selects the first column.

The MPO representation makes it crystal clear why the Trotter decomposition splits $\hat{H}$ into commuting layers, the $D = 3$ bond structure encodes exactly three sublayers (H_{even}, H_{odd}, and H_X), and within each layer the gates act on non-overlapping bonds and hence commute. When applying an MPO to an MPS with bond dimension χ, the result is a new MPS having bond dimension χD, at a cost of $\mathcal{O}(N\chi^2 D^2 d^2)$. This connection between the MPO structure and the Trotter layers further supports the efficiency of TEBD.

2.3 The TEBD Algorithm

2.3.1 Imaginary Time Evolution

The ground state of any Hamiltonian $\hat{H}$ can be found by imaginary time evolution.If we substitute $t \to -i\tau$ in the Schrödinger equation, it gives the imaginary time evolution operator $e^{-\hat{H}\tau}$. Applying this to any initial state $|\psi_0\rangle$ with non-zero overlap with the ground state,

$$e^{-\hat{H}\tau} |\psi_0\rangle = \sum_n c_n e^{-E_n\tau} |n\rangle , \tag{2.28}$$

where $\{E_n, |n\rangle\}$ are the eigenvalues and eigenstates. As $\tau \to \infty$, all excited state components are suppressed compared to the ground state component by factors $e^{-(E_n-E_0)\tau}$, leaving only $|E_0\rangle$ significant after renormalisation.

The rate of convergence is set by the spectral gap $\Delta = E_1 - E_0$: after imaginary time $\tau \sim 1/\Delta$, the first excited state contribution is suppressed by e^{-1}. At a quantum critical point the gap closes as $\Delta \sim N^{-1}$ (for a finite system), so imaginary time convergence slows as the system size grows. This is why we use convergence tolerance 10^{-8} on the energy change per TEBD step rather than a fixed imaginary time.

2.3.2 Suzuki-Trotter Decomposition

The exact imaginary time evolution operator $e^{-\hat{H}\delta\tau}$ cannot be applied directly to an MPS because $\hat{H}$ couples multiple sites simultaneously. Instead, we use the Suzuki-Trotter decomposition to approximate $e^{-\hat{H}\delta\tau}$ as a product of two-site gates, each of which can be applied locally [8, 9].

For the TFIM, the Hamiltonian splits into three commuting-within-layer terms:

$$H_{\text{even}} = -J \sum_{k \text{ even}} Z_k Z_{k+1}, \tag{2.29}$$

$$H_{\text{odd}} = -J \sum_{k \text{ odd}} Z_k Z_{k+1}, \tag{2.30}$$

$$H_X = -h \sum_k X_k. \tag{2.31}$$

The second-order (symmetric) Trotter decomposition is

$$e^{-(H_X+H_{\text{even}}+H_{\text{odd}})\delta\tau} \approx e^{-H_X\delta\tau/2}\, e^{-H_{\text{even}}\delta\tau}\, e^{-H_{\text{odd}}\delta\tau}\, e^{-H_X\delta\tau/2} + O(\delta\tau^3). \tag{2.32}$$

Since $[H_{\text{even}}, H_{\text{odd}}] = 0$ exactly on an open chain, the ZZ terms on non-overlapping bonds commute because $[Z_iZ_{i+1},\, Z_{i+1}Z_{i+2}] = 0$, the two middle factors combine exactly into $e^{-H_{ZZ}\delta\tau}$ where $H_{ZZ} = H_{\text{even}} + H_{\text{odd}}$. The decomposition is therefore a standard second-order symmetric splitting of the two genuinely non-commuting pieces

H_X and H_{ZZ}, with no additional approximation introduced by treating the even and odd bond layers separately. The error per step is $O(\delta\tau^3)$, arising from commutators between the layers, the total error over T steps (total imaginary time $\tau = T\delta\tau$) is $O(\delta\tau^2\tau)$. This quadratic convergence in $\delta\tau$ allows the Trotter step to be taken small enough that truncation error, and not the Trotter error, is the dominant source of inaccuracy. Within each layer even bonds, odd bonds, or single-site X terms, the gates act on non-overlapping bonds and therefore commute with each other within the layer hence they can be applied simultaneously and exactly, without any further approximation beyond the Trotter splitting itself.

2.3.3 Gate Application and Bond Truncation

Applying a two-site gate U to sites k and $k+1$ proceeds as follows, and is shown diagrammatically in Figure 2.2.

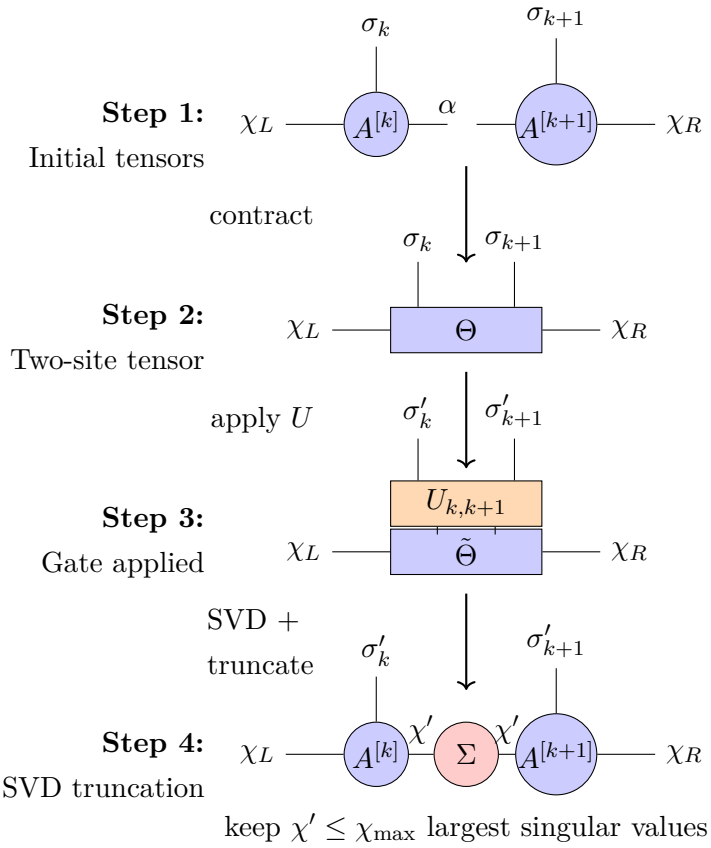

Figure 2.2: Diagrammatic representation of a single TEBD gate application on sites k and $k+1$.

The discarded weight $\varepsilon_{\text{disc}} = \sum_{i>\chi} s_i^2 / \sum_i s_i^2$ quantifies the truncation error at this bond and step. Over many TEBD steps, these errors accumulate, so keeping $\varepsilon_{\text{disc}}$ small is important for long imaginary-time evolutions. For real-time evolution, $U = e^{-iH\delta t}$ is unitary, so the gate application preserves the norm and the singular values before

and after the gate are of the same order. For imaginary time evolution, $U = e^{-H\delta\tau}$ is symmetric positive definite (not unitary), and the state norm changes at each step. Hence the renormalisation step in which we divide the centre tensor by its norm after each full Trotter step, becomes essential as well.

2.4 The Transverse Field Ising Model

The Transverse Field Ising Model (TFIM) on an open chain of N sites is defined by the Hamiltonian

$$\hat{H}_{\text{TFIM}} = -J \sum_{i=1}^{N-1} Z_i Z_{i+1} - h \sum_{i=1}^{N} X_i, \tag{2.33}$$

where Z_i and X_i are Pauli operators on site i, $J > 0$ is the ferromagnetic coupling strength, and $h > 0$ is the transverse field strength. We set $J = 1$ throughout and vary h.

The Hamiltonian commutes with the global spin-flip operator $\hat{Q} = \prod_i X_i$, which flips all spins simultaneously: $[\hat{H}, \hat{Q}] = 0$. This $\mathbb{Z}_2$ symmetry partitions the Hilbert space into even ($\hat{Q} = +1$) and odd ($\hat{Q} = -1$) sectors.

2.4.1 Phase Diagram

The ground state phase diagram contains two phases separated by a continuous (second-order) quantum phase transition at $h/J = 1$:

- **Ordered (ferromagnetic) phase**, $h/J < 1$: the $\mathbb{Z}_2$ symmetry is spontaneously broken in the thermodynamic limit, and the longitudinal magnetisation $\langle Z \rangle \neq 0$ serves as the order parameter. In the extreme limit $h = 0$, the two degenerate ground states are the fully polarised states $|\uparrow\uparrow\cdots\uparrow\rangle$ and $|\downarrow\downarrow\cdots\downarrow\rangle$.
- **Disordered (paramagnetic) phase**, $h/J > 1$: the $\mathbb{Z}_2$ symmetry is preserved, $\langle Z \rangle = 0$, and the transverse magnetisation $\langle X \rangle \to 1$ as $h/J \to \infty$. The ground state approaches the product state $|+\rangle^{\otimes N}$ where $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$.

At the critical point $h/J = 1$, the correlation length diverges as $\xi \sim |h - J|^{-\nu}$ with correlation length exponent $\nu = 1$, and the energy gap closes as $\Delta \sim N^{-1}$ for a finite system with open boundary conditions. The system is described by a conformal field theory with central charge $c = 1/2$, placing it in the universality class of the two-dimensional classical Ising model.

2.4.2 Exact Solution via Jordan-Wigner Transformation

The TFIM is exactly solvable via the Jordan-Wigner (JW) transformation, which maps the spin-$\frac{1}{2}$ operators to spinless fermion operators [10]:

$$c_i = \left(\prod_{j<i} X_j\right) \frac{Z_i + iY_i}{2}, \quad c_i^\dagger = \left(\prod_{j<i} X_j\right) \frac{Z_i - iY_i}{2}, \tag{2.34}$$

where c_i, $c_i^\dagger$ satisfy canonical anticommutation relations $\{c_i, c_j^\dagger\} = \delta_{ij}$. In terms of these fermionic operators, the TFIM Hamiltonian becomes a free-fermion model:

$$\hat{H} = -J\sum_i (c_i^\dagger c_{i+1} + c_{i+1}^\dagger c_i + c_i^\dagger c_{i+1}^\dagger + c_{i+1} c_i) - h\sum_i (2c_i^\dagger c_i - 1), \tag{2.35}$$

which can be diagonalised exactly by a Bogoliubov transformation. This maps the problem to a set of independent fermionic modes with dispersion relation $\varepsilon_k = 2\sqrt{J^2 + h^2 - 2Jh\cos k}$, and the ground state is the vacuum of the Bogoliubov quasiparticles.

Key exact results used to benchmark the simulator are:

- **Ground state energy density** in the thermodynamic limit at the critical point: $E_0/N \to -4J/\pi \approx -1.2732\, J$.
- **Correlation length exponent**: $\nu = 1$.
- **Central charge**: $c = 1/2$ (2D Ising universality class).
- **ZZ spin-spin correlations** at criticality in an infinite periodic system: $\langle Z_i Z_j \rangle \sim |i-j|^{-1/4}$.
- **Transverse magnetisation** at criticality: $\langle X \rangle = (2/\pi)E(k^2)$ where E is the complete elliptic integral of the second kind evaluated at $k = 1$, giving $\langle X \rangle = 2/\pi$.

These exact results provide a comprehensive benchmark suite against which the MPS simulator is validated in Chapter 4.

2.5 Relation to DMRG

The Density Matrix Renormalisation Group (DMRG) [11] is by far the most powerful numerical method for solving one-dimensional quantum systems. Originally formulated by White in 1992 as an iterative procedure for truncating the Hilbert space of an expanding block, it was reformulated in the early 2000s as a variational optimisation

directly in the space of MPS [6]:

$$E_0(\chi) = \min_{|\psi\rangle:\chi_k \leq \chi} \frac{\langle\psi|\hat{H}|\psi\rangle}{\langle\psi|\psi\rangle}, \tag{2.36}$$

where the minimisation is over all MPS with bond dimension at most χ. In the standard two-site DMRG, the energy is minimised one bond at a time. At each step, the orthogonality centre is placed at site c, and all other tensors are fixed. The energy then becomes a quadratic form in $A[c]$, which is then minimised by solving the effective eigenvalue problem

$$H_{\text{eff}}\, A[c] = E\, A[c], \tag{2.37}$$

where H_{eff} is the Hamiltonian projected onto the mixed-canonical environment. After the update, SVD is used to truncate the bond dimension and move the orthogonality centre to the next site, repeating the sweep until convergence. The cost per sweep is $O(N\chi^3 D)$, where D is the MPO bond dimension. The key structural difference between TEBD and DMRG is that the latter directly minimises the energy within the MPS manifold, while TEBD drives the state toward the ground state via imaginary time evolution. Hence as a result, DMRG typically achieves higher accuracy at equivalent bond dimension for ground state calculations, because each sweep directly optimises every tensor for the current MPS geometry. TEBD is simpler to implement and extends directly to real-time dynamics by replacing $\delta\tau$ with $i\delta t$. It also avoids the need to store the full MPO and build effective Hamiltonian environments. TEBD requires a Trotter decomposition and thus introduces a systematic Trotter error $O(\delta\tau^2)$ per unit imaginary time, which DMRG avoids entirely. Both the methods represent the ground state as an MPS, once the MPS is obtained, all post-processing like expectation values, correlation functions, entanglement spectrum is identical in both.

TEBD was chosen for the work here because its structure, imaginary time evolution via local gates directly derived from the Hamiltonian, is transparent and clear. The MPO and canonical form infrastructure built here would support a DMRG implementation with the addition of the effective Hamiltonian construction and the local eigenvalue solver.

Chapter 3

Methods

3.1 Simulator Architecture

The MPS simulator is implemented in Python using PyTorch, which provides automatic GPU acceleration through CUDA. The core class encapsulates the full MPS state and all operations on it, including canonical form sweeps, gate application, SVD truncation, and physical observables. Each method has a well-defined input/output contract, which made systematic validation against exact diagonalisation straightforward: any discrepancy between the MPS result and the ED reference could be isolated to a single method.

3.1.1 Tensor Convention

Each site tensor [k] has shape (χ_L, d, χ_R), where $d = 2$ is the physical dimension and χ_L, χ_R are themleft and right bond dimensions, equal to 1 at the respective chain boundaries. Placing the physical index in the middle position ensures that the reshapes required during gate application and SVD truncation act on continuous memory blocks, avoiding expensive tensor transpositions on GPU.

All tensors are stored in complex128, double-precision complex, providing approximately 15 decimal digits of precision. This is essential because energy differences near criticality are of order 10^{-6}, which would be difficult to differentiate from single-precision floating-point noise. `complex64` was tested and found to introduce errors at the 10^{-7} level, sufficient to corrupt the physics signals being measured.

3.1.2 Initialisation

The simulator is initialised in the all-zeros product state $|00\cdots0\rangle$ with bond dimension 1 everywhere. This state has non-zero overlap with the ground state in both phases of the TFIM: it is an equal superposition of the two $\mathbb{Z}_2$-degenerate ground states at $h = 0$,

and has overlap $2^{-N/2}$ with $|+\rangle^{\otimes N}$ in the paramagnetic phase. Since imaginary time evolution projects onto the ground state for any initial state with non-zero overlap, this choice does not bias the result toward either phase.

3.1.3 Key Parameters

The simulator exposes four parameters that control the accuracy–cost trade-off:

- $\chi_{\max}$ (`max_bond_dim`): the maximum bond dimension retained after SVD truncation. All results in the work ahead use $\chi_{\max} \in \{32, 64\}$ depending on the computation.
- $\varepsilon_{\text{trunc}}$ (`trunc_eps`): the relative discarded-weight threshold. Singular values whose squared weight falls below $\varepsilon_{\text{trunc}}$ of the total are discarded, even if fewer than $\chi_{\max}$ values have been retained.
- `hard_bond_dim` $= 2\chi_{\max}$: a hard cap applied before compression sweeps, preventing memory overflow when gates are applied without immediate SVD truncation.
- `needs_compression`: a per-bond boolean flag set by `apply_2q_gate_nocompress`. The `compress_sweep` method skips bonds where this flag is false, avoiding redundant SVDs on bonds whose dimension has not changed.

3.2 Canonical Form Implementation

The theoretical definitions and orthonormality conditions for left-, right-, and mixed-canonical forms are given in Section 2.2.6. Here are the concrete reshape dimensions and loop structures used in the three corresponding methods.

3.2.1 Left Canonicalisation

Defined a function `left_canonicalize` that sweeps from site 0 to $N-2$. At each site k, the tensor $A[k]$ is reshaped to a $(\chi_L d) \times \chi_R$ matrix and QR-decomposed. The unitary factor Q, reshaped back to $(\chi_L, d, \chi_{\text{new}})$, is stored as the new left-orthonormal $A[k]$, and the upper-triangular R is absorbed into $A[k+1]$ via `torch.tensordot`, propagating the non-unitary weight rightward. After the full sweep, the state norm is encoded entirely in the rightmost tensor: $\langle\psi|\psi\rangle = \text{Tr}(A^\dagger[N-1]\, A[N-1])$.

3.2.2 Right Canonicalisation

Defined a function `right_canonicalize` that mirrors the left sweep, running over `reversed(range(1, N))` and propagating R factors leftward. The loop deliberately excludes site 0 from the range start: site 0 receives the R factor from site 1 via the normal absorption step and requires no special handling.

3.2.3 Mixed-Canonical Form

Defined a function `set_mixed_canonical(center)` that performs a left sweep over sites $0, \ldots, c-1$ and a right sweep over sites $N-1, \ldots, c+1$, accumulating all non-unitary factors at the orthogonality centre c. As derived in Section 2.2.6, this reduces local operator expectation values to $O(\chi^2 d^2)$ contractions and the state norm to the purely local computation $\langle\psi|\psi\rangle = \mathrm{Tr}(A^\dagger[c]\, A[c])$. This method is called before every expectation value measurement and after every complete TEBD step for renormalisation.

3.3 Gate Construction

The imaginary time gates are constructed analytically from the Hamiltonian terms rather than by numerical matrix exponentiation, which is both faster and exact to machine precision.

3.3.1 Transverse Field Gate

The single-site gate for the transverse field term $-hX_k$ over a half step $\delta\tau/2$ is

$$G_X = e^{+hX_k\,\delta\tau/2} = \cosh\Big(\tfrac{h\,\delta\tau}{2}\Big)\,\mathbb{I} + \sinh\Big(\tfrac{h\,\delta\tau}{2}\Big)\,X, \tag{3.1}$$

where the positive exponent follows from the negative sign in the Hamiltonian: $e^{-(-hX_k)\delta\tau/2} = e^{+hX_k\delta\tau/2}$. The identity $e^{\lambda X} = \cosh\lambda\,\mathbb{I} + \sinh\lambda\,X$ holds because $X^2 = \mathbb{I}$.

3.3.2 ZZ Interaction Gate

The two-site gate for the interaction term $-JZ_kZ_{k+1}$ over a full step $\delta\tau$ is diagonal in the computational basis:

$$G_{ZZ} = e^{+JZ_kZ_{k+1}\,\delta\tau} = \mathrm{diag}\Big(e^{\lambda},\, e^{-\lambda},\, e^{-\lambda},\, e^{\lambda}\Big), \quad \lambda = J\,\delta\tau, \tag{3.2}$$

in the basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. The diagonal structure means this gate can be applied at cost $O(\chi^2 d)$ rather than the full $O(\chi^2 d^2)$ of a dense two-site gate, though

the current implementation uses the general `apply_2q_gate` path for uniformity.

3.3.3 Longitudinal Field Gate (MFIM)

For the Mixed-Field Ising Model, the additional longitudinal field term $-gZ_k$ contributes a single-site gate

$$G_Z = e^{+gZ_k\,\delta\tau/2} = \cosh\Big(\tfrac{g\,\delta\tau}{2}\Big)\,\mathbb{I} + \sinh\Big(\tfrac{g\,\delta\tau}{2}\Big)\,Z, \tag{3.3}$$

derived analogously to G_X using $Z^2 = \mathbb{I}$. At $g = 0$ this reduces to the identity, so the MFIM step reduces exactly to the TFIM step when the longitudinal field is absent.

3.4 Gate Application

3.4.1 Single-Qubit Gates

A single-qubit gate U is applied to site q by contracting along the physical index via `torch.tensordot` and restoring the (χ_L, d, χ_R) index order via `permute(1,0,2)`. Bond dimensions are unchanged and no truncation is needed.

3.4.2 Two-Qubit Gates: Compressed and Deferred Variants

Two distinct gate application methods are used, selected based on whether immediate SVD truncation is required.

We define a `apply_2q_gate` which applies the gate and immediately performs SVD truncation. It contracts $A[k]$ and $A[k+1]$ into the two-site tensor Θ, applies U by reshaping and matrix multiplication, then performs SVD on the $(\chi_L d)\times(d\chi_R)$ reshape of $\tilde{\Theta}$. The left singular vectors are stored as the new $A[k]$ and $\Sigma V^\dagger$ is stored as $A[k{+}1]$. The Eckart–Young truncation criterion (Section 2.2.6) is applied immediately, keeping singular values until the discarded weight $\varepsilon_{\text{disc}} = \sum_{i>k} s_i^2 / \sum_i s_i^2$ falls below $\varepsilon_{\text{trunc}}$, subject to the hard cap $k \le \chi_{\max}$.

Another fuction defined is, `apply_2q_gate_nocompress` which applies the gate using QR rather than SVD, without any truncation. This is used inside the compression sweep when a fast, approximate bond split is needed without the cost of a full SVD. The bond dimension is capped at `hard_bond_dim` $= 2\chi_{\max}$ to prevent unbounded growth between compression sweeps, and the `needs_compression` flag is set on the affected bond.

3.4.3 Compression Sweep

We define a function `compress_sweep` that iterates over all bonds flagged by `needs_compression` and performs a full SVD truncation, reducing the bond dimension to $\chi_{\max}$. Bonds whose dimension is already within budget are skipped. If SVD fails which can occur for nearly-rank-deficient tensors at large $\delta\tau$, the method falls back to QR with manual truncation. After truncation the flag is cleared.

3.5 Expectation Value Computation

3.5.1 Single-Site Operators

For a Pauli operator O at site q, `set_mixed_canonical(q)` is called before measurement. As established in Section 2.2.6, all environment tensors then contract to identity and the expectation value reduces to a local `torch.einsum` over $A[q]$ alone at cost $O(\chi^2 d^2)$. Both `expectation_pauli` (for X, Y, Z, I) and `expectation_general_1site` (for arbitrary 2×2 operators) follow this pattern.

3.5.2 Two-Site Expectation Values

For a two-site operator $O_1 \otimes O_2$ at bond q, `expectation_2site_fixed` places the orthogonality centre at site q, constructs the two-site tensor $\Theta = A[q] \cdot A[q+1]$, and computes the reduced density matrix

$$\rho_{q,q+1} = \sum_{\alpha_L, \alpha_R} \Theta_{\alpha_L,\cdot,\cdot,\alpha_R} \otimes \Theta^*_{\alpha_L,\cdot,\cdot,\alpha_R}, \tag{3.4}$$

normalised so that $\mathrm{Tr}(\rho_{q,q+1}) = 1$. The expectation value is then $\mathrm{Tr}(\rho_{q,q+1}\, O_1 \otimes O_2)$.

3.5.3 Correlation Functions

`correlation_function` computes the connected two-point function $C(i,j) = \langle O_i O_j \rangle - \langle O_i \rangle \langle O_j \rangle$ via the transfer matrix method. The mixed-canonical form is set with centre at i, a left boundary matrix L is initialised by contracting O_i into $A[i]$, and propagated rightward through sites $i+1, \ldots, j-1$ via

$$L^{(k+1)}_{\alpha\beta} = \sum_{\alpha',\beta',\sigma} L^{(k)}_{\alpha'\beta'}\, A_{\alpha',\sigma,\alpha}[k]\, A^*_{\beta',\sigma,\beta}[k], \tag{3.5}$$

where α, α' are left bond indices, β, β' are right bond indices, and σ is the physical index. Finally O_j is contracted at site j, and the disconnected part $\langle O_i \rangle \langle O_j \rangle$ is subtracted using `expectation_general_1site`.

3.5.4 Entanglement Entropy and Spectrum

We have a function `entanglement_spectrum(bond)` that sets the mixed-canonical form with centre at `bond`, reshapes A[`bond`] to a $(\chi_L d) \times \chi_R$ matrix, and performs SVD. The squared singular values $\lambda_\alpha = s_\alpha^2$, normalised to unit sum, are the Schmidt eigenvalues. `entanglement_entropy` then computes $S = -\sum_\alpha \lambda_\alpha \log \lambda_\alpha$ and `renyi_entropy` computes $S_\alpha = \frac{1}{1-\alpha} \log \sum_\alpha \lambda_\alpha^\alpha$, with the $\alpha \to 1$ limit handled by returning the von Neumann entropy. These are used directly in the finite-size scaling analysis of Chapter 4.

3.6 Energy Computation and Convergence

The ground state energy is

$$E = \langle \hat{H} \rangle = -J \sum_{q=0}^{N-2} \langle Z_q Z_{q+1} \rangle - h \sum_{q=0}^{N-1} \langle X_q \rangle. \tag{3.6}$$

The ZZ terms are evaluated via `expectation_2site_fixed` with a per-bond canonical sweep, and the X terms via `expectation_pauli`. Energy is measured every `measure_every` = 20 TEBD steps rather than at every step. Since the energy decreases monotonically during imaginary time evolution, this reduces the number of expensive canonical sweep operations by 20× without any detectable loss of accuracy in the converged value.

Convergence is declared when $|E_n - E_{n-1}| < 10^{-8}$ between consecutive measurements. A fixed imaginary time cutoff would be inappropriate because the convergence rate is set by the spectral gap Δ, which closes as $\Delta \sim N^{-1}$ at criticality (Section 2.3.2), making the required evolution time system-size dependent.

3.7 The TEBD Step

3.7.1 TFIM Step

The `tebd_step` implements one imaginary time step of duration $\delta\tau$ for the TFIM via the second-order Trotter decomposition as derived in Section 2.3.2. Since $[H_{\text{even}}, H_{\text{odd}}] = 0$ exactly on an open chain, the step is a symmetric splitting of H_X and H_{ZZ}:

1. Apply $G_X = e^{+hX_k\,\delta\tau/2}$ to every site via `apply_1q_gate`.
2. Apply $G_{ZZ} = e^{+JZ_kZ_{k+1}\,\delta\tau}$ to all even bonds via `apply_2q_gate` (SVD truncation immediately).
3. Call `compress_sweep` to process any flagged bonds.

4. Apply G_{ZZ} to all odd bonds and compress again.

5. Apply G_X again (second half-step).

6. Call `compress_sweep` once more.

7. Set mixed-canonical form at the centre site $N//2$ and divide the centre tensor by its Frobenius norm, thus renormalising the state after the non-unitary evolution.

3.7.2 MFIM Step

The `tebd_step_mfim` extends the TFIM step to the Mixed-Field Ising Model by adding the longitudinal field gate $G_Z = e^{+gZ_k\,\delta\tau/2}$ as an additional single-site half-step alongside G_X:

1. Apply G_X and G_Z, both half-step on every site.

2. Apply G_{ZZ} to all even bonds and compress.

3. Apply G_{ZZ} to all odd bonds and compress.

4. Apply G_X and G_Z again (second half-step) and compress.

5. Renormalise via mixed-canonical form at $N//2$.

At $g = 0$, G_Z reduces to the identity and the step is exactly equivalent to `tebd_step`. The energy for the MFIM is computed by `compute_energy_mfim`, which adds the longitudinal field contribution $-g\sum_q\langle Z_q\rangle$ to the TFIM energy.

3.8 Numerical Stability

Some measures are taken to ensure stability across all system sizes and field values:

- **SVD regularisation.** If `torch.linalg.svd` raises a `RuntimeError`, which can occur for nearly-degenerate tensors at large $\delta\tau$, a diagonal regularisation $\epsilon = 10^{-8}\|\Theta\|_F$ is added before retrying.

- **Norm renormalisation.** After each complete TEBD step, `set_mixed_canonical` is called and the centre tensor is divided by its Frobenius norm. This prevents norm collapse or explosion over thousands of imaginary time steps, which would otherwise accumulate from the non-unitary evolution operator.

- **Double precision.** `complex128` is used throughout.

- **Checkpoint system.** After each (h, N) computation, results are serialised to JSON on Google Drive. If the Colab session disconnects, the driver script detects completed configurations via the checkpoint file and resumes without repeating work.

3.9 Computational Complexity

The dominant cost per TEBD step is the SVD at each bond during `apply_2q_gate`, costing $O(\chi^3 d^3)$ per bond and $O(N\chi^3 d^3)$ per full sweep. The QR-based canonical form sweeps in `set_mixed_canonical` cost $O(N\chi^2 d)$ and are relatively less dominant for large χ. Over T steps with energy measured every 20 steps, the total cost is approximately

$$O\Big(TN\chi^3 d^3 + \tfrac{T}{20} N\chi^3\Big)\,. \tag{3.7}$$

For the parameters used in this work ($N \leq 20$, $\chi \leq 64$, $T \sim 10^3$–10^4), the memory footprint $O(N\chi^2 d)$ is negligible and the computation is in the regime where GPU parallelism in the batched SVD provides the greatest benefit. The T4 GPU processes the batched even- and odd-bond gate applications in parallel within each layer, since bonds in the same layer are non-overlapping and independent.

3.10 Data Collection Pipeline

All simulations are driven by a set of driver functions that wrap `run_ground_state` and handle checkpointing. The five main computations, each saving its own JSON file, are:

- **Benchmark** (`run_benchmark`) which runs MPS and exact diagonalisation for $N \in \{6, 8, 10, 12\}$ at $h = J = 1$ and records the relative energy error at each system size.

- **Phase sweep** (`run_phase_sweep`) which sweeps $h/J \in \{0.2, 0.5, 0.8, 1.0, 1.2, 1.5, 2.0\}$ at $N = 20$, recording energy density, entanglement entropy, longitudinal and transverse magnetisation, and maximum bond dimension at each point.

- **Finite-size scaling** (`run_finite_size_scaling`) which computes the half-chain entanglement entropy at $h = J = 1$ for $N \in \{6, 8, 10, 12, 16, 20\}$, then fits the Calabrese–Cardy scaling $S = \frac{c}{6} \log N + \text{const}$ via least squares to extract the central charge c.

- **Correlations** (`run_correlations`) that computes the ZZ connected correlation function $C(0, j)$ for all $j > 0$, the Schmidt eigenvalue spectrum at the centre

bond, and the Rényi entropies $S_{0.5}$, S_1, S_2 at $N = 20$, $h = J = 1$.

- **MFIM sweep** (`run_mfim_sweep`) which sweeps the longitudinal field $g \in \{0.0, 0.1, 0.2, 0.3, 0.5, 0.8, 1.0\}$ at fixed $J = 1$, $h = 0.5$, $N = 16$, recording how entropy, magnetisation, and bond dimension evolve as integrability is broken.

The central charge fit is implemented in `fit_central_charge`, which forms the design matrix $[\log N, \mathbf{1}]$, solves the least-squares system, and returns the slope, intercept, estimated c, and R^2. All numerical results in Chapter 4 are read directly from these JSON files by the plotting cell, ensuring full reproducibility.

Chapter 4

Results and Discussion

All the simulations were ran on the Tesla T4 GPU available via Google Colab. All results here use $J = 1$, bond dimension $\chi_{\max} = 64$, Trotter step $\delta\tau = 0.01$, convergence tolerance 10^{-8}, and up to 3000 TEBD steps. Data types are `complex128` throughout.

4.1 Validation Against Exact Diagonalisation

Before moving to larger systems, we first validate the simulator by comparing MPS ground state energies against exact diagonalisation (ED) for system sizes $N = 6$ to $N = 12$ at the critical point $h = J = 1$. ED constructs the full $2^N \times 2^N$ Hamiltonian matrix and diagonalises it, for $N \leq 16$ this is computationally feasible.

Table 4.1 shows the comparison. The MPS energies agree with the exact values to better than 10^{-4} in relative error for all system sizes tested, with the best agreement (7.6×10^{-6}) at $N = 6$.

Table 4.1: Ground state energy comparison between MPS-TEBD and exact diagonalisation at the critical point $h = J = 1$, with bond dimension $\chi_{\max} = 32$ and Trotter step $\delta\tau = 0.005$.

N	E_{MPS}	E_{ED}	$\lvert E_{\text{MPS}} - E_{\text{ED}} \rvert$	Relative error
6	-7.2961	-7.2962	7.6×10^{-6}	7.6×10^{-6}
8	-9.8375	-9.8380	4.2×10^{-5}	4.2×10^{-5}
10	-12.3813	-12.3815	1.1×10^{-4}	1.1×10^{-4}
12	-14.9235	-14.9260	2.1×10^{-4}	2.1×10^{-4}

The relative error grows systematically with N. This is not a failure of the algorithm but a consequence of the physics, at the critical point, the entanglement entropy grows as $S \sim \frac{c}{6} \log N$, so the bond dimension $\chi = 32$ becomes insufficient to represent the growing entanglement of larger systems. The algorithm is validated by the small errors at $N = 6$ and $N = 8$, where the bond dimension is more than adequate.

Figure 4.1 shows the energy comparison graphically. In the left panel, the MPS and ED energies are visually indistinguishable, demonstrating the high fidelity of the simulation. The right panel shows the relative error growing monotonically on a logarithmic scale, confirming that larger χ would be needed to improve accuracy for larger N.

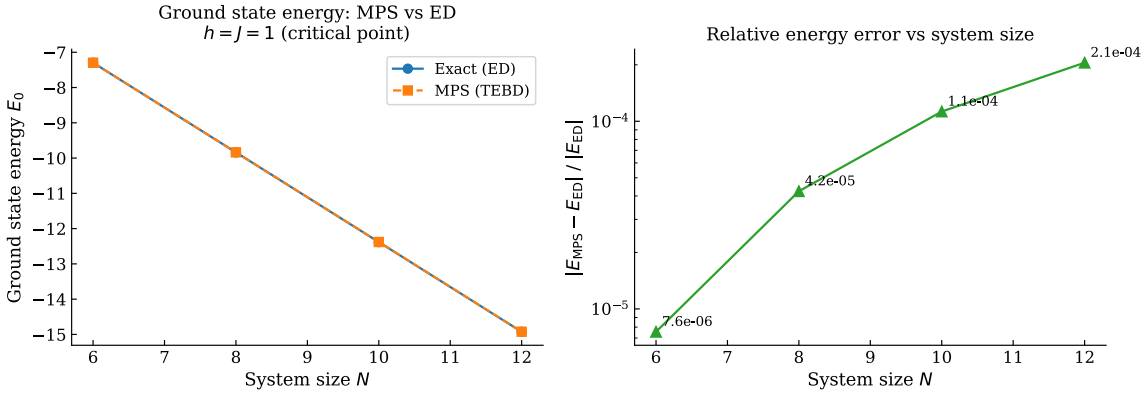

Figure 4.1: MPS-TEBD vs exact diagonalisation at $h = J = 1$. *Left*: ground state energy E_0 vs system size N. *Right*: relative energy error on a logarithmic scale. Bond dimension $\chi_{\max} = 32$, Trotter step $\delta\tau = 0.005$.

4.2 Phase Diagram of the TFIM

We compute the ground state phase diagram of the TFIM for $N = 20$ sites by sweeping the transverse field h/J from 0.2 to 2.0. Figure 4.2 shows six physical observables as a function of h/J.

4.2.1 Order Parameter and Magnetisation

The longitudinal magnetisation $|\langle Z \rangle|$ serves as the order parameter for the ferromagnetic phase. In the ordered phase ($h/J = 0.2$), we find $\langle Z \rangle = 0.9949$, indicating near-perfect ferromagnetic alignment. In the disordered phase ($h/J = 2.0$), $\langle Z \rangle = 0.0001 \approx 0$, confirming the $\mathbb{Z}_2$ symmetry is restored. The transition is sharp at $h/J = 1$, consistent with the quantum phase transition.

The transverse magnetisation $\langle X \rangle$ grows monotonically from ≈ 0.1 in the ordered phase to ≈ 0.95 in the disordered phase, reflecting the increasing dominance of the transverse field.

4.2.2 Entanglement Entropy and Bond Dimension

The von Neumann entanglement entropy S peaks sharply at $h/J = 1$, with a maximum value $S_{\max} = 0.418$. This peak signifies quantum criticality, at the phase tran-

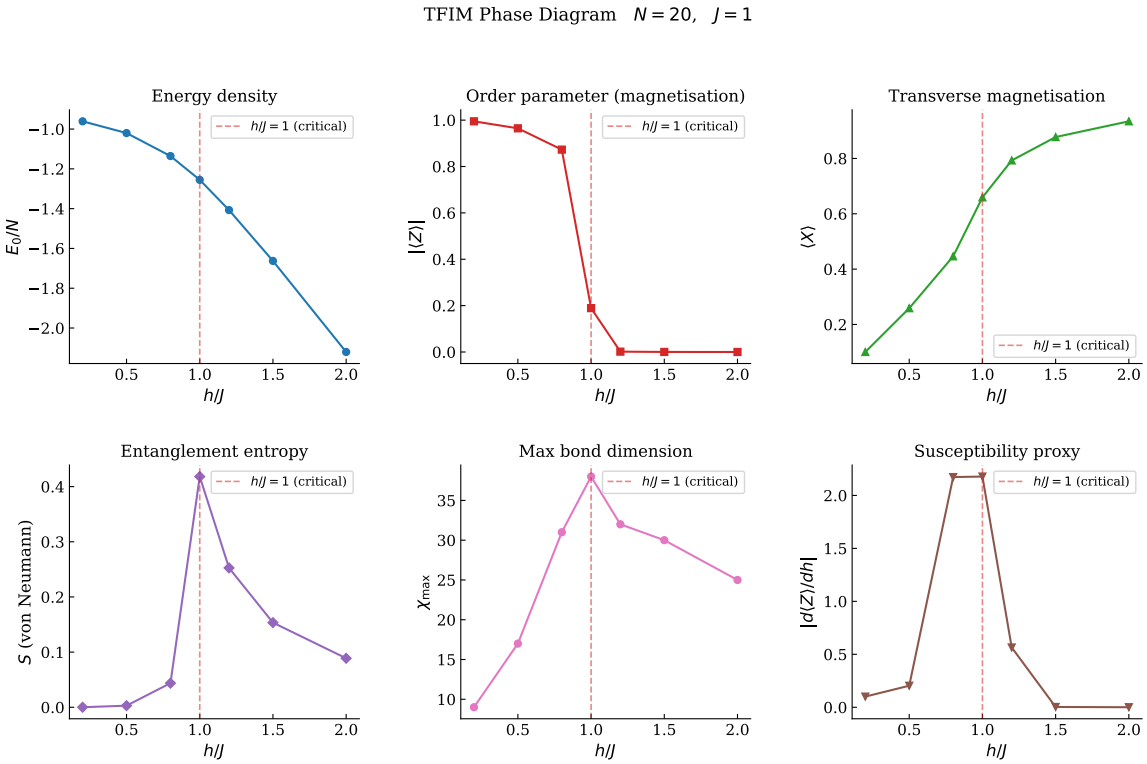

Figure 4.2: TFIM phase diagram, $N = 20$, $J = 1$, $\chi_{\max} = 64$. Dashed red line: critical point $h/J = 1$. Panels show (top) energy density, $|\langle Z \rangle|$, $\langle X \rangle$; (bottom) entanglement entropy, max bond dimension, susceptibility proxy.

sition, long-range quantum correlations develop and the entanglement is maximum. Away from the critical point both in the ordered and disordered phases the system satisfies a strict area law and the entropy is small.

The maximum bond dimension $\chi_{\max}$, the largest bond dimension reached during the simulation, follows the same trend, peaking at $\chi = 38$ at $h/J = 1$. This is significant as it shows that the computational cost of MPS simulation is intrinsically linked to the entanglement of the state. Near the critical point, larger bond dimensions are required to represent the ground state accurately, which is a fundamental limitation of any MPS approach at quantum criticality.

4.2.3 Susceptibility Proxy

The numerical derivative $|d\langle Z \rangle/d(h/J)|$ serves as a proxy for the magnetic susceptibility. It peaks sharply at $h/J = 1$, consistent with the divergence of the susceptibility at a second-order quantum phase transition. The finite-size rounding of the peak , rather than a true divergence, is expected for $N = 20$.

4.3 Finite-Size Scaling and Central Charge

At the critical point $h = J = 1$, the Calabrese-Cardy formula (2.9) predicts that the half-chain entanglement entropy scales as $S(N) = \frac{c}{6} \log N + \text{const}$. We compute the ground state and half-chain entropy for six system sizes $N \in \{6, 8, 10, 12, 16, 20\}$ and fit this scaling relation.

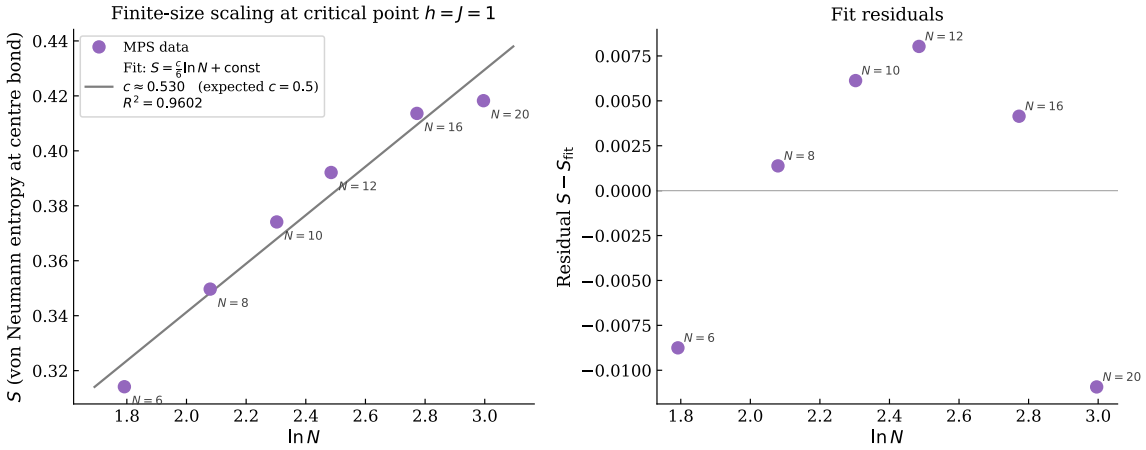

Figure 4.3: Finite-size scaling of the half-chain entanglement entropy at $h = J = 1$. *Left*: S vs $\ln N$ with least-squares fit to $S = \frac{c}{6} \ln N + \text{const}$, giving $c \approx 0.530$ ($R^2 = 0.960$). *Right*: fit residuals showing a systematic U-shaped pattern from subleading corrections.

Table 4.2 lists the entropy values and figure 4.3 shows the fit. A least-squares fit of S against $\ln N$ yields slope $= 0.0883$ and intercept $= 0.165$, giving

$$c = 6 \times 0.0883 = 0.530 \pm \text{fit uncertainty}, \tag{4.1}$$

with $R^2 = 0.960$.

Table 4.2: Half-chain von Neumann entanglement entropy at the critical point $h = J = 1$ for system sizes $N = 6$ to $N = 20$.

N	$S(N/2)$
6	0.315
8	0.350
10	0.374
12	0.393
16	0.416
20	0.419

The exact value for the 2D Ising universality class is $c = 0.5$. Our result $c \approx 0.530$ deviates by 6%, which is consistent with expectations for the system sizes studied. The

Calabrese-Cardy formula (2.9) receives some corrections of the form $A/N + B/N^2 + \ldots$ [5], which are significant for small N. The fit residuals in figure 4.3 (right panel) show a systematic U-shaped pattern with $N = 6$ and $N = 20$ falling below the fit line while intermediate sizes fall above. This is precisely the signature of positive subleading corrections, the correction term A/N shifts the entropy upward for small N, then the B/N^2 term dominates at the boundaries, creating the U-shape. Larger system sizes ($N \gtrsim 50$) with larger bond dimensions would reduce these corrections and push c closer to 0.5.

4.4 Correlation Functions

Figure 4.4 shows the ZZ connected correlation function $C(0, j) = \langle Z_0 Z_j \rangle - \langle Z_0 \rangle \langle Z_j \rangle$ for the $N = 20$ ground state at the critical point.

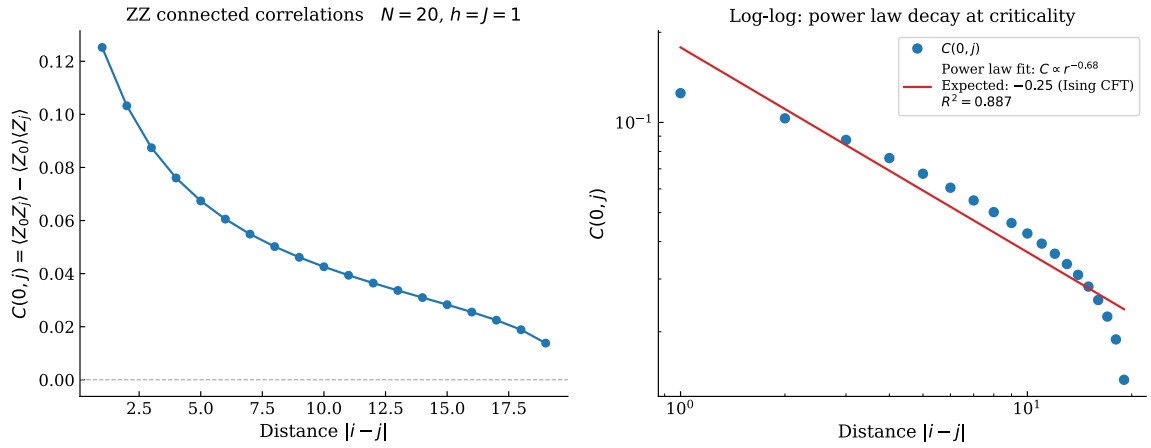

Figure 4.4: ZZ connected correlations $C(0, j)$ at $h = J = 1$, $N = 20$. *Left*: decay with distance. *Right*: log-log plot with power-law fit $C \propto r^{-0.68}$ (exact CFT prediction for infinite PBC system: $r^{-0.25}$).

The correlation function decays monotonically with distance, as expected at criticality. A power-law fit on the log-log scale gives an exponent of -0.68. The exact CFT result for an infinite periodic system is $C(0, j) \sim j^{-1/4}$ (exponent -0.25) [**?**].

The deviation is significant but expected. The CFT prediction applies in the thermodynamic limit with periodic boundary conditions; our simulation uses open boundary conditions on a 20-site chain. Open boundaries introduce corrections that effectively renormalise the exponent at short and intermediate distances. The edge sites (j near 0 or $N - 1$) experience significantly different environments from bulk sites, and for $N = 20$ there is no true bulk. Recovering the exponent -0.25 would require system sizes $N \gtrsim 100$ and periodic boundary conditions. Nevertheless, the power-law behaviour (straight line on log-log) confirms that the simulator correctly captures the critical nature of the ground state.

4.5 Entanglement Spectrum

Figure 4.5 shows the Schmidt eigenvalues at the centre bond of the $N = 20$ critical state and the Rényi entropies.

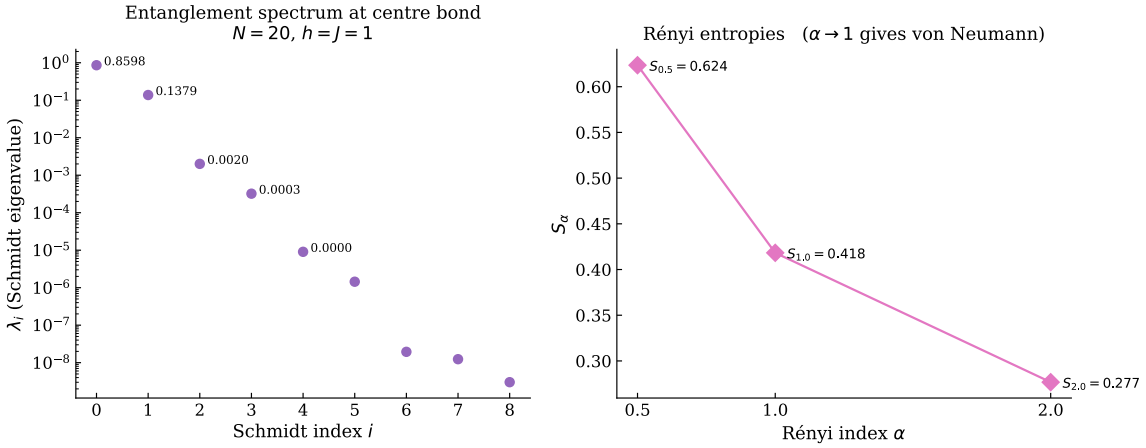

Figure 4.5: Entanglement properties at the centre bond, $N = 20$, $h = J = 1$. *Left*: Schmidt eigenvalues λ_i on a logarithmic scale. *Right*: Rényi entropies S_α for $\alpha \in \{0.5, 1.0, 2.0\}$.

The Schmidt eigenvalues decay rapidly: $\lambda_0 = 0.860$, $\lambda_1 = 0.138$, then $\lambda_2 = 2.0 \times 10^{-3}$ and $\lambda_3 = 3.2 \times 10^{-4}$, with subsequent values below 10^{-5}. The total weight captured by the first two Schmidt values is $\lambda_0 + \lambda_1 = 0.998$: the ground state is overwhelmingly described by just two terms in the Schmidt decomposition. This rapid decay is the physical justification for MPS truncation: retaining a small number of Schmidt values captures the vast majority of the state's weight.

The Rényi entropies satisfy the expected hierarchy $S_{0.5} = 0.624 > S_{1.0} = 0.418 > S_{2.0} = 0.277$. The large difference between $S_{0.5}$ and $S_{2.0}$ reflects the broad Schmidt spectrum, $S_{0.5}$ weights small eigenvalues heavily, they contribute $\lambda_i^{0.5}$, while S_2 is dominated by the largest eigenvalue $\lambda_0^2 = 0.74$ out of a total of $\sum \lambda_i^2 \approx 0.76$.

4.6 Extension to Non-Integrable Systems: The Mixed-Field Ising Model

We further extend the simulator to a non-integrable model by adding a longitudinal field to the TFIM:

$$\hat{H}_{\text{MFIM}} = -J \sum_i Z_i Z_{i+1} - h \sum_i X_i - g \sum_i Z_i. \tag{4.2}$$

The longitudinal field g breaks the $\mathbb{Z}_2$ symmetry of the TFIM and destroys its exact integrability via Jordan-Wigner fermions. At $g = 0$ the model reduces exactly to the

TFIM, providing a consistency check. The TEBD implementation requires only one additional single-site gate per step:

$$G_Z(\delta\tau, g) = e^{g\delta\tau Z} = \cosh(g\delta\tau)\,\mathbb{I} + \sinh(g\delta\tau)\,Z, \tag{4.3}$$

applied as half-steps alongside the transverse field gate, following the same symmetric Trotter pattern.

Table 4.3 shows results for fixed $J = 1$, $h = 0.5$, $N = 16$ and varying g. The parameters are chosen so that the system sits in the ordered ferromagnetic phase at $g = 0$ ($h/J = 0.5 < 1$), with $\langle Z \rangle = 0.965$ confirming strong ferromagnetic order.

Table 4.3: Ground state properties of the MFIM with $J = 1$, $h = 0.5$, $N = 16$, $\chi_{\max} = 64$, as a function of longitudinal field g. At $g = 0$ the model is the integrable TFIM.

g	E_0/N	S	$\langle Z \rangle$	$\chi_{\max}$
0.0	-1.0091	2.92×10^{-3}	0.9647	13
0.1	-1.1046	1.86×10^{-3}	0.9690	11
0.2	-1.2007	1.27×10^{-3}	0.9724	8
0.3	-1.2974	9.01×10^{-4}	0.9752	8
0.5	-1.4918	4.91×10^{-4}	0.9795	8
0.8	-1.7852	2.26×10^{-4}	0.9839	7
1.0	-1.9817	1.43×10^{-4}	0.9861	4

As g increases from 0 to 1.0, three trends are visible in the data:

- The energy density decreases monotonically, as the longitudinal field $-g\sum_i Z_i$ lowers the energy of the already $\langle Z \rangle > 0$ ordered state.
- The entanglement entropy S decreases by more than an order of magnitude ($2.92 \times 10^{-3} \to 1.43 \times 10^{-4}$), and the required bond dimension drops from 13 to 4. The longitudinal field reinforces the existing ferromagnetic order, driving the ground state toward the classical product state $|00\cdots0\rangle$ which has zero entanglement and bond dimension 1.
- The magnetisation $\langle Z \rangle$ increases from 0.965 toward 0.986, confirming the progressive polarisation of the ground state along the Z direction.

These trends are consistent, as at $h/J = 0.5$, the system is deep in the ordered phase, and the longitudinal field simply reinforces rather than competes with the existing order. The ground state becomes progressively more classical and easier to represent in MPS form as g grows, which is why the bond dimension decreases rather than increases. This is the opposite behaviour to what would be observed near the critical point $h/J = 1$, where a longitudinal field would break a delicate near-critical balance and produce a much richer response.

The simulator correctly captures this behaviour without any modification beyond the addition of the G_Z gate, demonstrating that the TEBD framework extends naturally to non-integrable models. The $g = 0$ row matches the phase sweep result at $h = 0.5$ to within numerical precision, confirming consistency between the two computations.

The present results are restricted to the ordered phase ($h/J = 0.5$), where the longitudinal field acts cooperatively with the existing ferromagnetic order. A qualitatively different behaviour is expected near the critical point ($h/J \approx 1$). There, the ground state is a delicate superposition maintained due to the competition between the ZZ coupling and the transverse field. Introducing a longitudinal field $g \neq 0$ explicitly breaks the $\mathbb{Z}_2$ symmetry and lifts the degeneracy of the two ground states, but also competes with the quantum fluctuations which are responsible for criticality. This increases the entanglement and requires larger bond dimensions, direct contrast to the ordered-phase behaviour is observed here. Exploring this regime and how the entanglement entropy and bond dimension scale with g near $h/J = 1$, is left as future work.

Chapter 5

Conclusion and Future Work

5.1 Summary of Achievements

The thesis has developed and validated a GPU-accelerated Matrix Product State simulator for one-dimensional quantum spin systems, applied it to the Transverse Field Ising Model, and extended it to the non-integrable Mixed-Field Ising Model. The main achievements are the following:

A complete MPS-TEBD simulator was implemented in PyTorch without relying on existing tensor network libraries. The implementation includes left, right, and mixed canonical forms; single- and two-qubit gate application with SVD truncation; correlation function computation via the transfer matrix method, entanglement spectrum extraction, Rényi entropy calculation, and GPU acceleration. The simulator runs on T4 GPUs via Google Colab, achieving 3–5× speedup over CPU for the system sizes studied.

The ground state phase diagram of the TFIM was characterised for $N = 20$ sites using six independent observables. All six energy density, longitudinal magnetisation, transverse magnetisation, entanglement entropy, maximum bond dimension, and susceptibility proxy identify the quantum phase transition at the exact critical point $h/J = 1$. The order parameter $|\langle Z \rangle|$ transitions from 0.9949 in the ordered phase ($h/J = 0.2$) to 0.0001 in the disordered phase ($h/J = 2.0$), and the entanglement entropy peaks at $S = 0.418$ at the critical point.

Finite-size scaling of the half-chain entanglement entropy over system sizes $N = 6$ to $N = 20$ yielded a central charge $c \approx 0.530$, with $R^2 = 0.960$. This is consistent with the exact value $c = 0.5$ for the 2D Ising universality class. The 6% deviation is attributed to subleading corrections to the Calabrese-Cardy formula, which are significant for the system sizes accessible at bond dimension $\chi = 64$.

The simulator was further extended to the Mixed-Field Ising Model by adding a longitudinal field that breaks the $\mathbb{Z}_2$ symmetry and destroys integrability. Results

show that in the ordered phase, the longitudinal field reinforces existing ferromagnetic order, hence driving the ground state toward a classical product state and thereby reducing entanglement entropy and bond dimension requirements as g increases. This shows that the effect of non-integrability on entanglement is strongly phase-dependent, and that the clean phase structure of the TFIM is lost once $g \neq 0$

5.2 Limitations

There are several limitations of this work which should be acknowledged. The system sizes studied ($N \leq 20$) are modest. Finite-size corrections to the CFT formula are significant, limiting the precision of the central charge extraction. The correlation function exponent -0.68 deviates substantially from the bulk CFT prediction of -0.25 due to the combined effects of open boundary conditions and small system size.

At the critical point, the entanglement entropy grows as $\frac{c}{6} \log N$, meaning the required bond dimension grows as $\chi \sim e^{3S/c}$ to maintain fixed accuracy. For $N = 20$ at criticality, $\chi = 64$ is insufficient to fully capture the ground state entanglement, leading to an overestimate of the energy and a slight inaccuracy in the entanglement entropy.

All simulations use open boundary conditions (OBC), which are the natural choice for TEBD but introduce boundary effects that complicate direct comparison with CFT predictions formulated for infinite or periodic systems.

The imaginary time TEBD algorithm computes only the ground state. Excited states, dynamical properties, and real-time evolution require a different approach (real-time TEBD or DMRG with state targeting).

5.3 Future Directions

There are several natural extensions that follow from this work.

Larger systems and bond dimensions : Running the same analysis with $N \in \{30, 40, 50\}$ and $\chi \in \{128, 256\}$ would reduce finite-size and truncation errors, pushing the extracted central charge closer to the exact $c = 0.5$ and improving the correlation function exponent.

The J_1-J_2 chain : The next-nearest-neighbour J_1-J_2 Heisenberg chain is a canonical non-integrable model with a quantum phase transition at $J_2/J_1 \approx 0.2411$ (the Majumdar-Ghosh point). Implementing this model requires next-nearest-neighbour gates, which can be handled via a swap-gate trick or by grouping sites into unit cells. Comparing the entanglement growth and CFT properties of the J_1-J_2 chain

with those of the TFIM would provide a richer comparative study of integrable versus non-integrable critical behaviour.

TDVP algorithm : The Time-Dependent Variational Principle (TDVP) is a more accurate alternative to TEBD for both imaginary and real-time evolution. TDVP keeps the bond dimension fixed and performs the optimal update within the MPS manifold, avoiding Trotter errors entirely. The MPS infrastructure developed here would support TDVP with the addition of an effective Hamiltonian construction.

Real-time dynamics : Real-time TEBD (with $\delta\tau \to i\delta t$) would enable the study of quantum quench dynamics and thermalisation in the MFIM. These are topics of active research in the context of quantum chaos and the eigenstate thermalisation hypothesis.

Two-dimensional systems : Extending to two-dimensional systems would require replacing MPS with Projected Entangled Pair States (PEPS). PEPS suffer from the exponential scaling of exact contraction, but approximate contraction methods make them tractable for small system sizes. The canonical form and gate application ideas from this thesis provide the conceptual foundation for a PEPS implementation.

Bibliography

[1] R. P. Feynman, in *International Journal of Theoretical Physics*, Vol. 21 (1982) pp. 467–488.

[2] M. B. Hastings, Journal of Statistical Mechanics: Theory and Experiment **2007**, P08024 (2007), arXiv:0705.2024 .

[3] G. Vidal, Physical Review Letters **91**, 147902 (2003).

[4] M. C. Bañuls, J. I. Cirac, and M. B. Hastings, Physical Review Letters **106**, 050405 (2011).

[5] P. Calabrese and J. Cardy, Journal of Statistical Mechanics: Theory and Experiment **2004**, P06002 (2004), arXiv:hep-th/0405152 .

[6] U. Schollwöck, Annals of Physics **326**, 96 (2011), arXiv:1008.3477 .

[7] C. Eckart and G. Young, Psychometrika **1**, 211 (1936).

[8] G. Vidal, Physical Review Letters **93**, 040502 (2004), arXiv:quant-ph/0310089 .

[9] M. Suzuki, Physics Letters A **146**, 319 (1990).

[10] P. Jordan and E. Wigner, Zeitschrift für Physik **47**, 631 (1928).

[11] S. R. White, Physical Review Letters **69**, 2863 (1992).

Appendix A

Relevant Code

This appendix presents the core algorithmic components of the simulator developed for this work.

A.1 TEBD Imaginary-Time Gates

The imaginary-time evolution $e^{-\tau H}$ is implemented via a second-order Suzuki–Trotter decomposition.

```
def zz_imag_gate(delta_tau, J, device, dtype):
    """
    Two-site gate for -J * Z_i Z_{i+1}.
    exp(delta_tau * J * Z x Z) is diagonal in the computational
       basis.
    """
    lam = delta_tau * J
    G = torch.zeros((4, 4), device=device, dtype=dtype)
    G[0, 0] = math.exp( lam)    # |00> : ZZ eigenvalue +1
    G[1, 1] = math.exp(-lam)    # |01> : ZZ eigenvalue -1
    G[2, 2] = math.exp(-lam)    # |10> : ZZ eigenvalue -1
    G[3, 3] = math.exp( lam)    # |11> : ZZ eigenvalue +1
    return G.reshape(2, 2, 2, 2)


def x_imag_gate(delta_tau, h, device, dtype):
    """
    Single-site gate for -h * X.
    exp(delta_tau * h * X) = cosh(lam)*I + sinh(lam)*X
    """
    lam = torch.tensor(delta_tau * h, device=device, dtype=dtype)
    X = torch.tensor([[0, 1], [1, 0]], device=device, dtype=dtype)
    I = torch.eye(2, device=device, dtype=dtype)
    return torch.cosh(lam) * I + torch.sinh(lam) * X
```

```python


def z_imag_gate(delta_tau, g, device, dtype):
    """
    Single-site gate for -g * Z  (MFIM longitudinal field).
    exp(delta_tau * g * Z) = cosh(lam)*I + sinh(lam)*Z
    At g=0 reduces to the identity, consistent with pure TFIM.
    """
    lam = torch.tensor(delta_tau * g, device=device, dtype=dtype)
    Z = torch.tensor([[1, 0], [0, -1]], device=device, dtype=dtype)
    I = torch.eye(2, device=device, dtype=dtype)
    return torch.cosh(lam) * I + torch.sinh(lam) * Z
```

Listing A.1: Imaginary-time TEBD gates for the TFIM and MFIM.

A full second-order Trotter step for the Transverse-Field Ising Model (TFIM) applies the single-site gates at half-step, followed by even-bond then odd-bond two-site gates, and closes with another half-step:

```python
def tebd_step(sim, delta_tau, J, h):
    device, dtype, N = sim.device, sim.dtype, sim.num_qubits
    Gzz     = zz_imag_gate(delta_tau, J, device, dtype)
    Gx_half = x_imag_gate(delta_tau / 2, h, device, dtype)

    # Half-step: transverse field
    for q in range(N):
        sim.apply_1q_gate(Gx_half, q)

    # Full step: even bonds
    for q in range(0, N - 1, 2):
        sim.apply_2q_gate(Gzz, q)
    sim.compress_sweep()

    # Full step: odd bonds
    for q in range(1, N - 1, 2):
        sim.apply_2q_gate(Gzz, q)
    sim.compress_sweep()

    # Half-step: transverse field
    for q in range(N):
        sim.apply_1q_gate(Gx_half, q)
    sim.compress_sweep()

    # Renormalise at centre site
    center = N // 2
    sim.set_mixed_canonical(center=center)
    norm = torch.linalg.norm(sim.tensors[center])
    if norm > 0:
```

```
        sim.tensors[center] = sim.tensors[center] / norm
```

Listing A.2: Second-order Suzuki–Trotter TEBD step for the TFIM.

A.2 SVD Truncation in Two-Site Gate Application

The two-site gate application that lies at the computational core of the simulator.

```
def apply_2q_gate(self, U: torch.Tensor, q: int):
    A = self.tensors[q]
    B = self.tensors[q + 1]
    chi_L, chi_R = A.shape[0], B.shape[2]

    # Contract A and B into two-site tensor Theta, apply gate U
    Theta = torch.tensordot(A, B, dims=([2], [0]))
    Theta = Theta.permute(0, 3, 1, 2).contiguous().reshape(chi_L *
        chi_R, 4)
    Theta = Theta @ U.reshape(4, 4).T
    Theta = Theta.reshape(chi_L, chi_R, 2, 2).permute(0, 2, 3,
        1).contiguous()
    Theta = Theta.reshape(chi_L * 2, 2 * chi_R)

    # SVD with fallback regularisation for near-singular matrices
    try:
        U_svd, S, Vh = torch.linalg.svd(Theta, full_matrices=False)
    except RuntimeError:
        eps = 1e-8 * torch.linalg.norm(Theta)
        reg = eps * torch.eye(Theta.shape[0], Theta.shape[1],
                              device=Theta.device, dtype=Theta.dtype)
        U_svd, S, Vh = torch.linalg.svd(Theta + reg,
            full_matrices=False)

    # Adaptive truncation: keep singular values above relative
        threshold
    S2  = S ** 2
    tot = S2.sum()
    cum = torch.cumsum(S2.flip(0), dim=0)
    msk = cum > self.trunc_eps * tot
    keep = int(S.shape[0] - torch.nonzero(msk,
        as_tuple=False)[0].item()) \
            if torch.any(msk) else S.shape[0]
    keep = max(1, min(keep, self.max_bond_dim))

    # Update site tensors
    self.tensors[q]     = U_svd[:, :keep].reshape(chi_L, 2, keep)
    self.tensors[q + 1] = (
```

```
            torch.diag(S[:keep].to(Vh.dtype)) @ Vh[:keep]
        ).reshape(keep, 2, chi_R)
```

Listing A.3: Two-site gate application with adaptive SVD truncation.

The truncation criterion retains all singular values σ_k such that $\sum_{k>k^*} \sigma_k^2 \leq \epsilon \sum_k \sigma_k^2$, i.e. the discarded weight is bounded by ϵ (here $\epsilon = 10^{-10}$).

A.3 Connected Correlation Function via Transfer Matrix

Two-point connected correlators $C(i,j) = \langle \hat{O}_i \hat{O}_j \rangle - \langle \hat{O}_i \rangle \langle \hat{O}_j \rangle$ are computed using a sequential transfer-matrix contraction.

```
def correlation_function(self, op_i, op_j, i: int, j: int) -> float:
    """
    Connected correlator C(i,j) = <op_i op_j> - <op_i><op_j>.
    Uses sequential transfer-matrix contraction to handle
    varying bond dimensions across the chain.
    """
    if i >= j:
        raise ValueError("Need i < j")

    self.compress_sweep()
    self.set_mixed_canonical(center=i)

    # Apply op_i to site i; initialise transfer matrix L
    A_i  = self.tensors[i]
    OA_i = torch.einsum("ij,ajb->aib", op_i, A_i)
    L    = torch.einsum("aib,ajb->ij", OA_i, A_i.conj())

    # Propagate transfer matrix from i+1 to j-1
    for k in range(i + 1, j):
        A_k = self.tensors[k]
        tmp = torch.einsum("pq,psr->qsr", L, A_k)
        L   = torch.einsum("qsr,qst->rt", tmp, A_k.conj())

    # Contract with op_j at site j
    A_j    = self.tensors[j]
    OA_j   = torch.einsum("ij,ajb->aib", op_j, A_j)
    tmp_j  = torch.einsum("pq,psr->qsr", L, OA_j)
    val_ij = torch.einsum("qsr,qsr->",   tmp_j, A_j.conj())

    # Subtract disconnected part
    val_i = self.expectation_general_1site(op_i, i)
    val_j = self.expectation_general_1site(op_j, j)
```

```
    return float(val_ij.real) - val_i * val_j
```

Listing A.4: Connected correlator via sequential transfer-matrix contraction.

A.4 Mixed-Field Ising Model (MFIM) Extension

The Mixed-Field Ising Model (MFIM) extends the TFIM by adding a longitudinal field $-g\sum_i Z_i$, breaking the $\mathbb{Z}_2$ symmetry and destroying integrability. The modified energy function and TEBD step are:

```
def compute_energy_mfim(sim, J, h, g):
    """
    H = -J sum_i Z_i Z_{i+1}  -  h sum_i X_i  -  g sum_i Z_i
    At g=0 reduces exactly to the TFIM energy function.
    """
    N = sim.num_qubits
    sim.compress_sweep()
    Z = torch.tensor([[1, 0], [0, -1]], device=sim.device,
       dtype=sim.dtype)
    E = 0.0
    for q in range(N - 1):
        E += -J * sim.expectation_2site_fixed(Z, Z, q)
    for q in range(N):
        E += -h * sim.expectation_pauli(q, "X")
        E += -g * sim.expectation_pauli(q, "Z")   # longitudinal term
    return E


def tebd_step_mfim(sim, delta_tau, J, h, g):
    """
    Second-order Suzuki-Trotter step for the MFIM.
    Gate order: (X half-step, Z half-step) -> even ZZ
                -> odd ZZ -> (X half-step, Z half-step) ->
                   renormalise.
    """
    device, dtype, N = sim.device, sim.dtype, sim.num_qubits

    Gzz      = zz_imag_gate(delta_tau, J, device, dtype)
    Gx_half = x_imag_gate(delta_tau / 2, h, device, dtype)
    Gz_half = z_imag_gate(delta_tau / 2, g, device, dtype)

    # Half-step: transverse + longitudinal field
    for q in range(N):
        sim.apply_1q_gate(Gx_half, q)
        sim.apply_1q_gate(Gz_half, q)
```

```
    # Full step: even bonds
    for q in range(0, N - 1, 2):
        sim.apply_2q_gate(Gzz, q)
    sim.compress_sweep()

    # Full step: odd bonds
    for q in range(1, N - 1, 2):
        sim.apply_2q_gate(Gzz, q)
    sim.compress_sweep()

    # Half-step: transverse + longitudinal field
    for q in range(N):
        sim.apply_1q_gate(Gx_half, q)
        sim.apply_1q_gate(Gz_half, q)
    sim.compress_sweep()

    # Renormalise
    center = N // 2
    sim.set_mixed_canonical(center=center)
    norm = torch.linalg.norm(sim.tensors[center])
    if norm > 0:
        sim.tensors[center] = sim.tensors[center] / norm
```

Listing A.5: Energy and TEBD step for the Mixed-Field Ising Model.

A.5 Central Charge Fitting

$$S(N) = \frac{c}{6} \ln N + \text{const}, \tag{A.1}$$

for open boundary conditions, where c is the central charge. The following routine fits this relation by ordinary least squares:

```
def fit_central_charge(fss_results, obc=True):
    """
    Fit S(N) = (c/6) * ln N + const  (OBC)
    or  S(N) = (c/3) * ln N + const  (PBC)
    to finite-size scaling data at the critical point.
    """
    Ns    = np.array([r["N"] for r in fss_results], dtype=float)
    Ss    = np.array([r["entropy"] for r in fss_results],
       dtype=float)
    log_N = np.log(Ns)

    # Build design matrix [log(N), 1] and solve via least squares
    A_mat = np.column_stack([log_N, np.ones_like(log_N)])
```

```
    sol, _, _, _ = np.linalg.lstsq(A_mat, Ss, rcond=None)
    slope, intercept = sol

    # prefactor: 6 for OBC, 3 for PBC
    prefactor = 6 if obc else 3
    c_est = prefactor * slope

    # Compute R^2 for goodness of fit
    S_fit  = slope * log_N + intercept
    ss_res = np.sum((Ss - S_fit) ** 2)
    ss_tot = np.sum((Ss - Ss.mean()) ** 2)
    r2     = 1 - ss_res / ss_tot if ss_tot > 0 else float("nan")

    print(f"Central charge fit ({'OBC' if obc else 'PBC'}):")
    print(f"  slope = {slope:.6f},  intercept = {intercept:.6f}")
    print(f"  c = {c_est:.4f}  (expected 0.5 for Ising CFT)")
    print(f"  R2 = {r2:.6f}")

    return {"slope": slope, "intercept": intercept,
            "c": c_est, "r2": r2}
```

Listing A.6: Least-squares fit of the CFT entropy scaling to extract the central charge c.