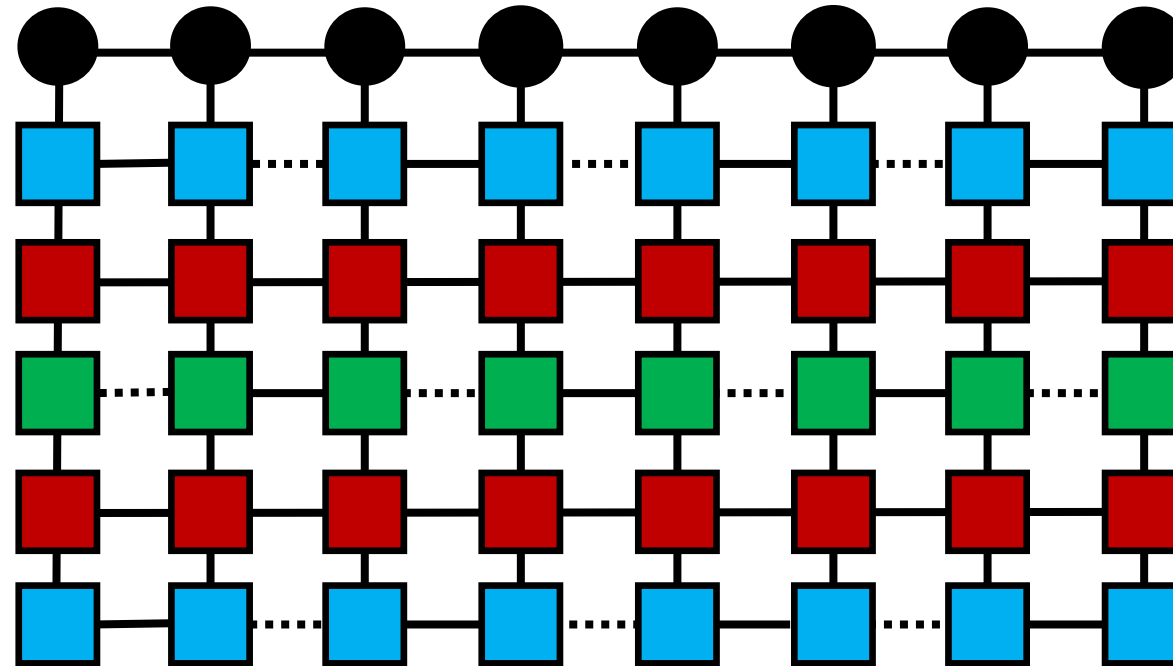


An Introduction to Matrix Product State Algorithms

Nicholas Woodford, Supervisor: Takis Anglides



DESY.
QUANTUM

Center for
Quantum Technology
and Applications

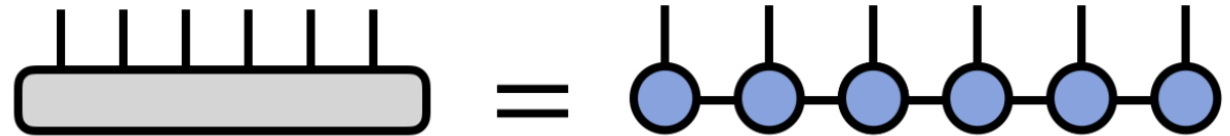


Matrix Product States

Tensor Networks

Matrix

$$M_{ij}$$

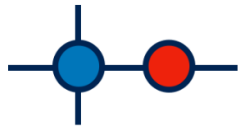


3-legged Tensor

$$T_{ijk}$$

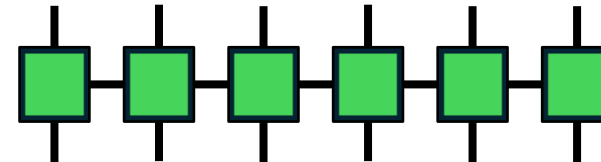


$$T^{s_1 s_2 s_3 s_4 s_5 s_6} = \sum_{\{\alpha\}} A_{\alpha_1}^{s_1} A_{\alpha_1 \alpha_2}^{s_2} A_{\alpha_2 \alpha_3}^{s_3} A_{\alpha_3 \alpha_4}^{s_4} A_{\alpha_4 \alpha_5}^{s_5} A_{\alpha_5}^{s_6}$$



=

$$\sum_k T_{ijkl} V_{km}$$



N – Length of MPS

d – size of the physical dimensions e.g. 2 for spins

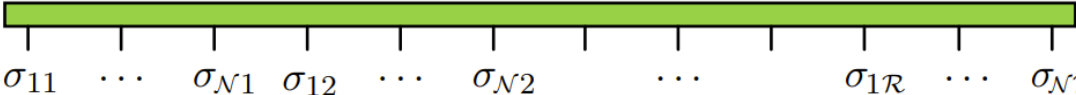
D – size of largest bond dimensions

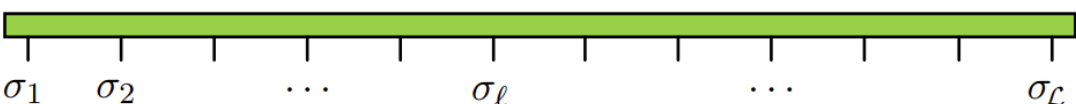
- Approximate tensor of size d^N with MPS of size NdD^2
- Bond dimensions transfer information between one site to the next
- Smaller bond dimension = More compression
- Larger bond dimensions = More ‘entanglement’

Matrix Product States

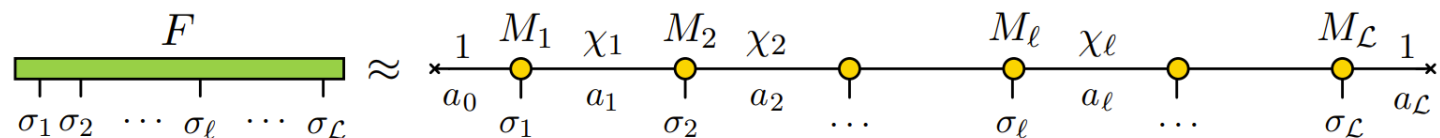
Tensor Cross Interpolation

- Can discretise function onto a tensor
- We can express any tensor as an MPS
- Compression of the MPS representation for a given error threshold depends on the structure of the function
- MPS of function allows for applications such as integration or Fourier transform

$$F_{\sigma} = f(\mathbf{x}(\sigma)) =$$


$$=$$


$$F_{\sigma} \approx \tilde{F}_{\sigma} = \prod_{\ell=1}^{\mathcal{L}} M_{\ell}^{\sigma_{\ell}} = [M_1]_{1a_1}^{\sigma_1} [M_2]_{a_1a_2}^{\sigma_2} \cdots [M_{\mathcal{L}}]_{a_{\mathcal{L}-1}1}^{\sigma_{\mathcal{L}}},$$



Matrix Product States

Basic Operations

Tensor Contraction:

$$\begin{aligned}
 & \text{Diagram 1} = \langle \phi | \psi \rangle \\
 & = \text{Diagram 2} = \text{Diagram 3}
 \end{aligned}$$

The diagram shows the contraction of two 1D tensor networks. The first diagram consists of two horizontal rows of six circles each, with red circles on top and blue circles on bottom, connected by vertical lines. This is equal to the inner product $\langle \phi | \psi \rangle$. The second diagram shows the same structure with a vertical gray bar on the left side of the first column. The third diagram shows the same structure with a gray L-shaped bracket on the left side of the first column.

Expectation values:

$$\text{Diagram 1} = \langle \psi | \hat{O} | \psi \rangle$$

The diagram shows a 1D tensor network for an expectation value. It has three horizontal rows of six nodes each. The top row has red circles, the middle row has green squares, and the bottom row has blue circles. All nodes in adjacent rows are connected by vertical lines, and nodes in the same row are connected by horizontal lines. This is equal to the expectation value $\langle \psi | \hat{O} | \psi \rangle$.

Efficient contraction algorithms have time complexity $O(dD^3N)$

Canonical Forms:

$$\begin{aligned}
 & \text{Diagram 1} = \text{Diagram 2} = \text{Diagram 3} , \\
 & \text{Diagram 4} = \text{Diagram 5}
 \end{aligned}$$

The first part shows the canonical form of a two-site tensor contraction. Diagram 1 has two red circles connected by a horizontal line. Diagram 2 has a red circle, a blue circle labeled 'A', a blue circle labeled 'A⁻¹', and another red circle, all connected in series. Diagram 3 has two green circles connected by a horizontal line. The second part shows a contraction of two tensors. Diagram 4 has two blue circles, one labeled 'U[†]' and the other 'U', connected by a horizontal line and a vertical line that forms a loop. Diagram 5 is a single curved line representing the contraction of the two tensors.

DMRG

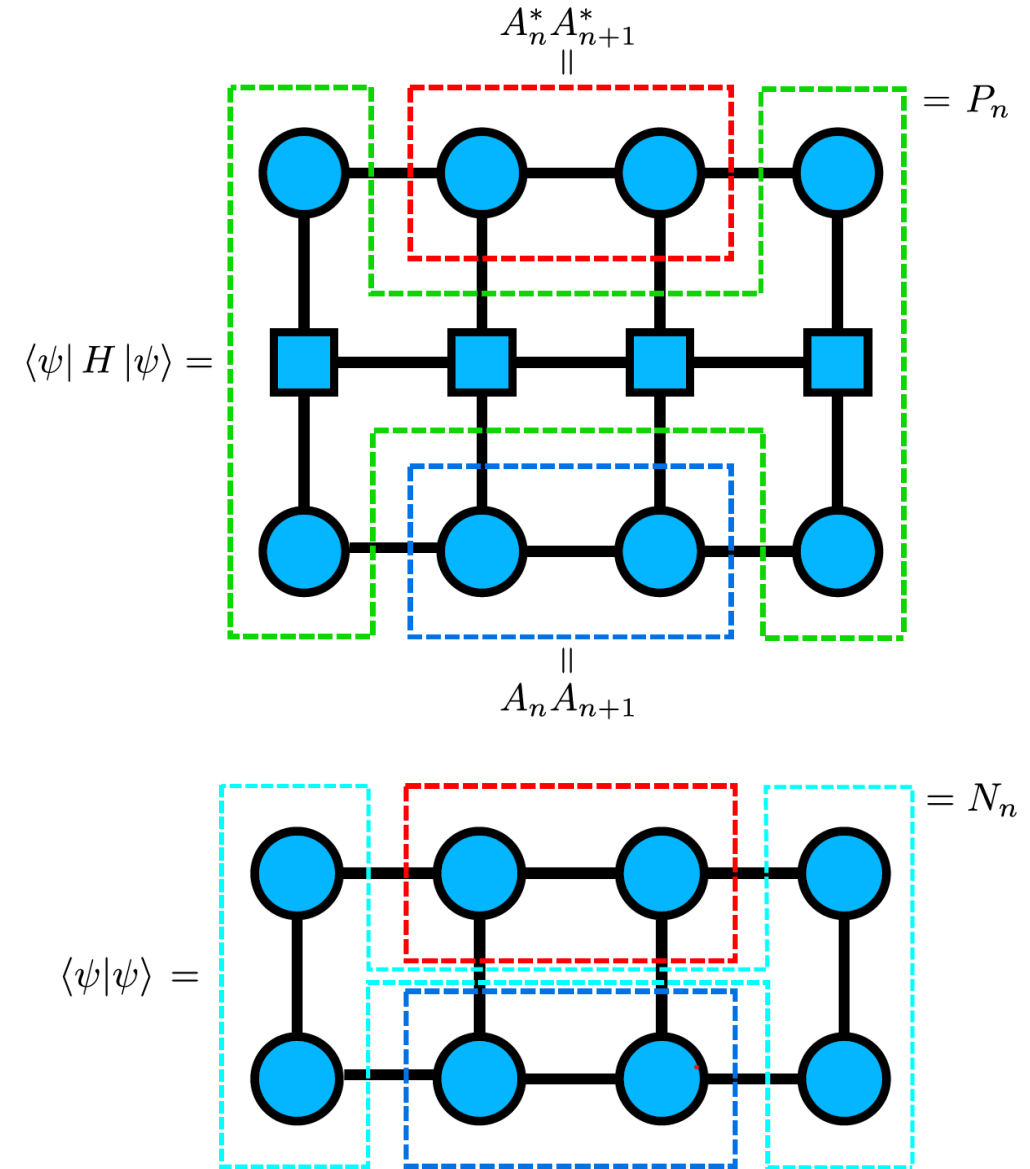
The Algorithm

- Used to find the ground state of Hamiltonians expressed as Matrix Product Operators.
- Minimise the energy, $E = \frac{\langle \psi | H | \psi \rangle}{\langle \psi | \psi \rangle}$ with respect to two sites at a time.

- Canonical forms simplify the calculation of N_n , at each site the equation to be solved is

$$P_n(A_n A_{n+1}) = E(A_n A_{n+1})$$

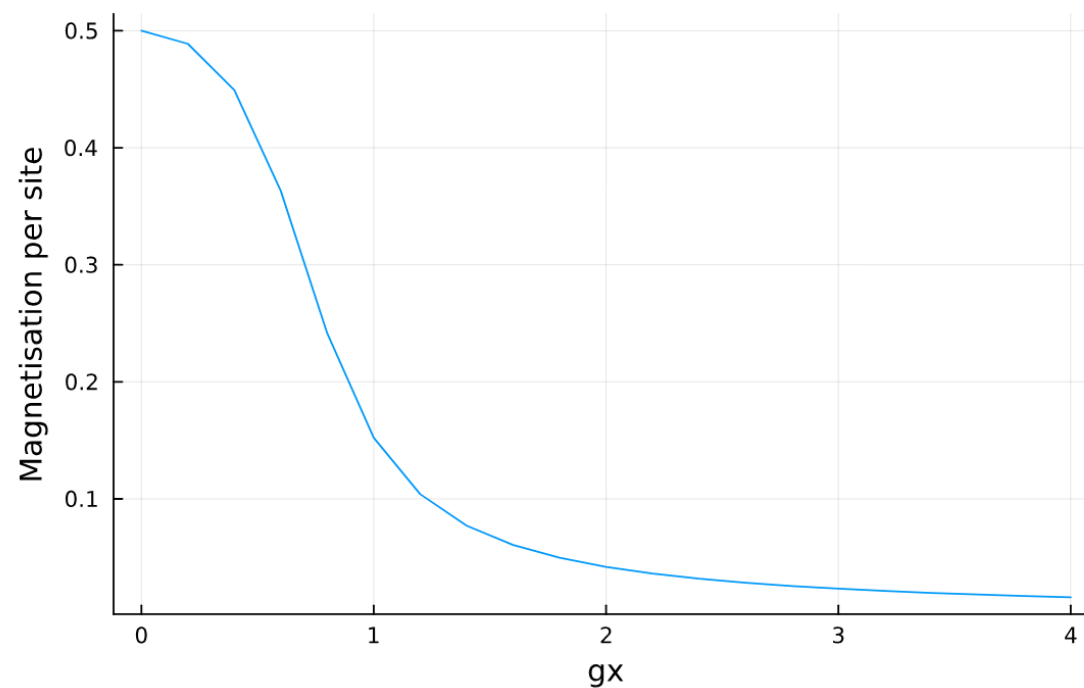
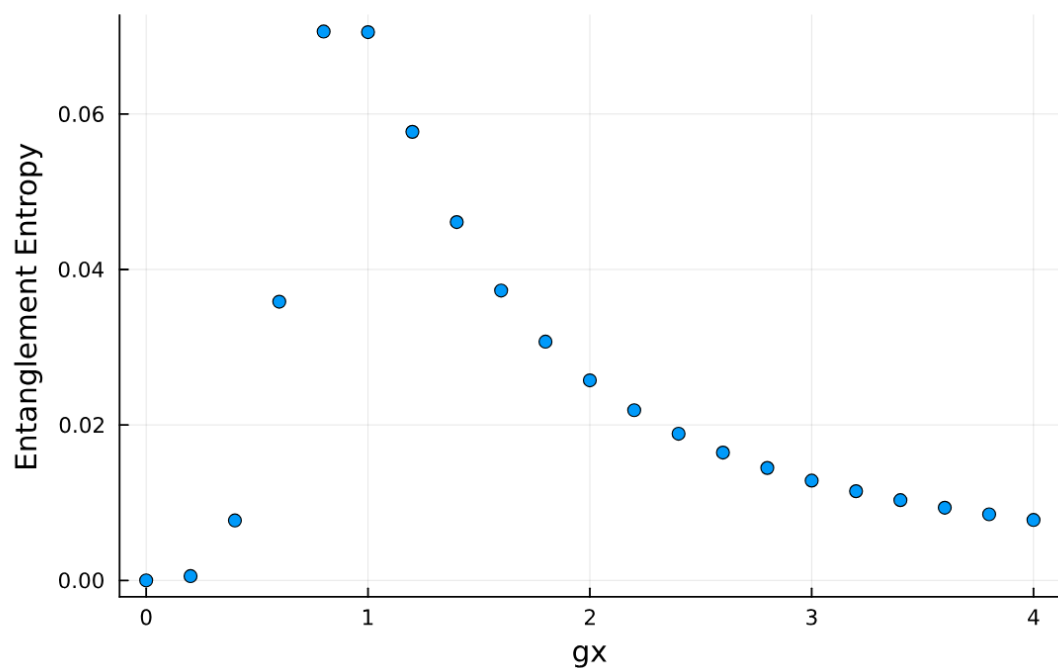
- The algorithm sweeps left and right updating sites until the percentage change in energy is below threshold for convergence
- Improve efficiency by storing and updating P_n at each step for reuse to avoid calculating contractions



DMRG

Transverse Field Ising Model

$$\hat{H} = J \sum_{i=1}^{N-1} Z_i Z_{i+1} + g_x \sum_{i=1}^N X_i + g_z \sum_{i=1}^N Z_i$$



DMRG

Schwinger Model

<https://doi.org/10.1038/s41534-024-00950-6>

$$\mathcal{H} = -i\bar{\psi}\gamma^1(\partial_1 - igA_1)\psi + m\bar{\psi}\psi + \frac{1}{2}\left(\dot{A}_1 + \frac{g\theta}{2\pi}\right)^2 \Rightarrow H_W = x(r-1)\sum_{n=0}^{N-2}(\sigma_{2n}^+Z_{2n+1}Z_{2n+2}\sigma_{2n+3}^- + \text{h.c.})$$

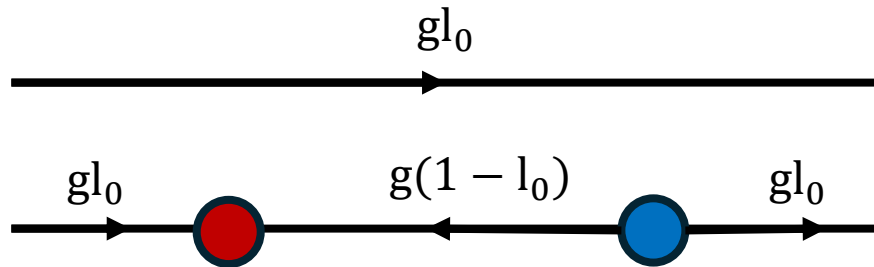
$$+ \frac{x(r+1)}{2}\sum_{n=0}^{N-2}(X_{2n+1}X_{2n+2} + Y_{2n+1}Y_{2n+2})$$

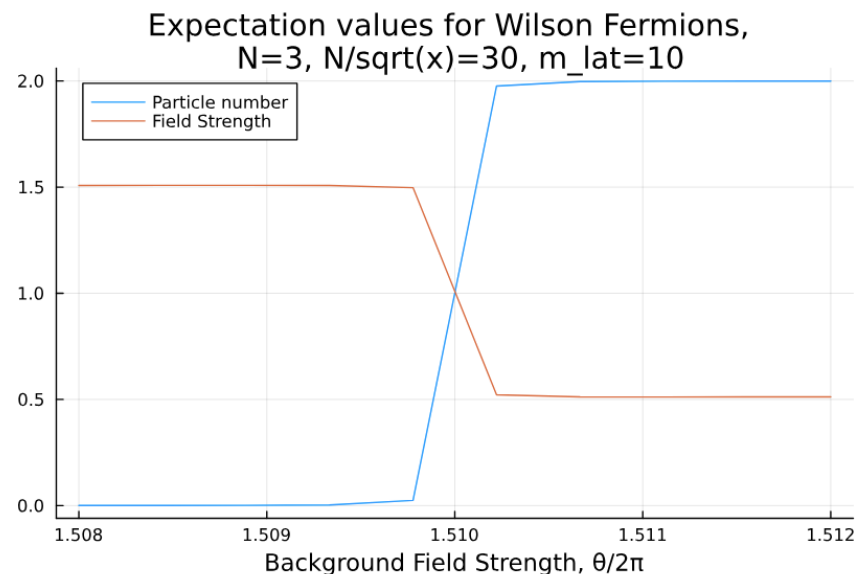
$$+ \left(\frac{m_{\text{lat}}}{g}\sqrt{x} + xr\right)\sum_{n=0}^{N-1}(X_{2n}X_{2n+1} + Y_{2n}Y_{2n+1})$$

$$+ \frac{1}{2}\sum_{n=0}^{2N-1}\sum_{k=n+1}^{2N-1}\left(N - \left\lceil \frac{k+1}{2} \right\rceil + \lambda\right)Z_nZ_k$$

$$+ l_0\sum_{n=0}^{2N-3}\left(N - \left\lceil \frac{n+1}{2} \right\rceil\right)Z_n$$

$$+ l_0^2(N-1) + \frac{1}{4}N(N-1) + \frac{\lambda N}{2}$$



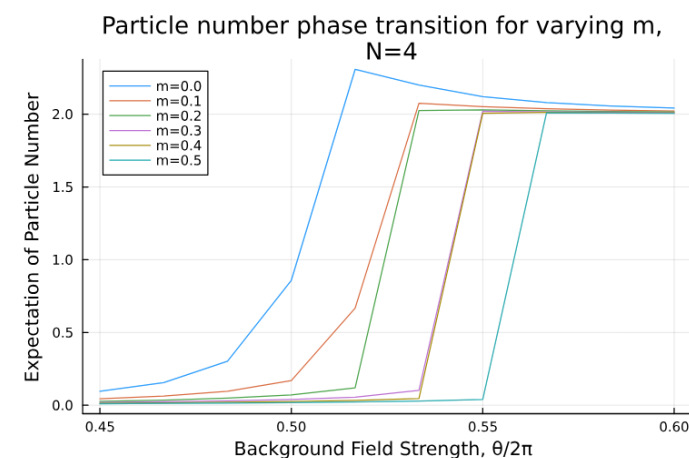
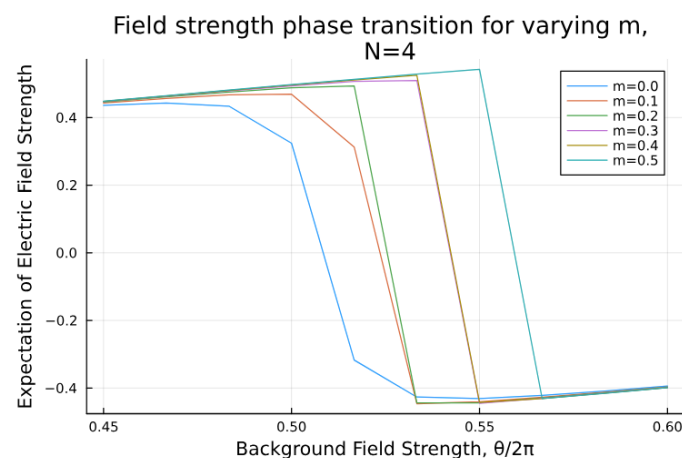


First Order Phase Transition:

- Mass > 0.33 , transition expected at $\theta/2\pi = l_0 = 0.5$ (lattice effects change the location of the phase transition)
- Pair production: particle number jumps from 0 to 2
- Particles screen the background field: field strength drops

Second Order Phase Transition:

- For masses below 0.33, no phase transition
- See particle number and field strength change smoothly



TEBD

The Algorithm

$$\begin{aligned}\hat{U}^{\text{exact}}(\delta) &= e^{-i\delta\hat{H}} \\ &= e^{-i\delta\hat{H}_{\text{even}}} e^{-i\delta\hat{H}_{\text{odd}}} e^{-i\delta^2[\hat{H}_{\text{even}},\hat{H}_{\text{odd}}]} \\ &\approx e^{-i\delta\hat{H}_{\text{even}}} e^{-i\delta\hat{H}_{\text{odd}}} \\ &\equiv \hat{U}^{\text{TEBD1}}(\delta).\end{aligned}$$

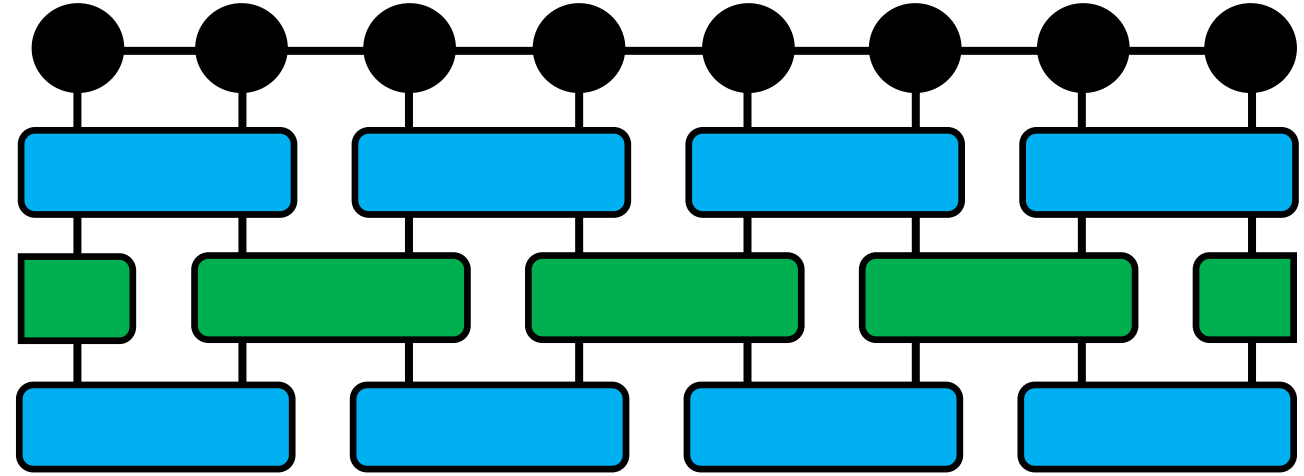
$$e^{-i\delta\hat{H}_{\text{even}}} = e^{-i\delta\sum_{j\text{ even}}\hat{h}_{j,j+1}} = \prod_{j\text{ even}} e^{-i\delta\hat{h}_{j,j+1}}$$

$$\hat{U}^{\text{TEBD2}}(\delta) \equiv e^{-i\frac{\delta}{2}\hat{H}_{\text{even}}} e^{-i\delta\hat{H}_{\text{odd}}} e^{-i\frac{\delta}{2}\hat{H}_{\text{even}}}$$

Split the Hamiltonian into internally commuting parts which have terms that can be exponentiated individually

TEBD2 has error $O(\delta^2)$ per time step T/δ

$$\hat{H} = \sum_j \hat{h}_{j,j+1}$$

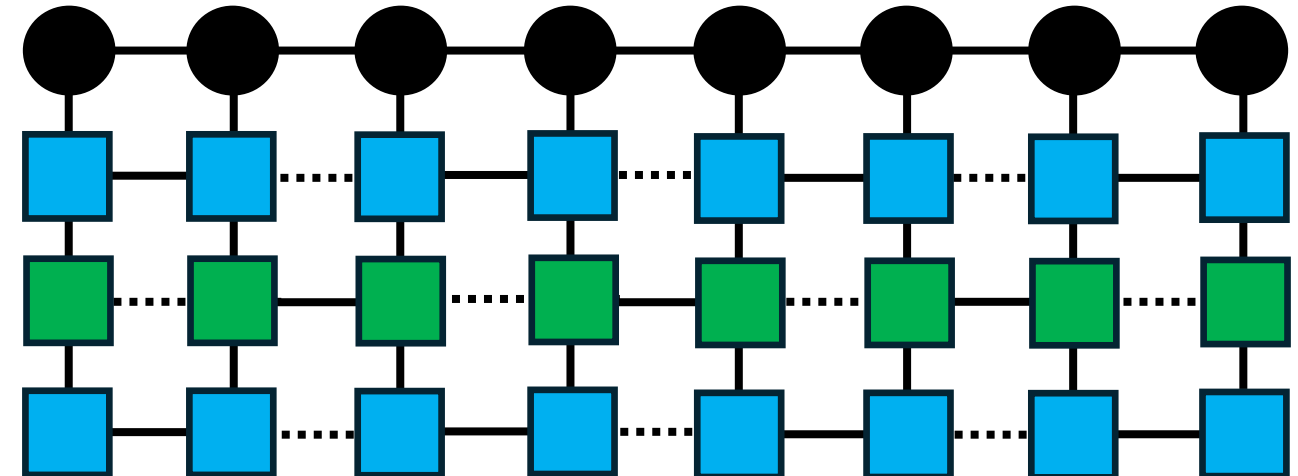


$$|\psi\rangle$$

$$\hat{U}_{\text{Odd}} = e^{-i\delta\hat{H}_{\text{Odd}}/2}$$

$$\hat{U}_{\text{Even}} = e^{-i\delta\hat{H}_{\text{Even}}}$$

$$\hat{U}_{\text{Odd}} = e^{-i\delta\hat{H}_{\text{Odd}}/2}$$



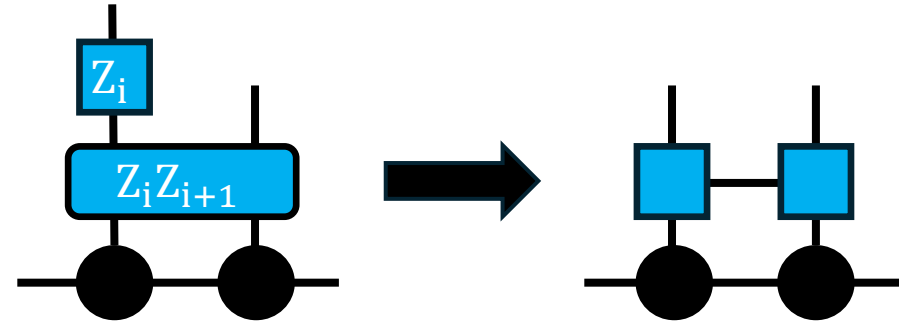
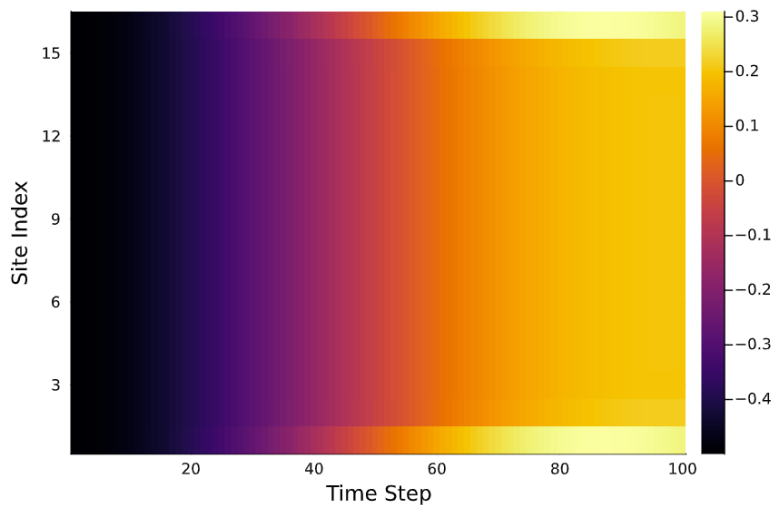
TEBD

Transverse Field Ising Model

Split Hamiltonian in two:
Z terms ($Z_i Z_{i+1}$ and Z_i), and X (transverse)
terms. Internally commute as same Paulis

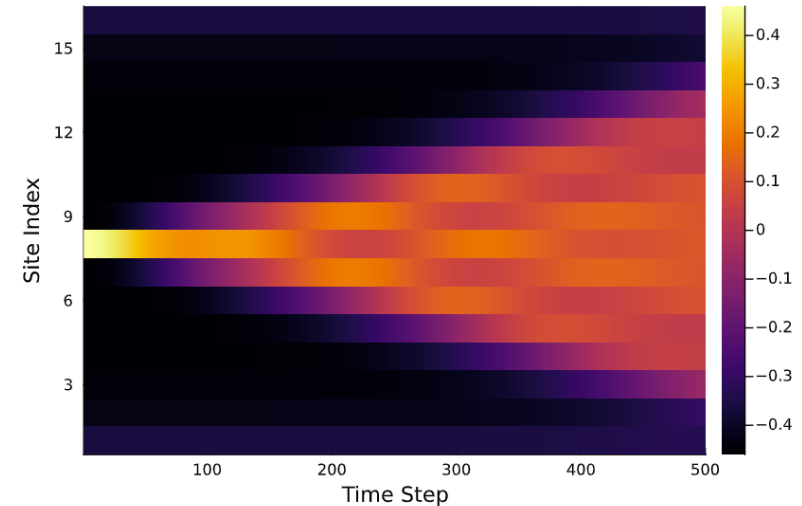
Quenches:

- Find the ground state with no external field
- Time evolve with an external field
- See how the spins change



Spinons:

- Find the ground state with an external field
- Flip one spin in ground state
- See how the entanglement grows through the system

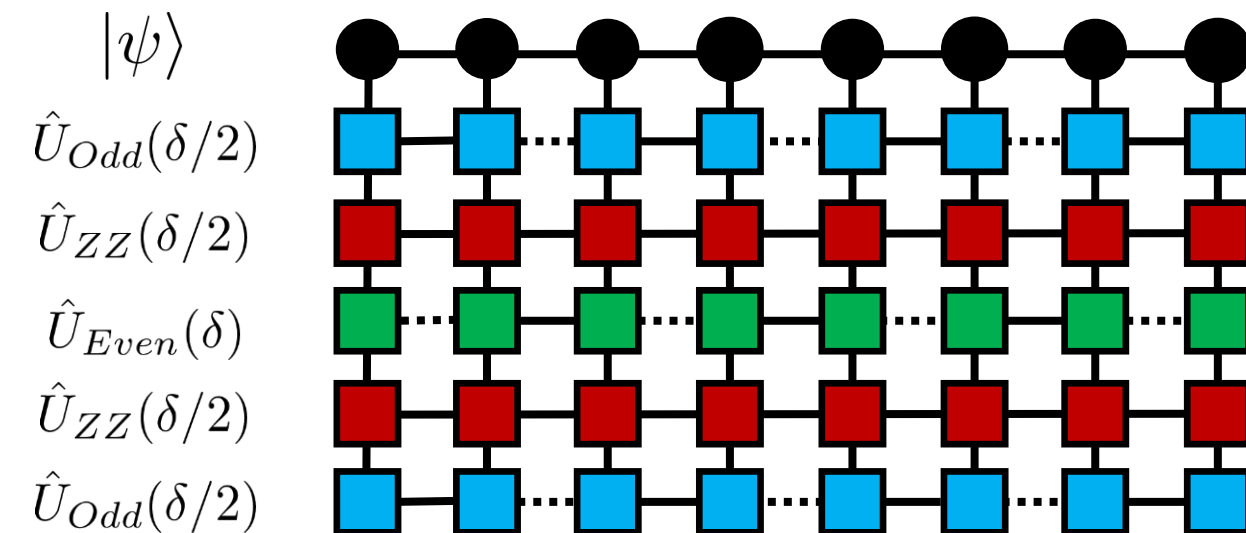


TEBD

Time Dependent Schwinger Model

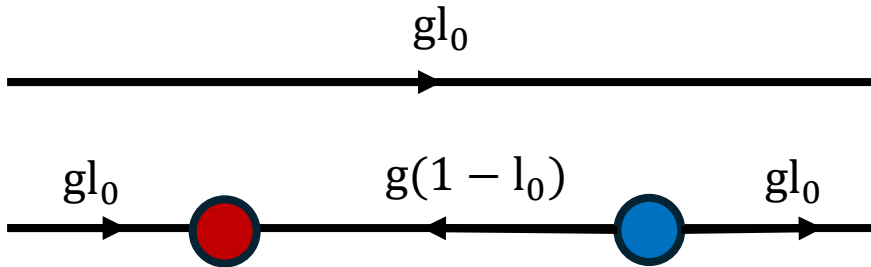
The Schwinger Hamiltonian has long range $Z_i Z_j$ interaction terms which are inefficient to express with gates – can do it by implementing swaps but it is costly – so we separate them from the rest and approximate the terms with an MPO of truncated bond dimensions

$$\begin{aligned}
 H_W = & x(r-1) \sum_{n=0}^{N-2} \left(\sigma_{2n}^+ Z_{2n+1} Z_{2n+2} \sigma_{2n+3}^- + \text{h.c.} \right) \\
 & + \frac{x(r+1)}{2} \sum_{n=0}^{N-2} (X_{2n+1} X_{2n+2} + Y_{2n+1} Y_{2n+2}) \\
 & + \left(\frac{m_{\text{lat}}}{g} \sqrt{x} + xr \right) \sum_{n=0}^{N-1} (X_{2n} X_{2n+1} + Y_{2n} Y_{2n+1}) \\
 & + \frac{1}{2} \sum_{n=0}^{2N-1} \sum_{k=n+1}^{2N-1} \left(N - \left\lfloor \frac{k+1}{2} \right\rfloor + \lambda \right) Z_n Z_k \\
 & + l_0 \sum_{n=0}^{2N-3} \left(N - \left\lfloor \frac{n+1}{2} \right\rfloor \right) Z_n \\
 & + l_0^2 (N-1) + \frac{1}{4} N(N-1) + \frac{\lambda N}{2}
 \end{aligned}$$

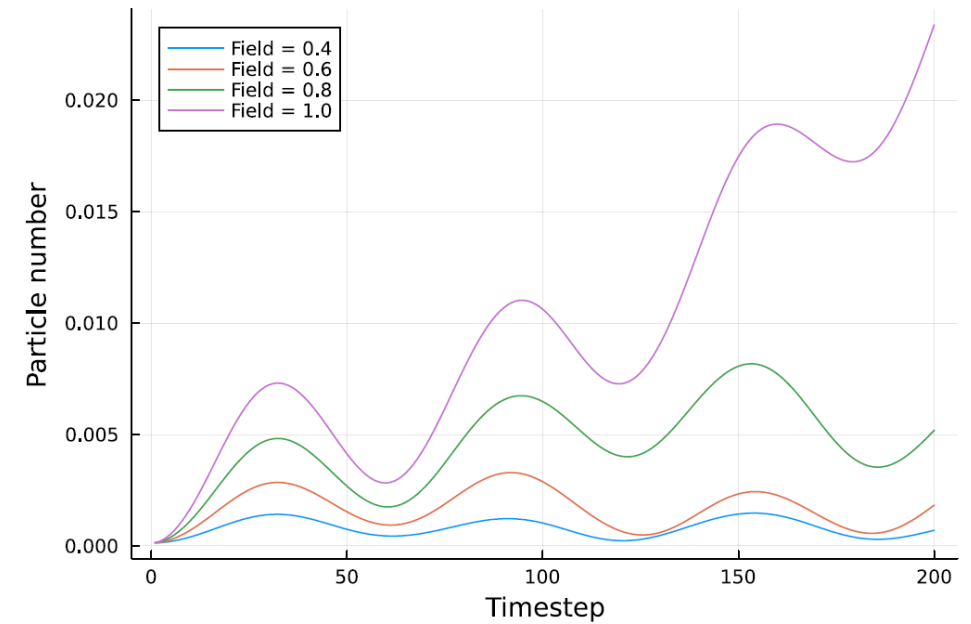


TEBD

Time Dependent Schwinger Model



When we quench the Hamiltonian from no external field to different values, we see oscillations in particle number



Other Applications of Matrix Product States

Thanks for Listening!

- Integrating Feynman diagrams for high energy physics
- Machine learning
- Simulating quantum Fourier transform
- Simulating and benchmarking quantum computing
- Solving PDEs

