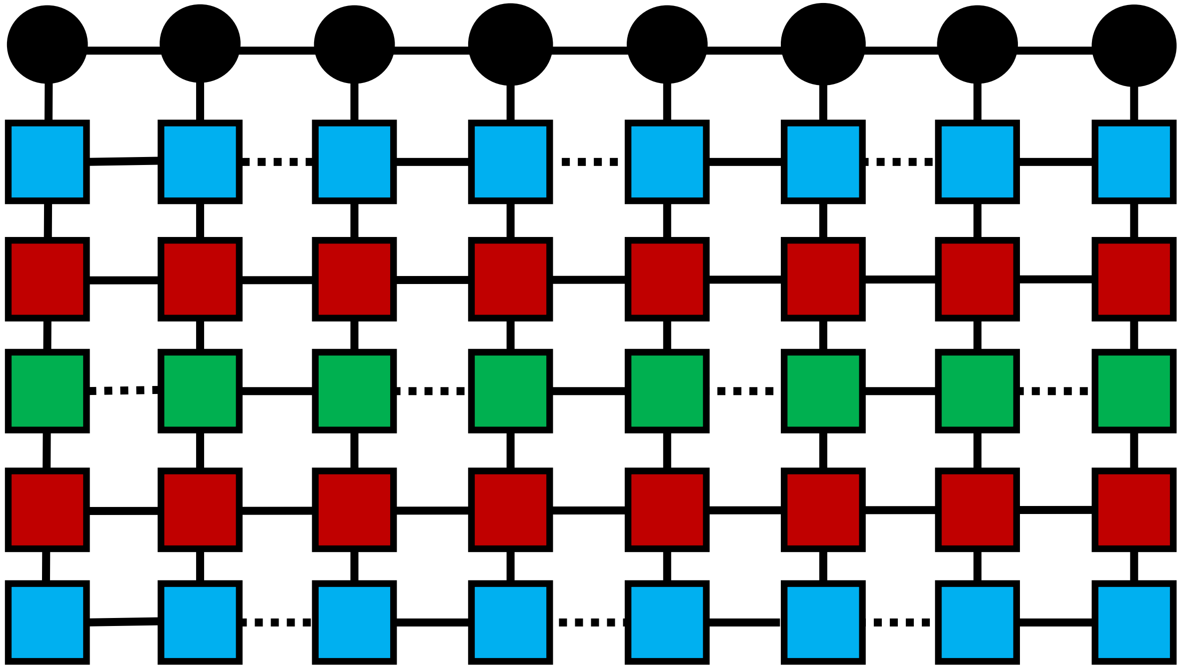
An Introduction to Matrix Product State Algorithms

Nicholas Woodford, Supervisor: Takis Anglides



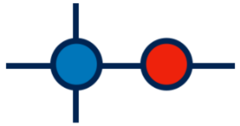
DESY. QUANTUM
Center for Quantum Technology and Applications

Matrix Product States

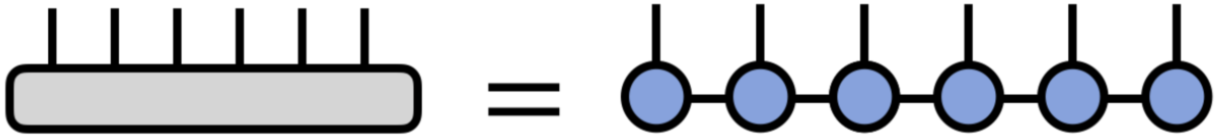
Tensor Networks

Matrix M_{ij}

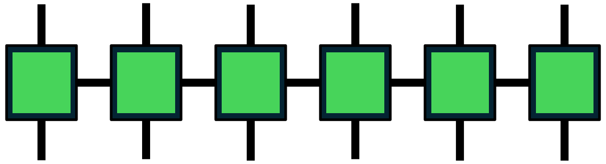
3-legged Tensor T_{ijk}



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



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



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’

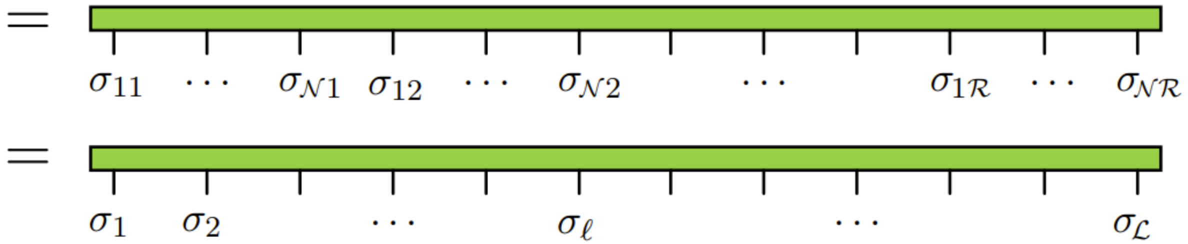
https://arxiv.org/pdf/2407.02454
https://arxiv.org/pdf/2207.06135

Matrix Product States

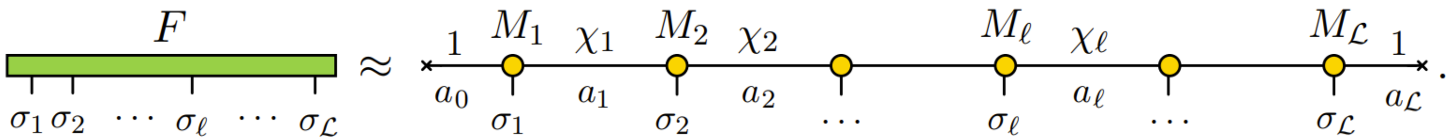
Tensor Cross Interpolation

- 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

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



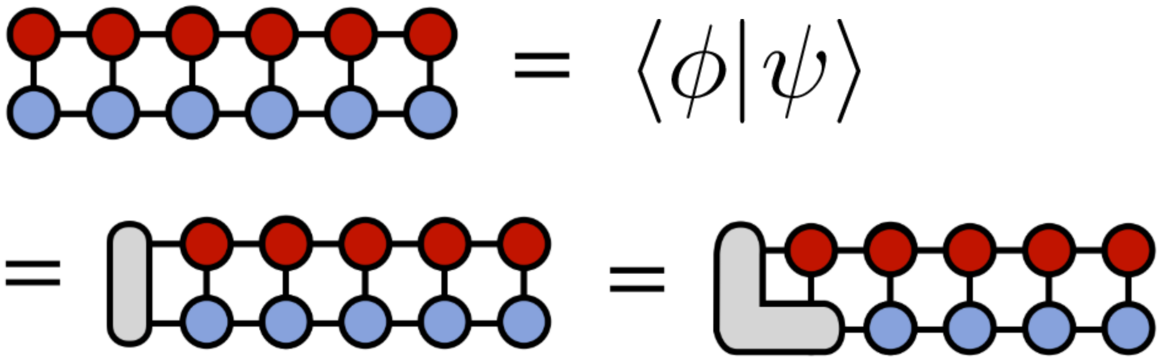
$$F_{\boldsymbol{\sigma}} \approx \widetilde{F}_{\boldsymbol{\sigma}} = \prod_{\ell=1}^{\mathcal{L}} M_\ell^{\sigma_\ell} = [M_1]^{\sigma_1}_{1a_1}[M_2]^{\sigma_2}_{a_1a_2}\cdots[M_{\mathcal{L}}]^{\sigma_{\mathcal{L}}}_{a_{\mathcal{L}-1}1},$$



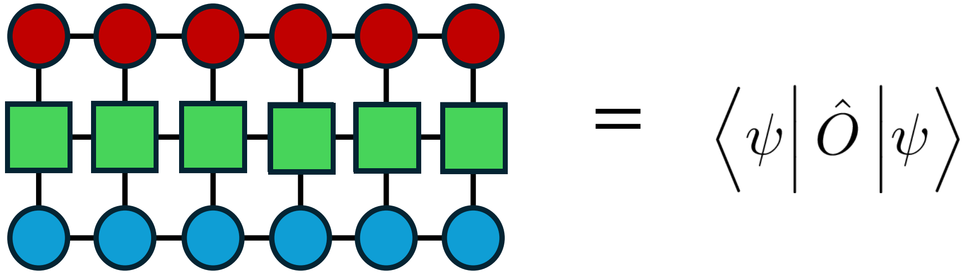
Matrix Product States

Basic Operations

Tensor Contraction:

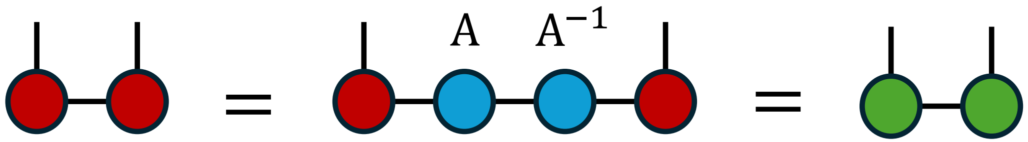


Expectation values:

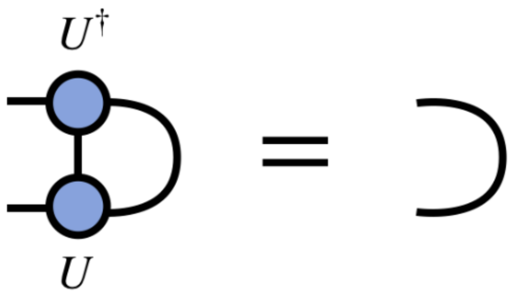


Efficient contraction algorithms have time complexity $O(dD^3N)$

Canonical Forms:



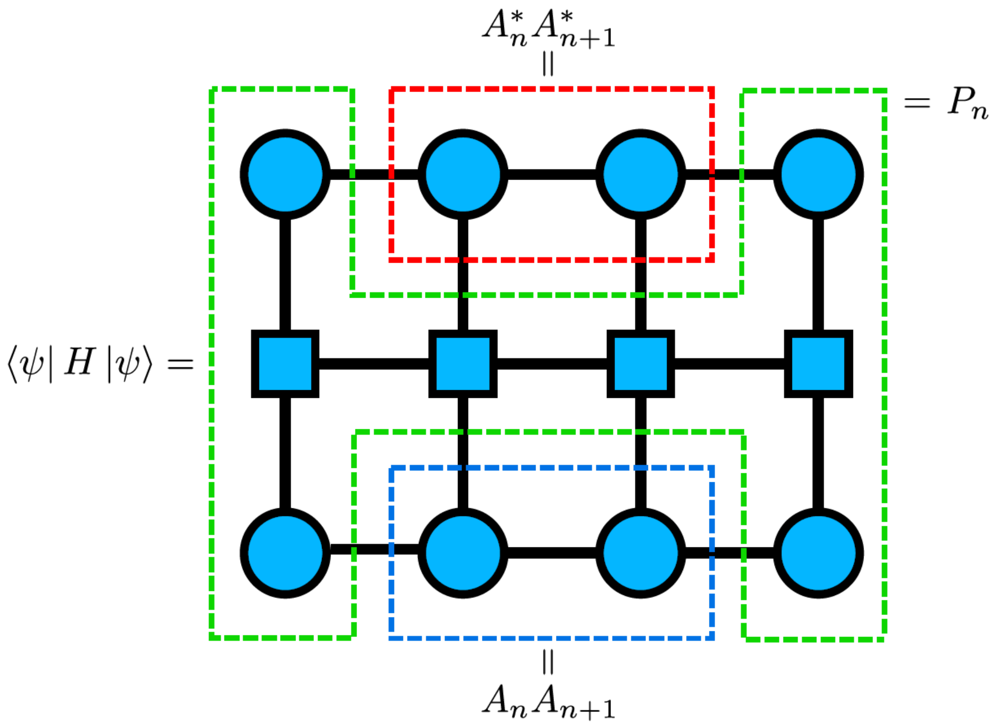
,

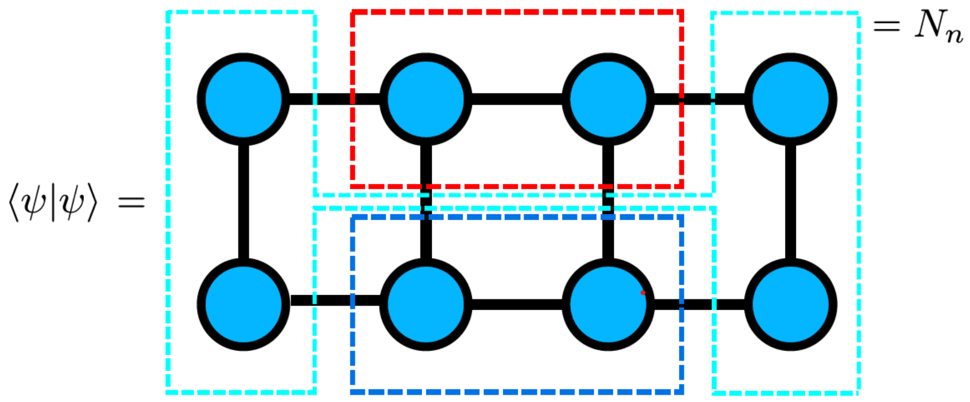


DMRG

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
$$P_n(A_nA_{n+1}) = E(A_nA_{n+1})$$
- 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

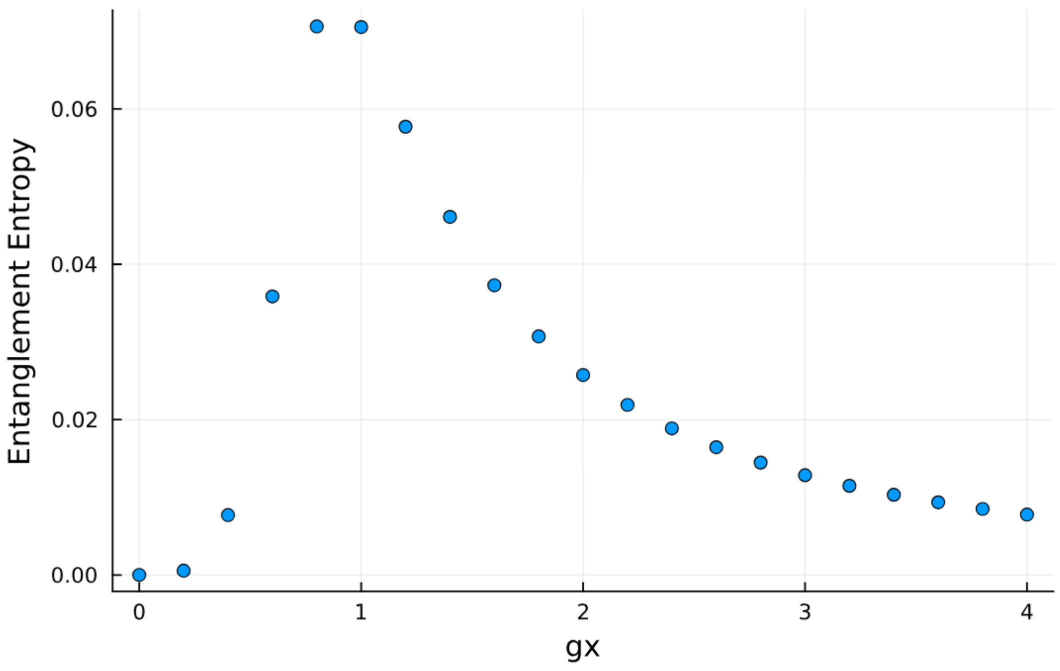


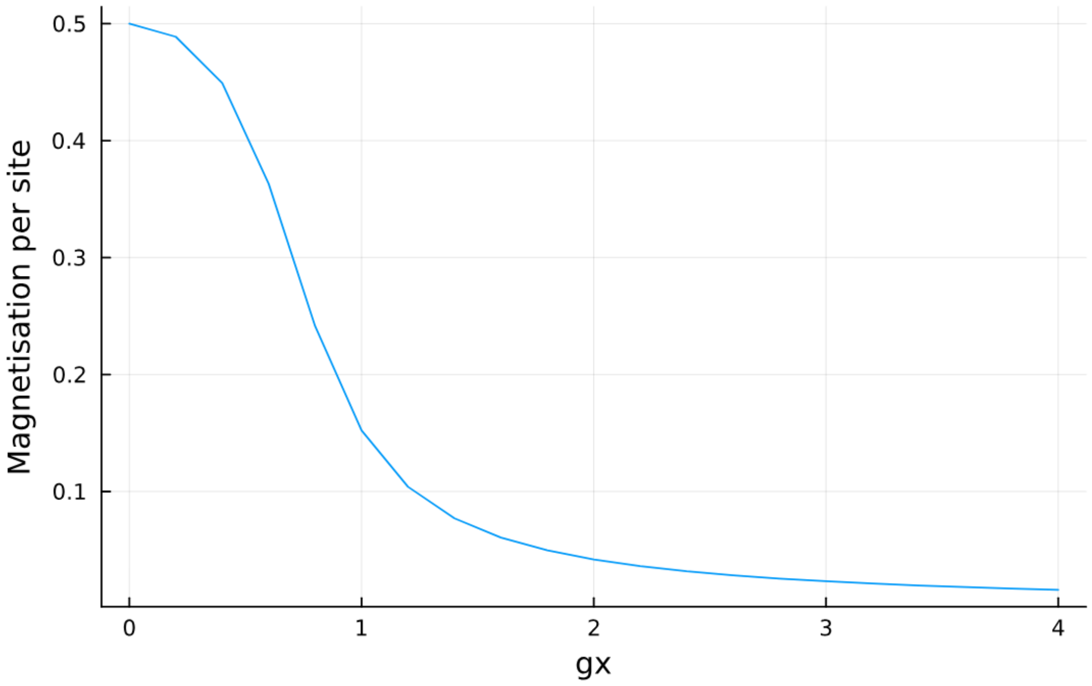


DMRG

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

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

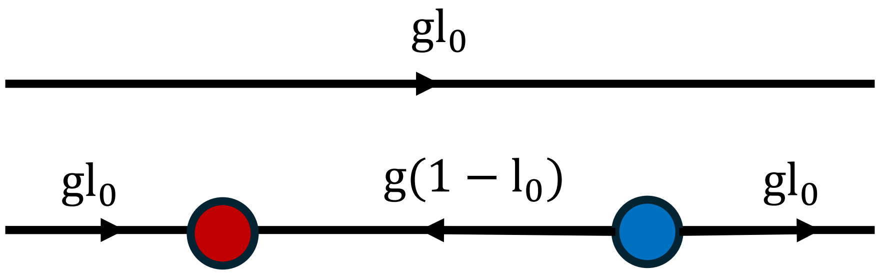
$$+ \frac{x(r+1)}{2}\sum_{n=0}^{N-2}\left(X_{2n+1}X_{2n+2} + Y_{2n+1}Y_{2n+2}\right)$$

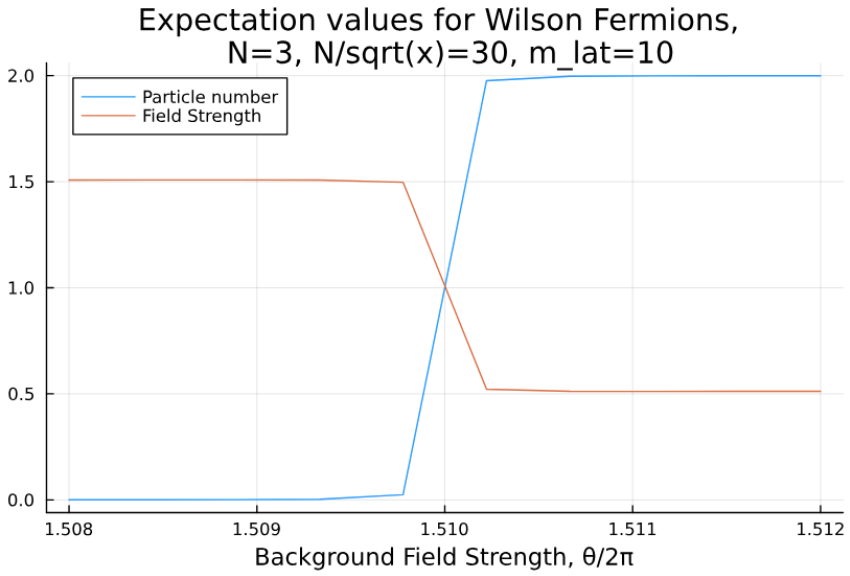
$$+ \left(\frac{m_{\text{lat}}}{g}\sqrt{x} + xr\right)\sum_{n=0}^{N-1}\left(X_{2n}X_{2n+1} + Y_{2n}Y_{2n+1}\right)$$

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

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

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



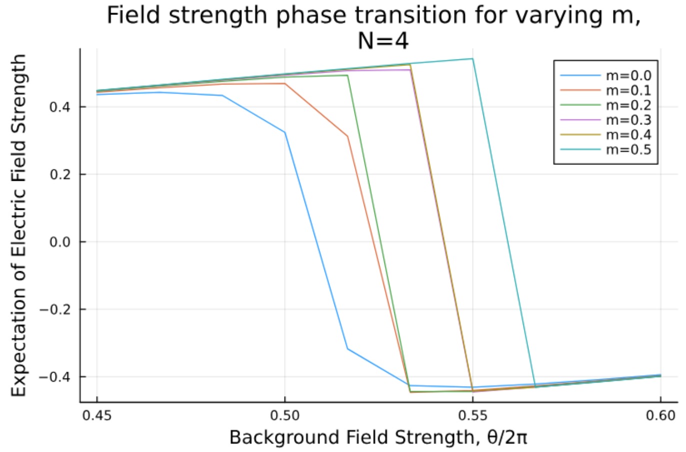


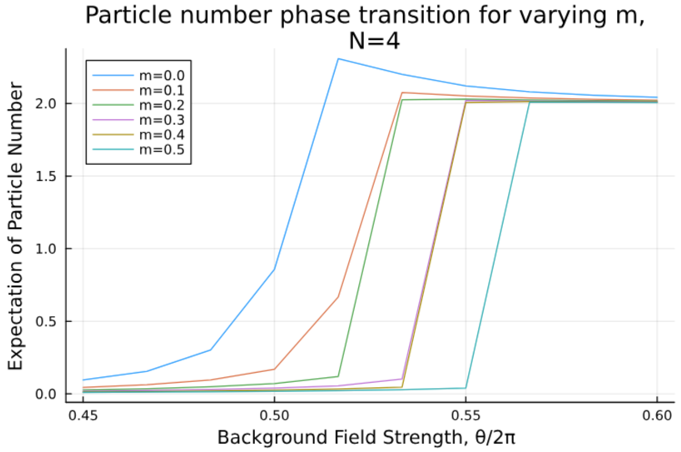
First Order Phase Transition:

- 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





https://iopscience.iop.org/article/10.1088/1742-5468/2004/04/P04005/pdf

TEBD

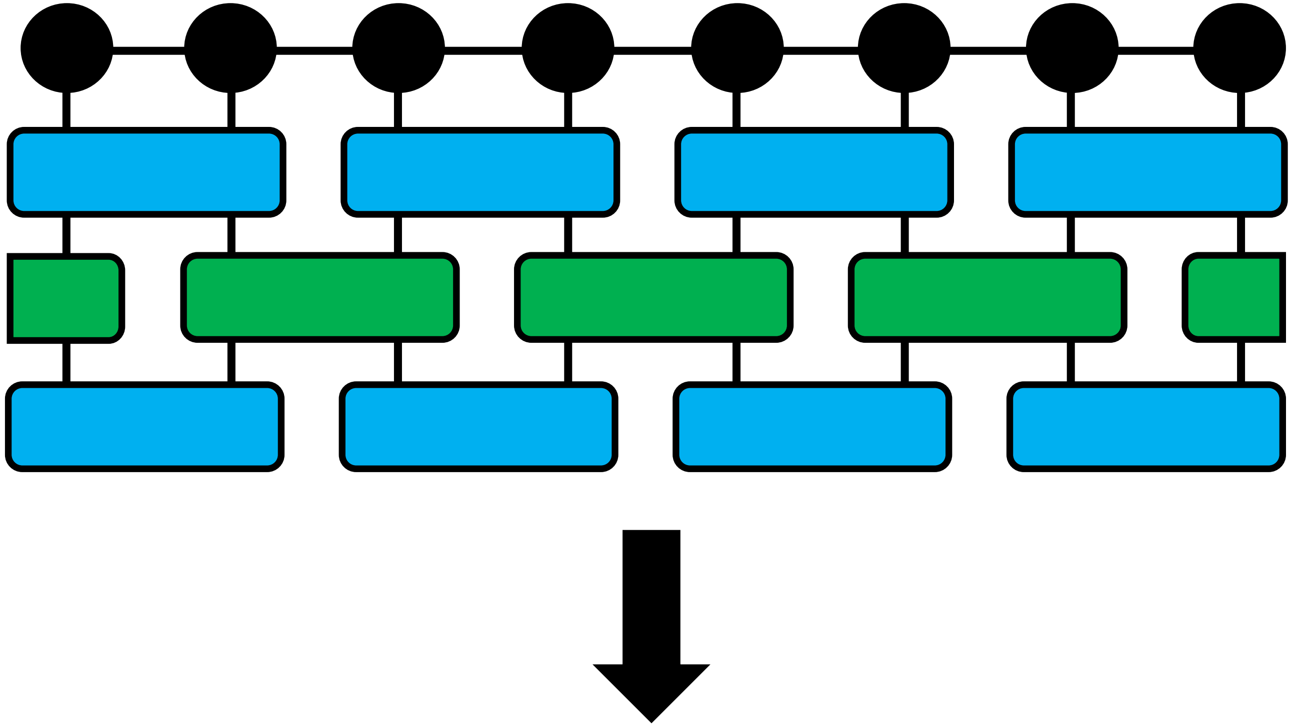
The Algorithm

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

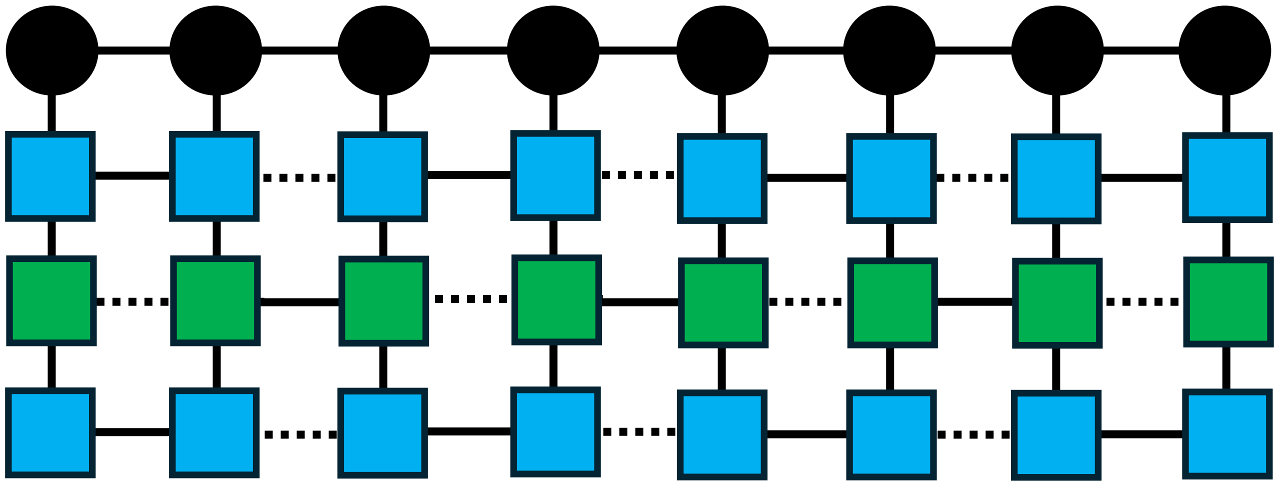
$$\hat{H} = \sum_j \hat{h}_{j,j+1}$$

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

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



$|\psi\rangle$



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

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

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

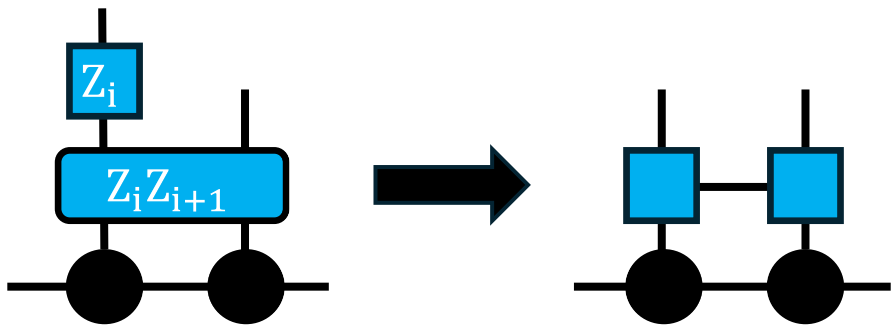
Split the Hamiltonian into internally commuting parts which have terms that can be exponentiated individually

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

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

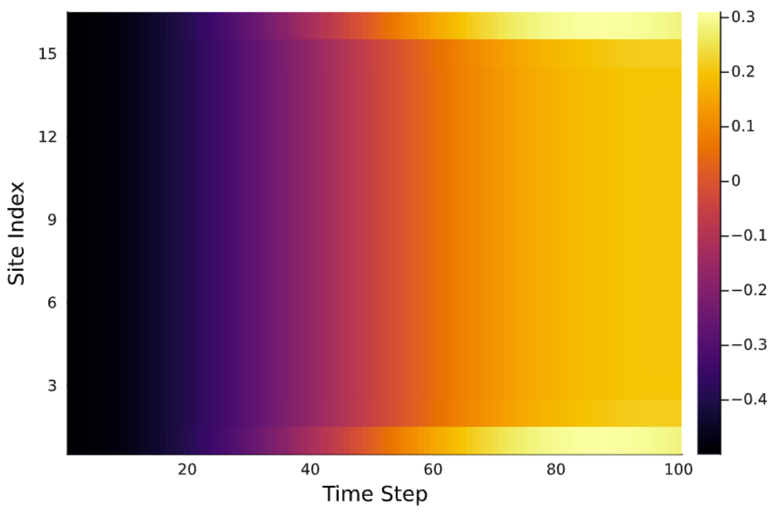


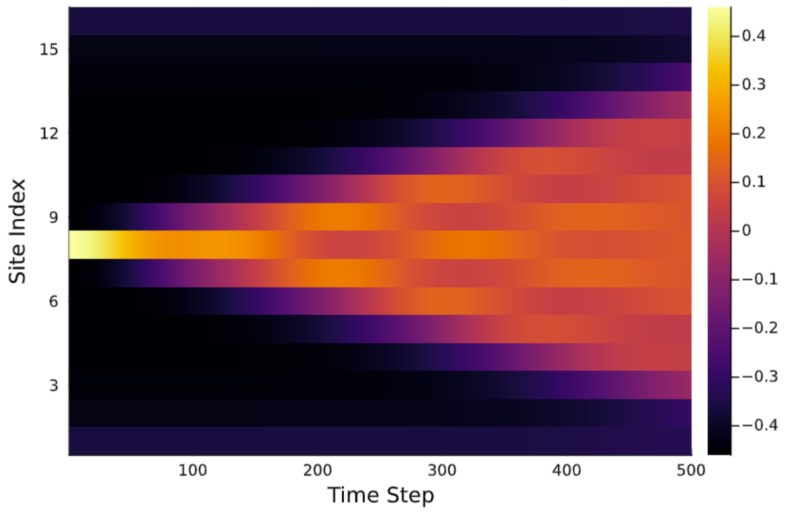
Quenches:

- 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





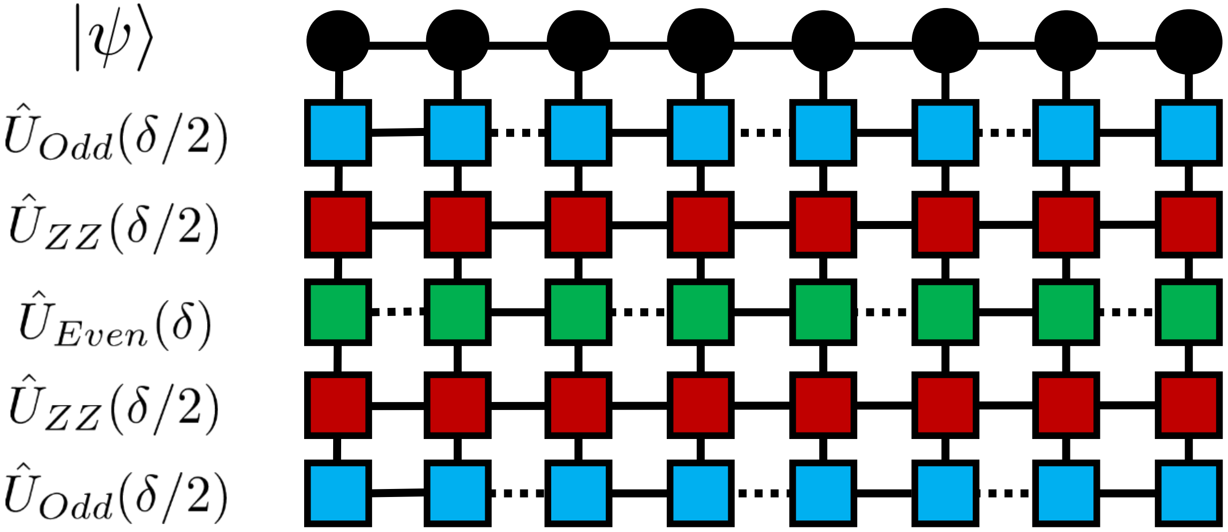
arxiv.org/pdf/1909.04821

TEBD

Time Dependent Schwinger Model

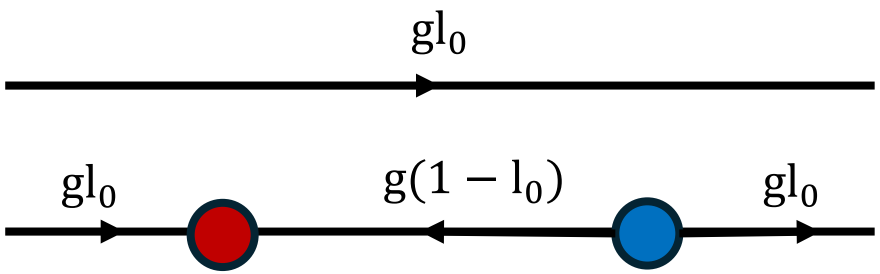
The Schwinger Hamiltonian has long range $Z_i Z_j$ interaction terms which are inefficient to express with gates – can do it by implementing swaps but it is costly – so we separate them from the rest and approximate the terms with an MPO of truncated bond dimensions

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

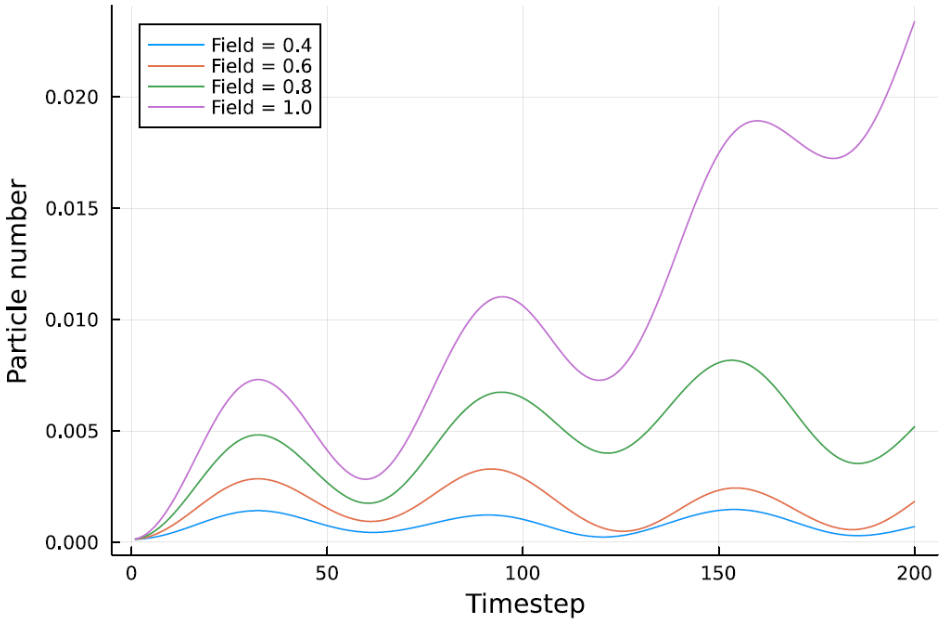


TEBD

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