Matrix Product States, Operators, and the Thermodynamic Limit of DMRG

Ian McCulloch

University of Queensland
Entanglement, Many-Body Systems & Simulations Group

28/1/2009

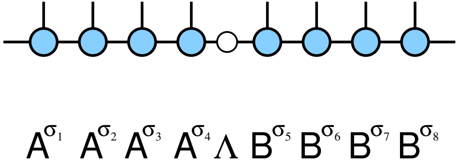
Outline

1. MPS Introduction
2. Infinite size DMRG
3. Scaling relations in the thermodynamic limit
4. Expectation values of iMPO's
5. Conclusions

MPS representations

DMRG produces a matrix product state wavefunction.

$$|\Psi\rangle = \mathrm{Tr} \sum_{s_1, s_2, \ldots} A^{s_1} A^{s_2} A^{s_3} A^{s_4} \cdots |s_1\rangle |s_2\rangle |s_3\rangle |s_4\rangle \cdots$$

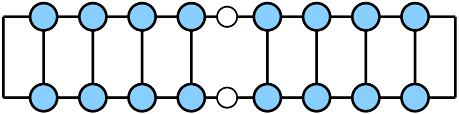


Λ is the wavefunction in the *Schmidt basis*
$|\Psi\rangle = \sum_i \Lambda_{ii} |i\rangle_L |i\rangle_R$

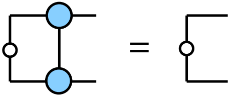
Key advantage of the MPS representation:
Manipulations of the wavefunction are easy - both conceptually and numerically

MPS operations - normalization

Simple example: how do we calculate the norm?

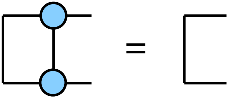


Sequential contractions:

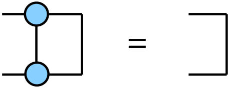


$$\tilde{E}_n = \sum_s A^{s\dagger} \, \tilde{E}_{n-1} \, A^s$$

This process is greatly simplified if we require that the MPS is *orthonormalized*

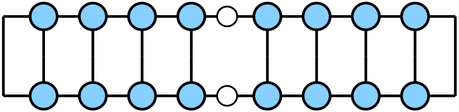


$$\sum_s A^{s\dagger} A^s = I$$



$$\sum_s B^s B^{s\dagger} = I$$

Sequential contraction for the norm



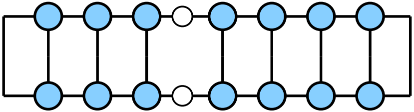
Remaining tensor:

$$|\Psi\rangle = \sum_i \Lambda_{ii} |i\rangle_L |i\rangle_R$$

$$\mathrm{Tr}\, \Lambda^\dagger \Lambda = \langle\Psi|\Psi\rangle = 1$$

This works because we know how to contract tensors at the boundary

Sequential contraction for the norm



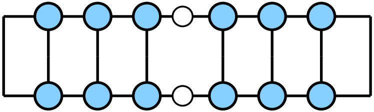
Remaining tensor:

$$|\Psi\rangle = \sum_i \Lambda_{ii} |i\rangle_L |i\rangle_R$$

$$\mathrm{Tr}\,\Lambda^\dagger \Lambda = \langle\Psi|\Psi\rangle = 1$$

This works because we know how to contract tensors at the boundary

Sequential contraction for the norm



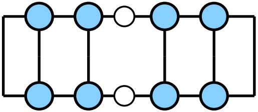
Remaining tensor:

$$|\Psi\rangle = \sum_i \Lambda_{ii} |i\rangle_L |i\rangle_R$$

$$\mathrm{Tr}\,\Lambda^\dagger \Lambda = \langle \Psi | \Psi \rangle = 1$$

This works because we know how to contract tensors at the boundary

Sequential contraction for the norm



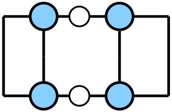
Remaining tensor:

$$|\Psi\rangle = \sum_i \Lambda_{ii} |i\rangle_L |i\rangle_R$$

$$\mathrm{Tr}\, \Lambda^\dagger \Lambda = \langle \Psi | \Psi \rangle = 1$$

This works because we know how to contract tensors at the boundary

Sequential contraction for the norm



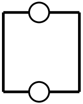
Remaining tensor:

$$|\Psi\rangle = \sum_i \Lambda_{ii} |i\rangle_L |i\rangle_R$$

$$\mathrm{Tr}\, \Lambda^\dagger \Lambda = \langle \Psi | \Psi \rangle = 1$$

This works because we know how to contract tensors at the boundary

Sequential contraction for the norm



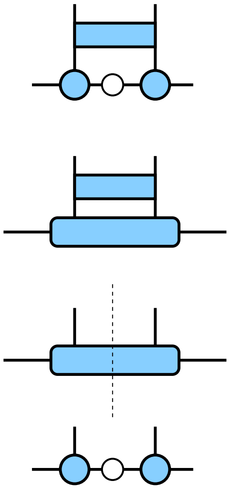
Remaining tensor:

$$|\Psi\rangle = \sum_i \Lambda_{ii} |i\rangle_L |i\rangle_R$$

$$\mathrm{Tr}\, \Lambda^\dagger \Lambda = \langle \Psi | \Psi \rangle = 1$$

This works because we know how to contract tensors at the boundary

Bond Evolution



Apply operator $U_{s_1 s_2}^{s_1' s_2'}$ to two sites:

$A'^{s_1'} \; \Lambda' \; B'^{s_2'} = U_{s_1 s_2}^{s_1' s_2'} \, A^{s_1} \; \Lambda \; B^{s_2}$

Merge tensors:

$X^{s_1 s_2} = A^{s_1} \; \Lambda \; B^{s_2}$

Apply evolution operator:

$X' s_1' s_2' = U_{s_1 s_2}^{s_1' s_2'} \, X^{s_1 s_2}$

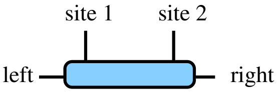
Singular value decomposition:

$X' s_1' s_2' = A'^{s_1'} \; \Lambda' \; B'^{s_2'}$

This is the basis of TEBD

DMRG: Local minimization of the Hamiltonian with an eigensolver

- We need a representation of the Hamiltonian operator in the Hilbert space of the tensors at the center of the lattice

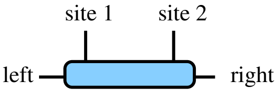


Partition the Hamiltonian operator:

$$
\begin{array}{rcl cccc}
 & & & \text{left} & \text{site 1} & \text{site 2} & \text{right} \\
H & = & & \tilde{H} & I & I & \tilde{I} \\
 & & + & \tilde{I} & H & I & \tilde{I} \\
 & & + & \tilde{I} & I & H & \tilde{I} \\
 & & + & \tilde{I} & I & I & \tilde{H} \\
 & & + & \tilde{b}^\dagger & b & I & \tilde{I} \\
 & & + & \tilde{I} & b^\dagger & b & \tilde{I} \\
 & & + & \tilde{I} & I & b^\dagger & \tilde{b} \\
 & & + & \text{h.c.} & & & \\
 & & + & \cdots & & &
\end{array}
$$

DMRG: Local minimization of the Hamiltonian with an eigensolver

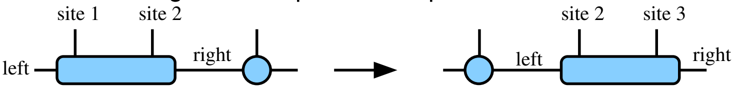
- We need a representation of the Hamiltonian operator in the Hilbert space of the tensors at the center of the lattice



Partition the Hamiltonian operator:

$$
\begin{array}{rcccccc}
 & & & \text{left} & \text{site 1} & \text{site 2} & \text{right} \\
H & = & & \tilde{H} & I & I & \tilde{I} \\
 & & + & \tilde{I} & H & I & \tilde{I} \\
 & & + & \tilde{I} & I & H & \tilde{I} \\
 & & + & \tilde{I} & I & I & \tilde{H} \\
 & & + & \tilde{b}^\dagger & b & I & \tilde{I} \\
 & & + & \tilde{I} & b^\dagger & b & \tilde{I} \\
 & & + & \tilde{I} & I & b^\dagger & \tilde{b} \\
 & & + & \text{h.c.} & & & \\
 & & + & \cdots & & &
\end{array}
$$

The second ingredient: update the operators in the course of a DMRG sweep

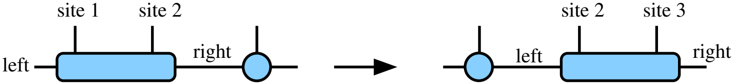


New block operator $\tilde{Y} \otimes Z \to \tilde{X}$

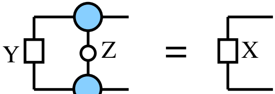
$$\sum_{s' s} \langle s' | Z | s \rangle A^{s'\dagger} \tilde{Y} A^{s} = \tilde{X}$$

But our new block operators may be sums of these terms.
Example: $\tilde{H} \leftarrow \tilde{H} \otimes I + \tilde{I} \otimes H + \tilde{b}^{\dagger} \otimes b \ldots$

The second ingredient: update the operators in the course of a DMRG sweep



New block operator $\tilde{Y} \otimes Z \rightarrow \tilde{X}$

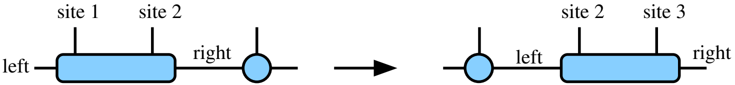


$$\sum_{s' s} \langle\, s' \,|\, Z \,|\, s \,\rangle\, A^{s'\dagger}\, \tilde{Y}\, A^{s} = \tilde{X}$$

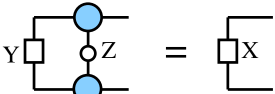
But our new block operators may be sums of these terms.

Example: $\tilde{H} \leftarrow \tilde{H} \otimes I + \tilde{I} \otimes H + \tilde{b}^{\dagger} \otimes b \ldots$

The second ingredient: update the operators in the course of a DMRG sweep



New block operator $\tilde{Y} \otimes Z \to \tilde{X}$



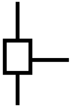
$$\sum_{s' s} \langle s' \,|\, Z \,|\, s \rangle \, A^{s' \dagger} \, \tilde{Y} \, A^{s} = \tilde{X}$$

But our new block operators may be sums of these terms.
Example: $\tilde{H} \leftarrow \tilde{H} \otimes I + \tilde{I} \otimes H + \tilde{b}^{\dagger} \otimes b \ldots$

Constructing the block operators

How to we represent the block update in a generic way?

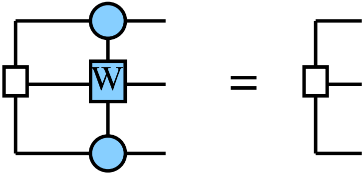
Denote the block operators by a set, labeled by index α



α

$$\tilde{E}^{\alpha},\ \alpha = 1, 2, \ldots, M$$

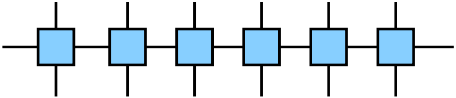
Generalize the local operator into a 4-index *matrix product operator* $W^{s's}_{\alpha'\alpha}$



$$\tilde{E}'^{\alpha'} = \sum_{s' s \alpha} W^{s's}_{\alpha'\alpha} A^{s'\dagger} \tilde{E}^{\alpha} A^{s}$$

Dimension M of the MPO is equal to the number of block operators - typically small

$$X = \mathrm{Tr} \sum_{s_1, s_2, \ldots} W^{s_1' s_1} A^{s_2' s_2} A^{s_3' s_3} A^{s_4' s_4} \cdots |s_1'\rangle\langle s_1| \, |s_2'\rangle\langle s_2| \, |s_3'\rangle\langle s_3| \, |s_4'\rangle\langle s_4| \cdots$$



The principal advantage of the MPO representation is that it allows arithmetic operations on the operators

$$X = Y + Z \to W_X = W_Y \oplus W_Z$$

Dimension increases: $M_X = M_Y + M_Z$

$$X = YZ \to W_X = W_Y \otimes W_Z$$

Dimension increases: $M_X = M_Y M_Z$

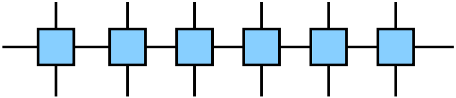
For finite, open systems, the boundary components are 1-dimensional.

'Orthogonalization' of the MPO keeps the dimension under control.

Operator arithmetic

IPM J. Stat. Mech. P10014 (2007)

$$X = \mathrm{Tr} \sum_{s_1,s_2,\ldots} W^{s_1' s_1} A^{s_2' s_2} A^{s_3' s_3} A^{s_4' s_4} \cdots |s_1'\rangle\langle s_1| \, |s_2'\rangle\langle s_2| \, |s_3'\rangle\langle s_3| \, |s_4'\rangle\langle s_4| \cdots$$



The principal advantage of the MPO representation is that it allows arithmetic operations on the operators

$$X = Y + Z \to W_X = W_Y \oplus W_Z$$

Dimension increases: $M_X = M_Y + M_Z$

$$X = YZ \to W_X = W_Y \otimes W_Z$$

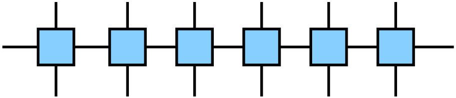
Dimension increases: $M_X = M_Y M_Z$

For finite, open systems, the boundary components are 1-dimensional.

'Orthogonalization' of the MPO keeps the dimension under control.

Operator arithmetic   IPM J. Stat. Mech. P10014 (2007)

$$X = \mathrm{Tr} \sum_{s_1, s_2, \ldots} W^{s_1' s_1} A^{s_2' s_2} A^{s_3' s_3} A^{s_4' s_4} \cdots |s_1'\rangle\langle s_1| \, |s_2'\rangle\langle s_2| \, |s_3'\rangle\langle s_3| \, |s_4'\rangle\langle s_4| \cdots$$



The principal advantage of the MPO representation is that it allows arithmetic operations on the operators

$$X = Y + Z \to W_X = W_Y \oplus W_Z$$

Dimension increases: $M_X = M_Y + M_Z$

$$X = YZ \to W_X = W_Y \otimes W_Z$$

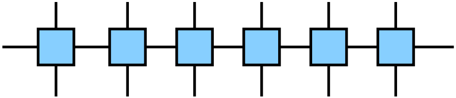
Dimension increases: $M_X = M_Y M_Z$

For finite, open systems, the boundary components are 1-dimensional.

'Orthogonalization' of the MPO keeps the dimension under control.

Operator arithmetic IPM J. Stat. Mech. P10014 (2007)

$$X = \mathrm{Tr} \sum_{s_1, s_2, \ldots} W^{s_1' s_1} A^{s_2' s_2} A^{s_3' s_3} A^{s_4' s_4} \cdots |s_1'\rangle\langle s_1| \; |s_2'\rangle\langle s_2| \; |s_3'\rangle\langle s_3| \; |s_4'\rangle\langle s_4| \cdots$$



The principal advantage of the MPO representation is that it allows arithmetic operations on the operators

$$X = Y + Z \rightarrow W_X = W_Y \oplus W_Z$$

Dimension increases: $M_X = M_Y + M_Z$

$$X = YZ \rightarrow W_X = W_Y \otimes W_Z$$

Dimension increases: $M_X = M_Y M_Z$
For finite, open systems, the boundary components are 1-dimensional.
'Orthogonalization' of the MPO keeps the dimension under control.

- Orthonormalizing an MPO in the same way as an MPS is a bad idea
  - Norm of an MPO is the Frobenius norm $\sim \dim(\mathcal{H})$
  - Difference in norms of two operators also scales as $\sim \dim(\mathcal{H})$
  - As soon as this difference exceeds 10^{16}, all precision in the orthogonalization transform is lost
- Instead, collapse only those rows/columns that are *parallel*, ie. identical, up to a constant

Some examples:

Sum of local terms $H = \sum_i X_i$

$$W_H = \begin{pmatrix} I & 0 \\ X & I \end{pmatrix}$$

Boundary vectors $\begin{pmatrix} 0 & I \end{pmatrix}$ and $\begin{pmatrix} I \\ 0 \end{pmatrix}$

Sum of nearest-neighbor terms $H = \sum_i X_i Y_{i+1}$

$$W_H = \begin{pmatrix} I & 0 & 0 \\ Y & 0 & 0 \\ 0 & X & I \end{pmatrix}$$

Sums of operators can always be represented as *lower triangular* forms

- Orthonormalizing an MPO in the same way as an MPS is a bad idea
  - Norm of an MPO is the Frobenius norm $\sim \dim(\mathcal{H})$
  - Difference in norms of two operators also scales as $\sim \dim(\mathcal{H})$
  - As soon as this difference exceeds 10^{16}, all precision in the orthogonalization transform is lost
- Instead, collapse only those rows/columns that are *parallel*, ie. identical, up to a constant

Some examples:

Sum of local terms $H = \sum_i X_i$

$$W_H = \begin{pmatrix} I & 0 \\ X & I \end{pmatrix}$$ Boundary vectors $\begin{pmatrix} 0 & I \end{pmatrix}$ and $\begin{pmatrix} I \\ 0 \end{pmatrix}$

Sum of nearest-neighbor terms $H = \sum_i X_i Y_{i+1}$

$$W_H = \begin{pmatrix} I & 0 & 0 \\ Y & 0 & 0 \\ 0 & X & I \end{pmatrix}$$

Sums of operators can always be represented as *lower triangular* forms

- Orthonormalizing an MPO in the same way as an MPS is a bad idea
  - Norm of an MPO is the Frobenius norm $\sim \dim(\mathcal{H})$
  - Difference in norms of two operators also scales as $\sim \dim(\mathcal{H})$
  - As soon as this difference exceeds 10^{16}, all precision in the orthogonalization transform is lost
- Instead, collapse only those rows/columns that are *parallel*, ie. identical, up to a constant

Some examples:

Sum of local terms $H = \sum_i X_i$

$$W_H = \begin{pmatrix} I & 0 \\ X & I \end{pmatrix} \qquad \text{Boundary vectors } \begin{pmatrix} 0 & I \end{pmatrix} \text{ and } \begin{pmatrix} I \\ 0 \end{pmatrix}$$

Sum of nearest-neighbor terms $H = \sum_i X_i Y_{i+1}$

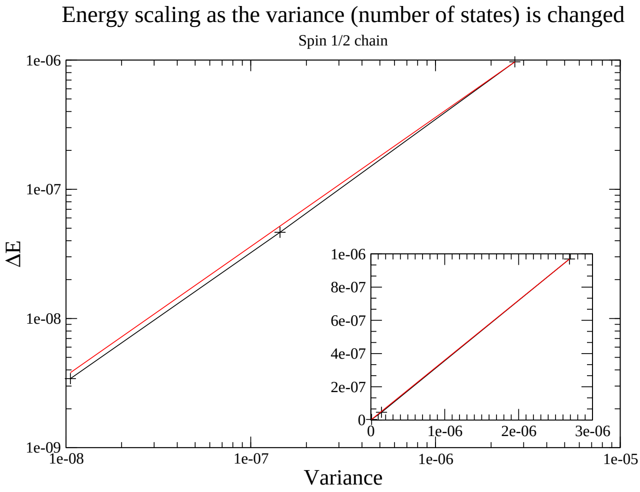
$$W_H = \begin{pmatrix} I & 0 & 0 \\ Y & 0 & 0 \\ 0 & X & I \end{pmatrix}$$

Sums of operators can always be represented as *lower triangular* forms

Calculating observables of ‘complicated’ operators is often easy

Error estimates:

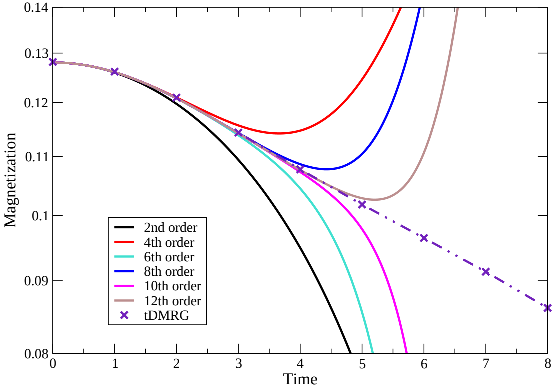
- truncation error: a property of the *algorithm*, not always well defined
- variance $\langle H\rangle^2 - \langle H^2\rangle$: a propery of the *state*, always well-defined



The limits of what is possible (in an hour or so):

$$A(t) = A + \frac{it}{\hbar}[H,A] - \frac{t^2}{2!\hbar^2}[H,[H,A]] - \frac{it^3}{3!\hbar^3}[H,[H,[H,A]]] + \cdots \, .$$

Relaxation of an impurity in the SIAM



$\Gamma = 0.1$, $U = 0.2$, $h_0 = 0.1$, $\epsilon_{d0} = 0.05$, on a log-discretized Wilson chain with $\Lambda = 1.8$.

At time $t = 0$, the Hamiltonian was switched to $h_1 = 0$ and $\epsilon_{d1} = -0.1$.

command line: `mp-expectation siam:"[H,[H,[H,[H,Sz(0)]]]]" psi`

DMRG in the infinite size limit (arxiv:0804.2509)

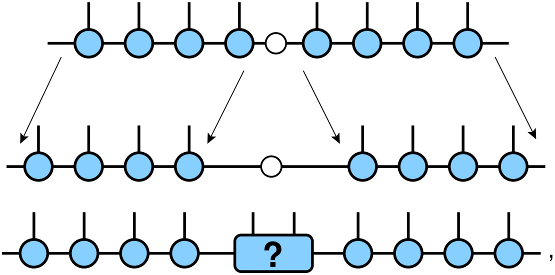
Infinite-size translationally invariant MPS

- The “infinite size” DMRG algorithm has existed since the year 0AD
- It doesn’t produce a translationally invariant MPS fixed point
- No prescription for constructing the initial wavefunction at next iteration
- Rarely used in the literature, and often incorrectly
- iTEBD produces a translationally invariant MPS, but for groundstates imaginary time evolution is not so fast

DMRG in the infinite size limit (arxiv:0804.2509)

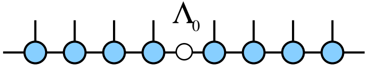
Infinite-size translationally invariant MPS

- The "infinite size" DMRG algorithm has existed since the year 0AD
- It doesn't produce a translationally invariant MPS fixed point
- No prescription for constructing the initial wavefunction at next iteration
- Rarely used in the literature, and often incorrectly
- iTEBD produces a translationally invariant MPS, but for groundstates imaginary time evolution is not so fast

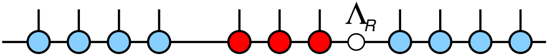


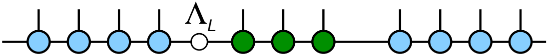
A recurrence relation for MPS

Suppose we can an initial state:



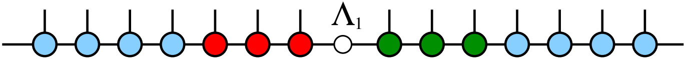
Suppose we also have the same MPS with an extra unit cell:





Note: Λ_L and Λ_R are not necessarily diagonal

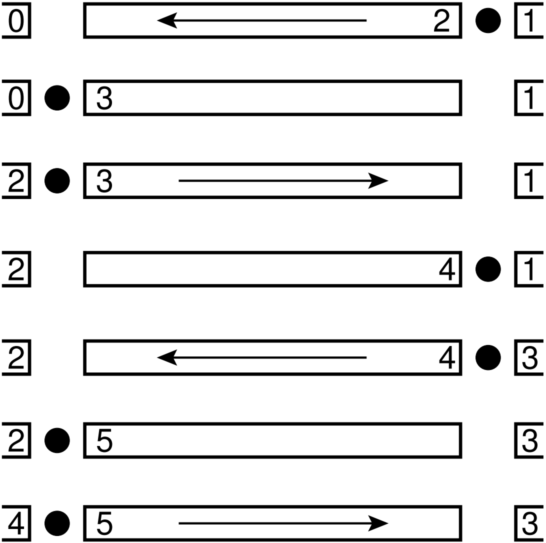
Now we can insert one more unit cell:



$$\Lambda_1 = \Lambda_R \, \Lambda_0^{-1} \, \Lambda_L$$

Variant of the finite system algorithm

- Different treatment of the boundaries



$$\Lambda_{23} = \Lambda_{21}\,\Lambda_{01}^{-1}\,\Lambda_{03}$$

$$\Lambda_{43} = \Lambda_{41}\,\Lambda_{21}^{-1}\,\Lambda_{23}$$

$$\Lambda_{45} = \Lambda_{43}\,\Lambda_{23}^{-1}\,\Lambda_{25}$$

Comparison with iTEBD

- Local minimization is *much* faster than imaginary time evolution
- Unit cell can be an arbitrary size $(1, 2, \ldots)$
- Efficiency is largely independent of unit cell size
- Scaling $\Delta\tau \to 0$ is not needed
- Not yet any scheme for krlov-based time evolution

iTEBD and iDMRG are the same algorithm, except the update works slightly differently

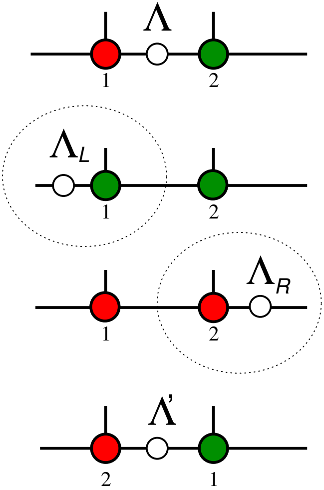
- iTEBD: evolve a single bond at the middle of the chain
- iDMRG: locally minimize the energy with respect to the Hamiltonian of the whole system

Comparison with iTEBD

- Local minimization is *much* faster than imaginary time evolution
- Unit cell can be an arbitrary size $(1, 2, \ldots)$
- Efficiency is largely independent of unit cell size
- Scaling $\Delta\tau \to 0$ is not needed
- Not yet any scheme for krlov-based time evolution

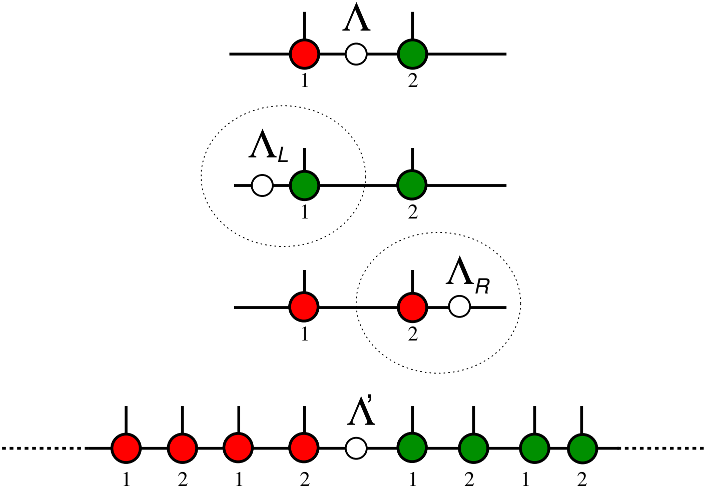
iTEBD and iDMRG are the same algorithm, except the update works slightly differently

- iTEBD: evolve a single bond at the middle of the chain
- iDMRG: locally minimize the energy with respect to the Hamiltonian of the whole system



$\Lambda' = \Lambda_R \, \Lambda^{-1} \, \Lambda_L$ gives an initial state for the evolution

With respect to the whole lattice, the MPS is *always orthogonal*

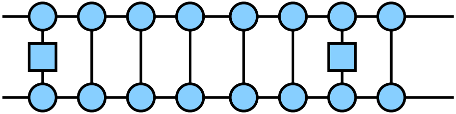


$\Lambda' = \Lambda_R \, \Lambda^{-1} \, \Lambda_L$ gives an initial state for the evolution

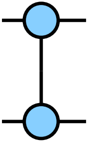
With respect to the whole lattice, the MPS is *always orthogonal*

Expectation Values

Correlation functions



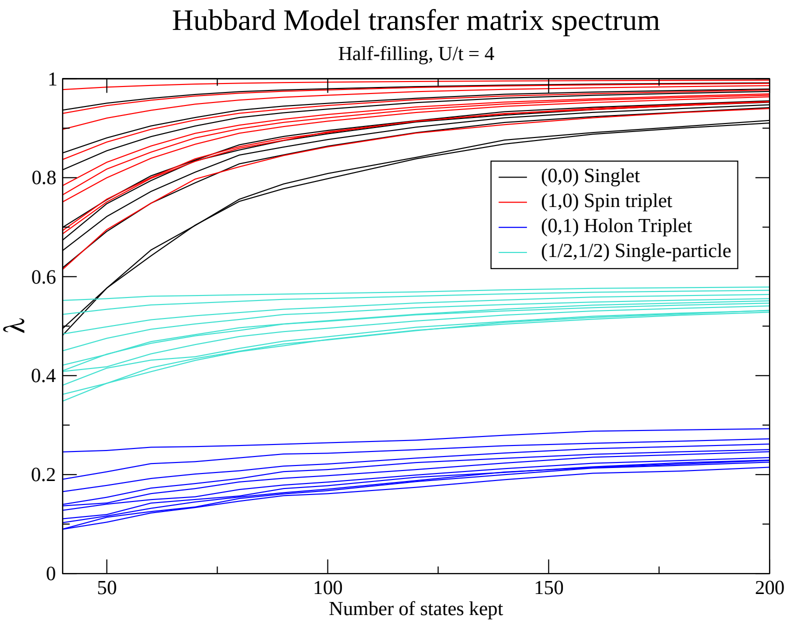
The form of correlation functions are determined by the eigenvalues of the *transfer operator*



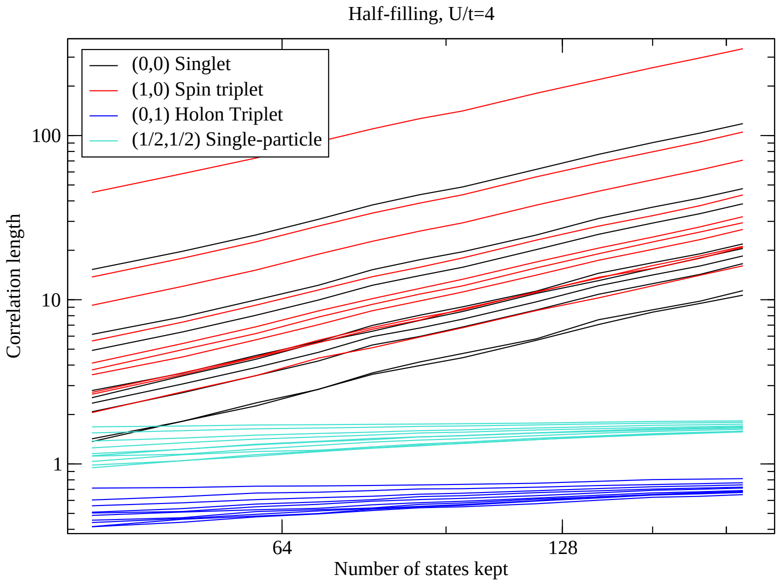
- All eigenvalues ≤ 1
- One eigenvalue equal to 1, corresponding to the identity operator

Expansion in terms of eigenspectrum λ_i:

$$\langle O(x) O(y) \rangle = \sum_i a_i \, \lambda_i^{|y-x|}$$


Hubbard Model transfer matrix spectrum
Half-filling, U/t = 4
λ
1
0.8
0.6
0.4
0.2
0
50
100
150
200
Number of states kept
(0,0) Singlet
(1,0) Spin triplet
(0,1) Holon Triplet
(1/2,1/2) Single-particle

Hubbard model transfer matrix spectrum



CFT Parameters

For a critial mode, the correlation length increases with number of states m as a power law,

$$\xi \sim m^{\kappa}$$

[T. Nishino, K. Okunishi, M. Kikuchi, Phys. Lett. A **213**, 69 (1996)
M. Andersson, M. Boman, S. Östlund, Phys. Rev. B **59**, 10493 (1999)
L. Tagliacozzo, Thiago. R. de Oliveira, S. Iblisdir, J. I. Latorre, Phys. Rev. B **78**, 024410 (2008)]

This exponent is a function *only* of the central charge,

$$\kappa = \frac{6}{\sqrt{12c} + c}$$

[Pollmann et al, arxiv:0812.2903, see poster in foyer]

Can we find an expression for the scaling dimension?

Suppose we have a two-point correlator that has a power-law at large distances

$$\langle O(x)O(y)\rangle = |y - x|^{-2\Delta}$$

As we increase the number of states kept m the correlation length increases, so the region of validity of the power law increases.

- Take two different calculations with m_1 and m_2
- Correlation lengths ξ_1 and ξ_2
- We expect: $\dfrac{O(\xi_2)}{O(\xi_1)} = \left(\dfrac{\xi_2}{\xi_1}\right)^{\Delta}$
- We have: $O(x) \sim a\,\lambda^x$ (with $\xi = -1/(1-\lambda)$)
- Prefactor a is overlap of operator O with next-leading eigenvector of transfer operator
- $\cdots$ algebra $\cdots$

$$a = (1 - \lambda)^{\Delta}$$

This gives directly the operator scaling dimensions by direct fit

Suppose we have a two-point correlator that has a power-law at large distances

$$\langle O(x)O(y)\rangle = |y-x|^{-2\Delta}$$

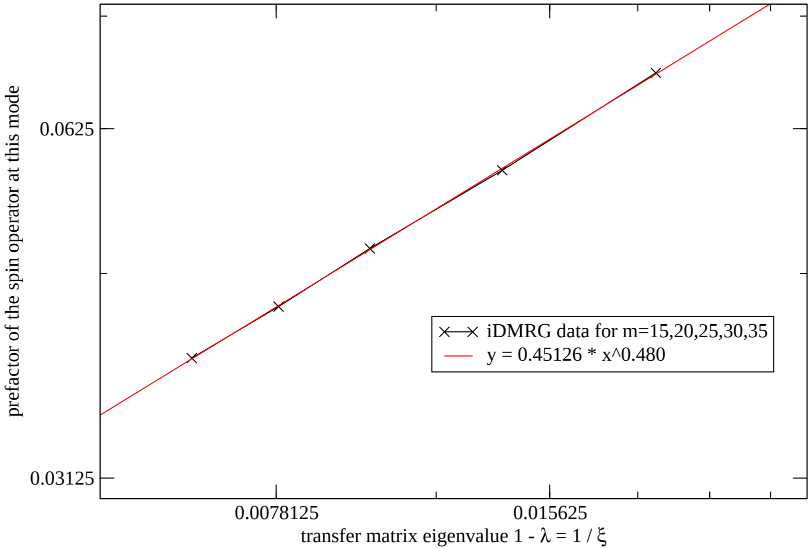
As we increase the number of states kept m the correlation length increases, so the region of validity of the power law increases.

- Take two different calculations with m_1 and m_2
- Correlation lengths ξ_1 and ξ_2
- We expect: $\frac{O(\xi_2)}{O(\xi_1)} = \left(\frac{\xi_2}{\xi_1}\right)^{\Delta}$
- We have: $O(x) \sim a\,\lambda^x$ (with $\xi = -1/(1-\lambda)$)
- Prefactor a is overlap of operator O with next-leading eigenvector of transfer operator
- $\cdots$ algebra $\cdots$

$$a = (1-\lambda)^{\Delta}$$

This gives directly the operator scaling dimensions by direct fit

Heisenberg model fit for the scaling dimension

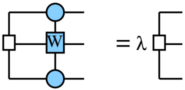


Expectation values of MPO's

We have seen that we can write many interesting operators in the form of a matrix product operator

- Can we evaluate the expectation value of an arbitrary MPO?

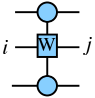
If the MPO has no Jordan structure, this is a simple eigenvalue problem



For a lower triangular MPO, this doesn't work.

- But we can make use of the triangular structure
- index by index, each component is a function only of the previously calculated terms

Choose bond indices i, j of W_{ij}, and denote $T_{W_{ij}}$

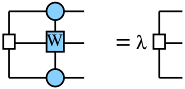


Expectation values of MPO's

We have seen that we can write many interesting operators in the form of a matrix product operator

- Can we evaluate the expectation value of an arbitrary MPO?

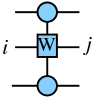
If the MPO has no Jordan structure, this is a simple eigenvalue problem



For a lower triangular MPO, this doesn't work.

- But we can make use of the triangular structure
- index by index, each component is a function only of the previously calculated terms

Choose bond indices i, j of W_{ij}, and denote $T_{W_{ij}}$



At the n^{th} iteration, we have

$$\tilde{E}_i[n] = T_{W_{ii}}(\tilde{E}_i[n-1]) + \underbrace{\sum_{j>i} T_{W_{ij}}(\tilde{E}_j[n-1])}_{\text{const. operator}}$$

Case 1: $W_{ii} = 0$

- $\tilde{E}_i =$ const

Case 2: $W_{ii} = xI$ (proportional to identity operator)

- A geometric series with respect to transfer matrix eigenvalues. $\sum_i c_i \sum_n \lambda_i^n$. Component in the direction of the identity operator diverges linearly, remaining components can be solved efficently:

$$(1-T)\tilde{E}_i = \text{const. operator}$$

Similar scheme works if x is complex

Case 3: W_{ii} unitary

- similar, keep track of diverging component giving long-range string order

At the n^{th} iteration, we have

$$\tilde{E}_i[n] = T_{W_{ii}}(\tilde{E}_i[n-1]) + \underbrace{\sum_{j>i} T_{W_{ij}}(\tilde{E}_j[n-1])}_{\text{const. operator}}$$

Case 1: $W_{ii} = 0$

- $\tilde{E}_i = \text{const}$

Case 2: $W_{ii} = xI$ (proportional to identity operator)

- A geometric series with respect to transfer matrix eigenvalues. $\sum_i c_i \sum_n \lambda_i^n$. Component in the direction of the identity operator diverges linearly, remaining components can be solved efficently:

$$(1 - T)\tilde{E}_i = \text{const. operator}$$

Similar scheme works if x is complex

Case 3: W_{ii} unitary

- similar, keep track of diverging component giving long-range string order

At the n^{th} iteration, we have

$$\tilde{E}_i[n] = T_{W_{ii}}(\tilde{E}_i[n-1]) + \underbrace{\sum_{j>i} T_{W_{ij}}(\tilde{E}_j[n-1])}_{\text{const. operator}}$$

Case 1: $W_{ii} = 0$

- $\tilde{E}_i =$ const

Case 2: $W_{ii} = xI$ (proportional to identity operator)

- A geometric series with respect to transfer matrix eigenvalues. $\sum_i c_i \sum_n \lambda_i^n$. Component in the direction of the identity operator diverges linearly, remaining components can be solved efficently:

$$(1-T)\tilde{E}_i = \text{const. operator}$$

Similar scheme works if x is complex

Case 3: W_{ii} unitary

- similar, keep track of diverging component giving long-range string order

At the n^{th} iteration, we have

$$\tilde{E}_i[n] = T_{W_{ii}}(\tilde{E}_i[n-1]) + \underbrace{\sum_{j>i} T_{W_{ij}}(\tilde{E}_j[n-1])}_{\text{const. operator}}$$

Case 1: $W_{ii} = 0$

- $\tilde{E}_i = \text{const}$

Case 2: $W_{ii} = xI$ (proportional to identity operator)

- A geometric series with respect to transfer matrix eigenvalues. $\sum_i c_i \sum_n \lambda_i^n$. Component in the direction of the identity operator diverges linearly, remaining components can be solved efficently:

$$(1 - T)\tilde{E}_i = \text{const. operator}$$

Similar scheme works if x is complex

Case 3: W_{ii} unitary

- similar, keep track of diverging component giving long-range string order

Summary

- MPO's are convienent in practical applications
- Variance $\langle H \rangle^2 - \langle H^2 \rangle$ as a general purpose measure of accuracy, scales similarly to the truncation error in DMRG
- Perturbative expansions, commutators, higher moments, ...

- iDMRG is a very efficient method to construct translationally invariant thermodynamic states
- All expectation values can be expressed in terms of the eigenmodes of the transfer matrix
- Scaling with respect to m (or D, or χ) can give power laws
- Easy to extract scaling dimensions of operators
- For a critical state, can we extract the primary fields and the operator algebra of the CFT? Analogies to finite-width transfer matrix spectra.