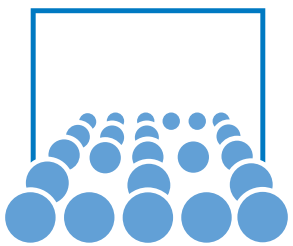
Approximation Techniques for Special Eigenvalue Problems

Konrad Waldherr

June 2011



Joint work with Thomas Huckle

Outline

Problem: Computation of Ground States

Given:

- Physical system: 1D spin chain with N particles

Problem: Computation of Ground States

Given:

- Physical system: 1D spin chain with N particles

- Interaction within the system (e.g. nearest-neighbor interaction)

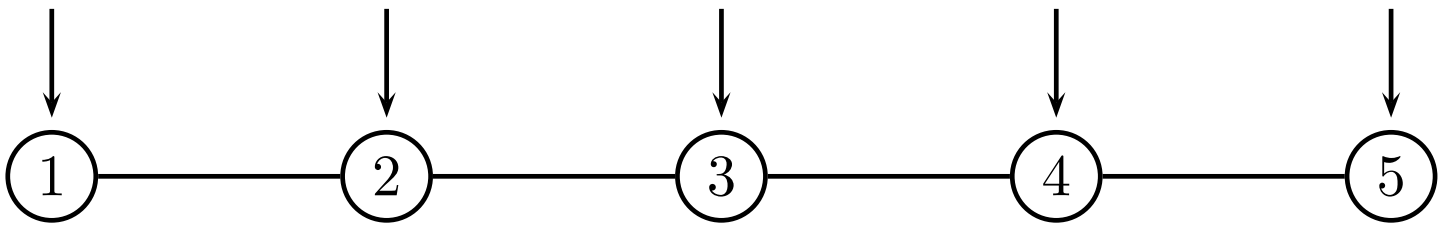
Problem: Computation of Ground States

Given:

- Physical system: 1D spin chain with N particles

- Interaction within the system (e.g. nearest-neighbor interaction)

- External interaction (e.g. exterior magnetic field)



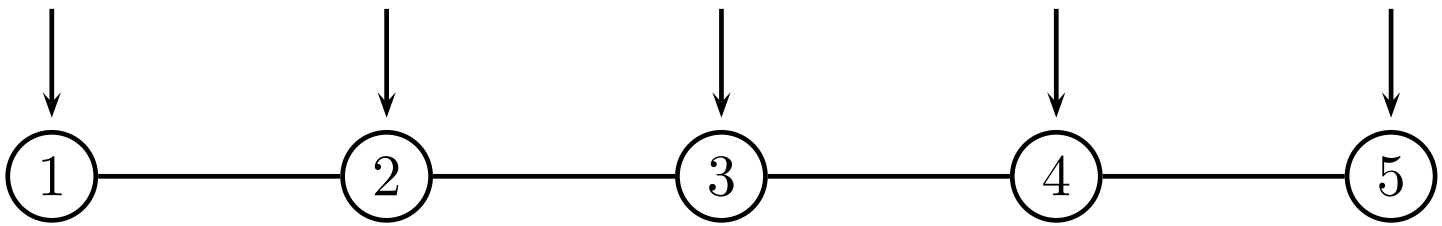
Problem: Computation of Ground States

Given:

- Physical system: 1D spin chain with N particles

- Interaction within the system (e.g. nearest-neighbor interaction)

- External interaction (e.g. exterior magnetic field)



Goal:

- Find ground state, i.e. the state related to the smallest energy of the system.

Mathematical Model

- Any state of the system is represented by a vector $x \in \mathbb{C}^{2^N}$.
- The physical system can be described by the Hamiltonian $H \in \mathbb{C}^{2^N \times 2^N}$.

Mathematical Model

- Any state of the system is represented by a vector $x \in \mathbb{C}^{2^N}$.
- The physical system can be described by the Hamiltonian $H \in \mathbb{C}^{2^N \times 2^N}$.
- The Hamiltonian may be formulated as a weighted sum of Kronecker products of Pauli and identity matrices.
- Pauli matrices:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

- Kronecker product:

$$A \otimes B := \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix} \otimes B = \begin{pmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{pmatrix}$$

Mathematical Model

- Any state of the system is represented by a vector $x \in \mathbb{C}^{2^N}$.
- The physical system can be described by the Hamiltonian $H \in \mathbb{C}^{2^N \times 2^N}$.
- The Hamiltonian may be formulated as a weighted sum of Kronecker products of Pauli and identity matrices.
- Pauli matrices:

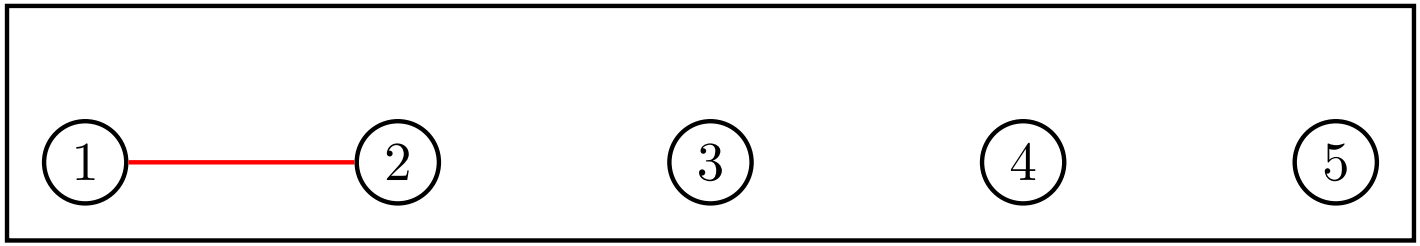
$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

- Kronecker product:

$$A \otimes B := \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix} \otimes B = \begin{pmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{pmatrix}$$

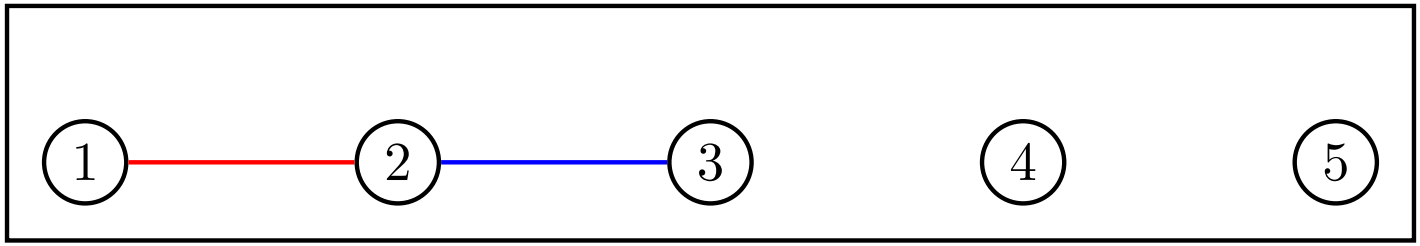
- Then, the ground state corresponds to the **eigenvector related to the smallest eigenvalue**.

Example: Ising-type Hamiltonian



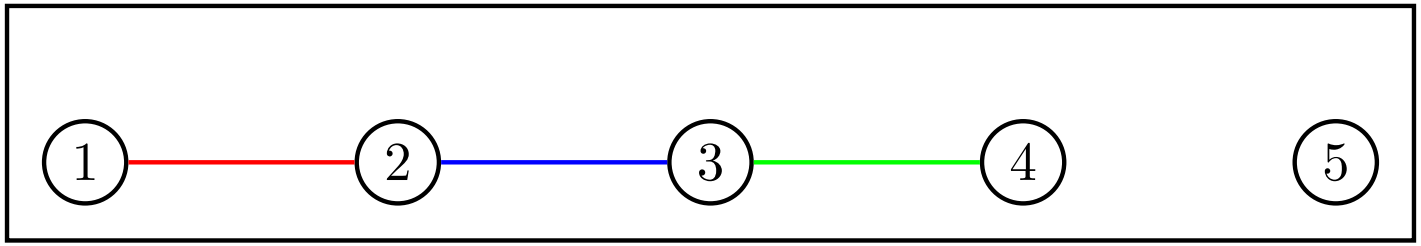
$$H = \sigma_z \otimes \sigma_z \otimes I \otimes I \otimes I$$

Example: Ising-type Hamiltonian



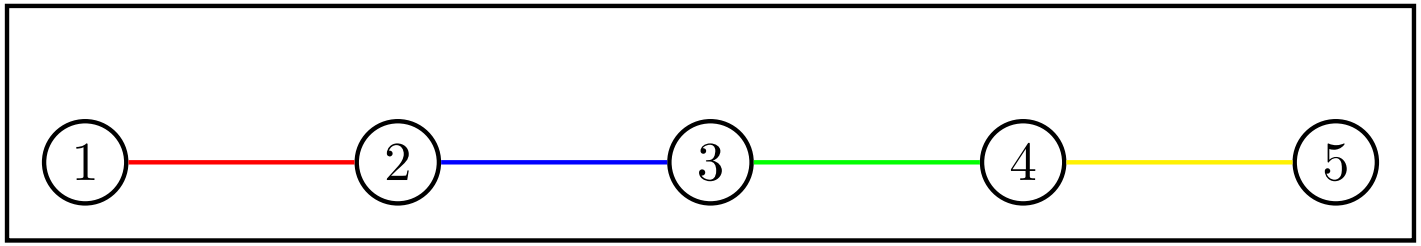
$$
\begin{aligned}
H &= \sigma_z \otimes \sigma_z \otimes I \otimes I \otimes I \\
&+ I \otimes \sigma_z \otimes \sigma_z \otimes I \otimes I
\end{aligned}
$$

Example: Ising-type Hamiltonian



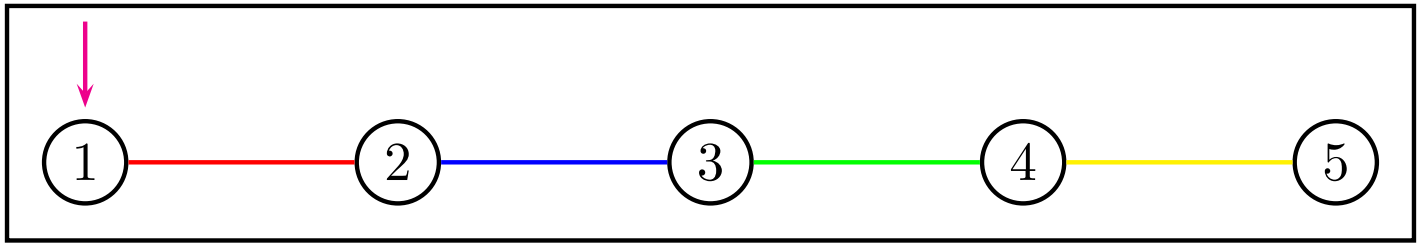
$$
\begin{aligned}
H &= \sigma_z \otimes \sigma_z \otimes I \otimes I \otimes I \\
&+ I \otimes \sigma_z \otimes \sigma_z \otimes I \otimes I \\
&+ I \otimes I \otimes \sigma_z \otimes \sigma_z \otimes I
\end{aligned}
$$

Example: Ising-type Hamiltonian



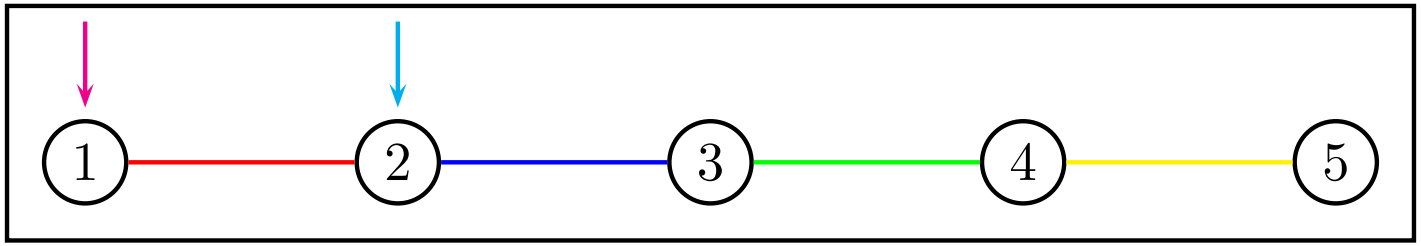
$$
\begin{aligned}
H &= \sigma_z \otimes \sigma_z \otimes I \otimes I \otimes I \\
&+ I \otimes \sigma_z \otimes \sigma_z \otimes I \otimes I \\
&+ I \otimes I \otimes \sigma_z \otimes \sigma_z \otimes I \\
&+ I \otimes I \otimes I \otimes \sigma_z \otimes \sigma_z
\end{aligned}
$$

Example: Ising-type Hamiltonian



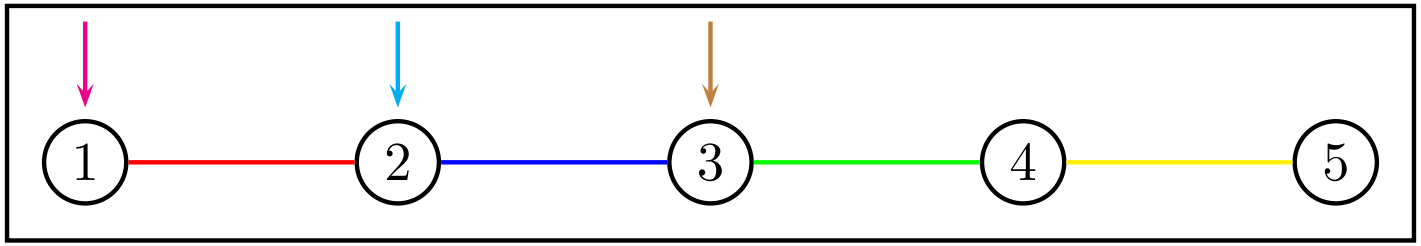
$$
\begin{aligned}
H &= \sigma_z \otimes \sigma_z \otimes I \otimes I \otimes I \\
&+ I \otimes \sigma_z \otimes \sigma_z \otimes I \otimes I \\
&+ I \otimes I \otimes \sigma_z \otimes \sigma_z \otimes I \\
&+ I \otimes I \otimes I \otimes \sigma_z \otimes \sigma_z \\
&+ \sigma_x \otimes I \otimes I \otimes I \otimes I
\end{aligned}
$$

Example: Ising-type Hamiltonian



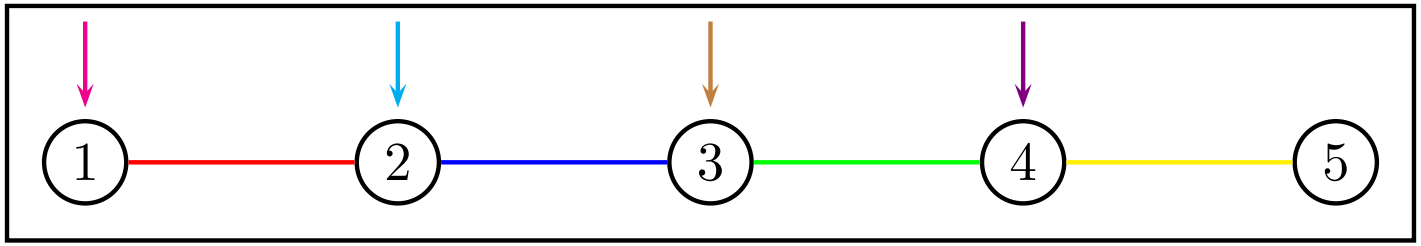
$$
\begin{aligned}
H &= \sigma_z \otimes \sigma_z \otimes I \otimes I \otimes I \\
&+ I \otimes \sigma_z \otimes \sigma_z \otimes I \otimes I \\
&+ I \otimes I \otimes \sigma_z \otimes \sigma_z \otimes I \\
&+ I \otimes I \otimes I \otimes \sigma_z \otimes \sigma_z \\
&+ \sigma_x \otimes I \otimes I \otimes I \otimes I \\
&+ I \otimes \sigma_x \otimes I \otimes I \otimes I
\end{aligned}
$$

Example: Ising-type Hamiltonian



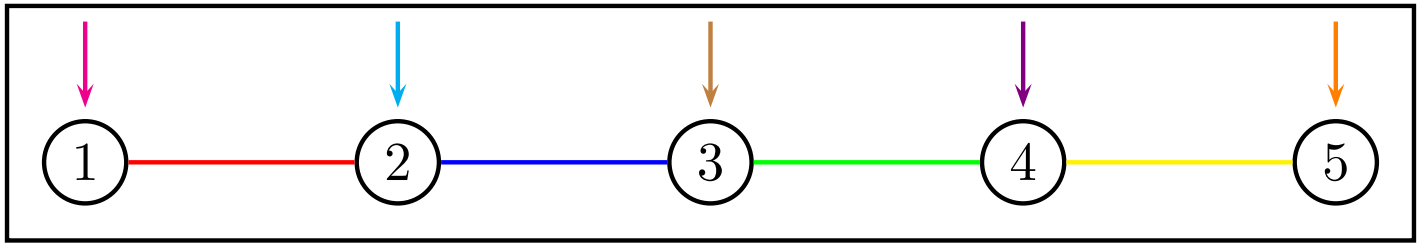
$$
\begin{aligned}
H &= \sigma_z \otimes \sigma_z \otimes I \otimes I \otimes I \\
&+ I \otimes \sigma_z \otimes \sigma_z \otimes I \otimes I \\
&+ I \otimes I \otimes \sigma_z \otimes \sigma_z \otimes I \\
&+ I \otimes I \otimes I \otimes \sigma_z \otimes \sigma_z \\
&+ \sigma_x \otimes I \otimes I \otimes I \otimes I \\
&+ I \otimes \sigma_x \otimes I \otimes I \otimes I \\
&+ I \otimes I \otimes \sigma_x \otimes I \otimes I
\end{aligned}
$$

Example: Ising-type Hamiltonian



$$
\begin{aligned}
H &= \sigma_z \otimes \sigma_z \otimes I \otimes I \otimes I \\
&+ I \otimes \sigma_z \otimes \sigma_z \otimes I \otimes I \\
&+ I \otimes I \otimes \sigma_z \otimes \sigma_z \otimes I \\
&+ I \otimes I \otimes I \otimes \sigma_z \otimes \sigma_z \\
&+ \sigma_x \otimes I \otimes I \otimes I \otimes I \\
&+ I \otimes \sigma_x \otimes I \otimes I \otimes I \\
&+ I \otimes I \otimes \sigma_x \otimes I \otimes I \\
&+ I \otimes I \otimes I \otimes \sigma_x \otimes I
\end{aligned}
$$

Example: Ising-type Hamiltonian



$$
\begin{aligned}
H &= \sigma_z \otimes \sigma_z \otimes I \otimes I \otimes I \\
&+ I \otimes \sigma_z \otimes \sigma_z \otimes I \otimes I \\
&+ I \otimes I \otimes \sigma_z \otimes \sigma_z \otimes I \\
&+ I \otimes I \otimes I \otimes \sigma_z \otimes \sigma_z \\
&+ \sigma_x \otimes I \otimes I \otimes I \otimes I \\
&+ I \otimes \sigma_x \otimes I \otimes I \otimes I \\
&+ I \otimes I \otimes \sigma_x \otimes I \otimes I \\
&+ I \otimes I \otimes I \otimes \sigma_x \otimes I \\
&+ I \otimes I \otimes I \otimes I \otimes \sigma_x
\end{aligned}
$$

Problem Setting

- General formulation of the Hamiltonian H:

$$H = \sum_{k=1}^{M} \alpha_k Q_1^{(k)} \otimes \cdots \otimes Q_N^{(k)} \ , Q_i^{(k)} \in \{id, \sigma_x, \sigma_y, \sigma_z\} \subset \mathbb{C}^{2\times 2}$$

Problem Setting

- General formulation of the Hamiltonian H:

$$H = \sum_{k=1}^{M} \alpha_k Q_1^{(k)} \otimes \cdots \otimes Q_N^{(k)} \ , Q_i^{(k)} \in \{id, \sigma_x, \sigma_y, \sigma_z\} \subset \mathbb{C}^{2\times 2}$$

- Variational ansatz: Minimization of the Rayleigh quotient:

$$\min_{x \neq 0} \frac{x^{\mathrm{H}} H x}{x^{\mathrm{H}} x}$$

Problem Setting

- General formulation of the Hamiltonian H:

$$H = \sum_{k=1}^{M} \alpha_k Q_1^{(k)} \otimes \cdots \otimes Q_N^{(k)} \ , Q_i^{(k)} \in \{id, \sigma_x, \sigma_y, \sigma_z\} \subset \mathbb{C}^{2\times 2}$$

- Variational ansatz: Minimization of the Rayleigh quotient:

$$\min_{x \neq 0} \frac{x^{\mathrm{H}} H x}{x^{\mathrm{H}} x}$$

- Problems:
  - the vector space $\mathcal{V} = \mathbb{C}^{2^N}$ grows exponentially in N,
  - the problem is not generic.

Problem Setting

- General formulation of the Hamiltonian H:

$$H = \sum_{k=1}^{M} \alpha_k Q_1^{(k)} \otimes \cdots \otimes Q_N^{(k)} \ , Q_i^{(k)} \in \{id, \sigma_x, \sigma_y, \sigma_z\} \subset \mathbb{C}^{2\times 2}$$

- Variational ansatz: Minimization of the Rayleigh quotient:

$$\min_{x \neq 0} \frac{x^{\mathrm{H}} H x}{x^{\mathrm{H}} x}$$

- Problems:
  - the vector space $\mathcal{V} = \mathbb{C}^{2^N}$ grows exponentially in N,
  - the problem is not generic.
- Idea: Find data-sparse representation formats that allow to overcome the curse of dimensionality.

Data-sparse representation formats

- The data-sparse format has to allow for
  1. proper reproduction of the physical setting (e.g. nearest-neighbor interaction),
  2. representations that are only polynomial in N,
  3. good approximation properties,
  4. efficient evaluation of the Rayleigh quotient

$$\min_{x \in \mathcal{U}} \frac{x^{\mathrm{H}} H x}{x^{\mathrm{H}} x} \tag{1}$$

$\Longrightarrow$ efficient computation of both Hx and $y^{\mathrm{H}} x$.

Data-sparse representation formats

- The data-sparse format has to allow for
  1. proper reproduction of the physical setting (e.g. nearest-neighbor interaction),
  2. representations that are only polynomial in N,
  3. good approximation properties,
  4. efficient evaluation of the Rayleigh quotient

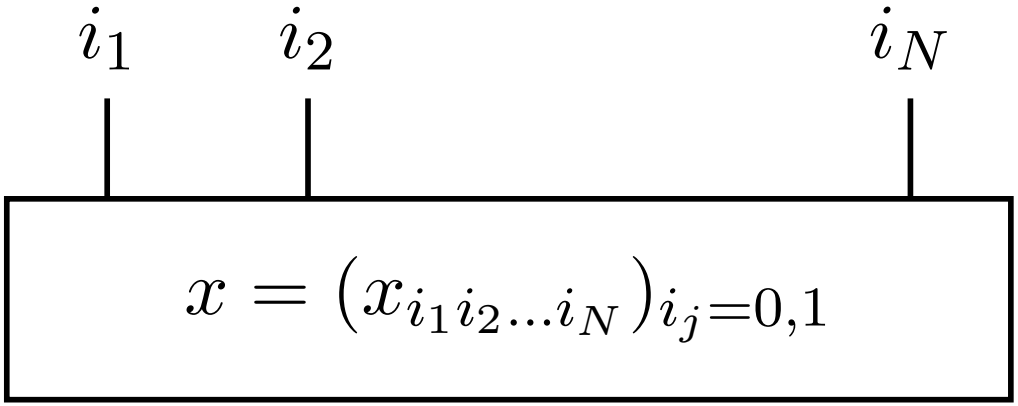
$$\min_{x \in \mathcal{U}} \frac{x^{\mathrm{H}} H x}{x^{\mathrm{H}} x} \tag{1}$$

  $\Longrightarrow$ efficient computation of both Hx and $y^{\mathrm{H}} x$.

- Problem setting:
  - Not: Find a data-sparse format of a given data structure
  - But: Use the data-sparse format **as an ansatz** for the minimization of the Rayleigh quotient (1).

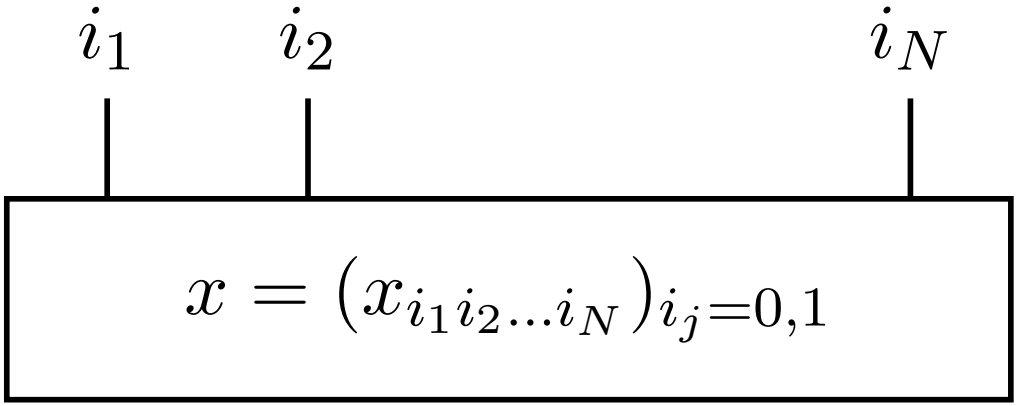
Tensor Decompositions

- Any state $x \in \mathbb{C}^{2^N}$ may be considered as a Nth order tensor:



Tensor Decompositions

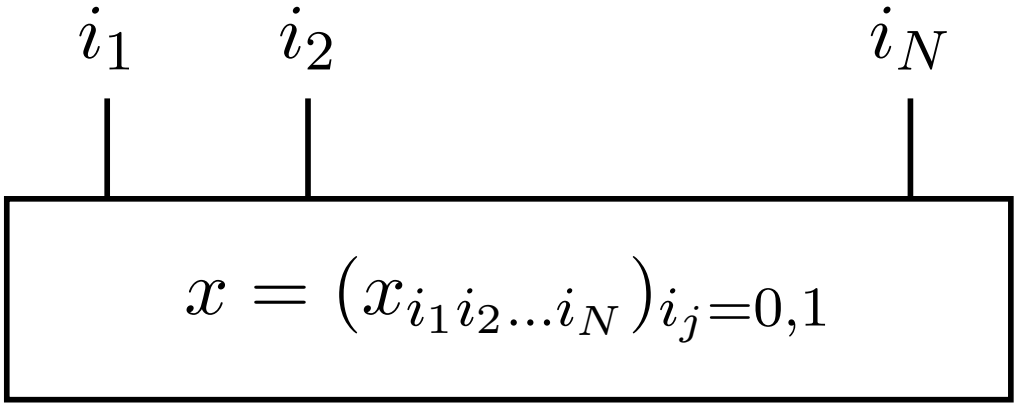
- Any state $x \in \mathbb{C}^{2^N}$ may be considered as a Nth order tensor:



- Goal: Find convenient tensor decomposition as ansatz for (1)

Tensor Decompositions

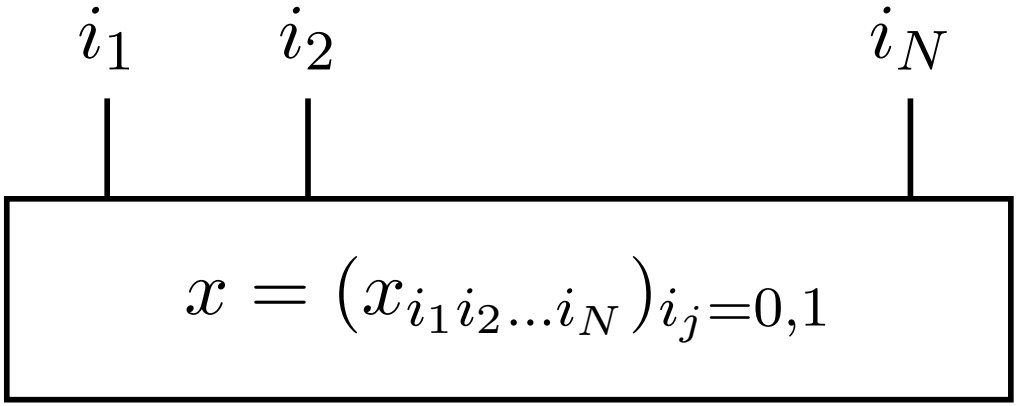
- Any state $x \in \mathbb{C}^{2^N}$ may be considered as a Nth order tensor:



- Goal: Find convenient tensor decomposition as ansatz for (1)
- Classical decomposition formats:
  - Tucker decomposition,
  - Canonical decomposition.

Tensor Decompositions

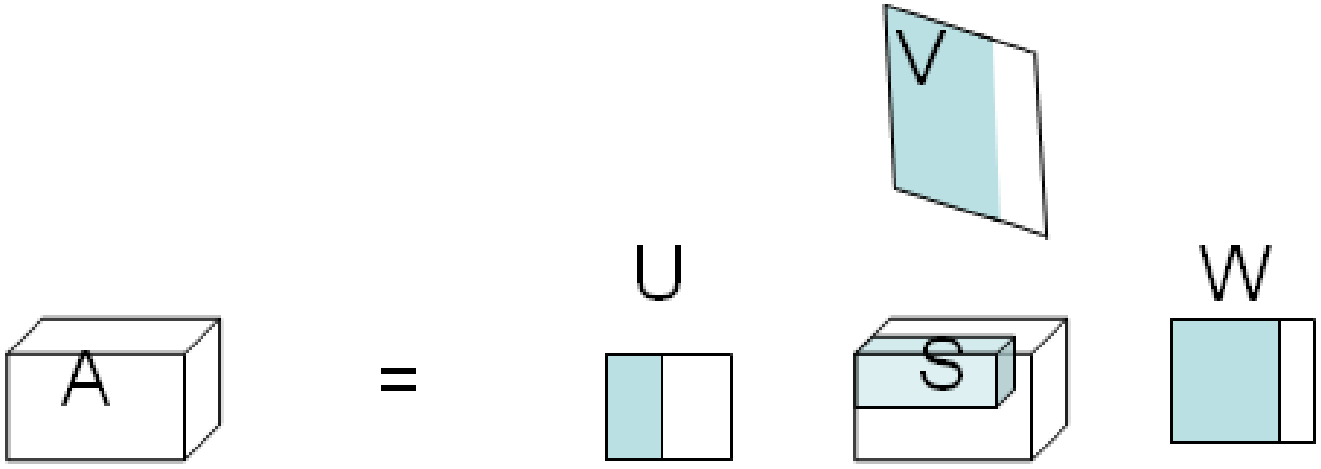
- Any state $x \in \mathbb{C}^{2^N}$ may be considered as a Nth order tensor:



- Goal: Find convenient tensor decomposition as ansatz for (1)
- Classical decomposition formats:
  - Tucker decomposition,
  - Canonical decomposition.
- Physically motivated formats:
  - Matrix Product States (MPS) for $1D$ problems,
  - Projected Entangled Pair States (PEPS) for $2D$ problems.

Decompositions of a tensor

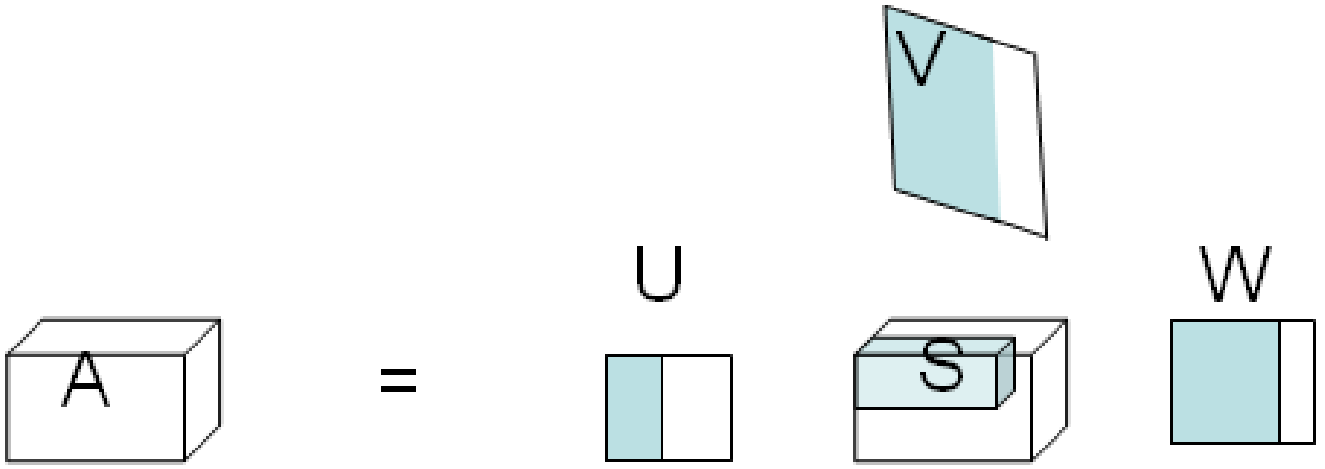
- Tucker decomposition:



Not advantageous in our case of a binary tensor.

Decompositions of a tensor

- Tucker decomposition:



Not advantageous in our case of a binary tensor.

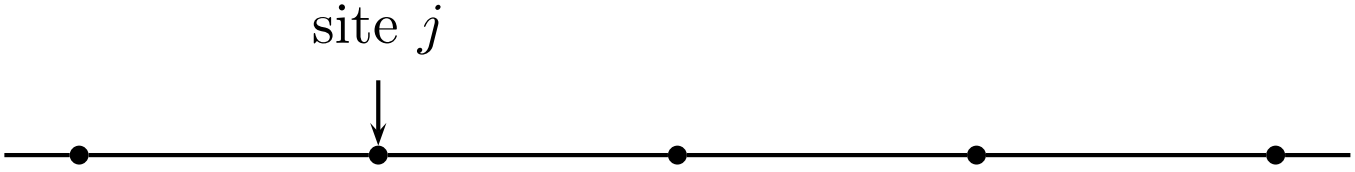
- Canonical decomposition (CP):

Allows generalizations in several ways: (see [Huckle 2011])

- blockings and mixed blockings,
- permutations.

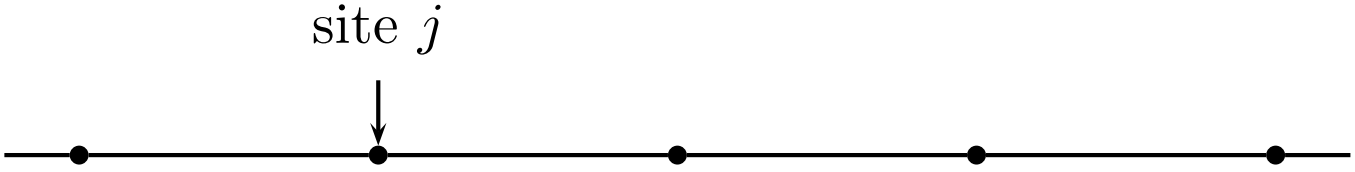
Matrix Product States (MPS)

- Consider again the linear 1D spin chain.

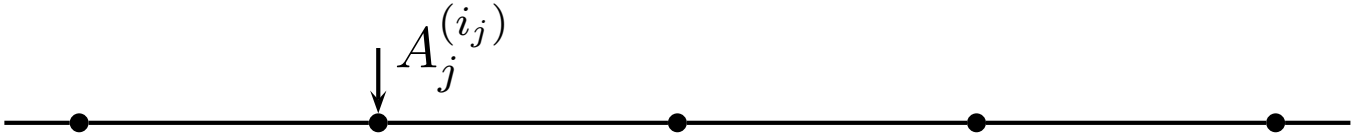


Matrix Product States (MPS)

- Consider again the linear 1D spin chain.

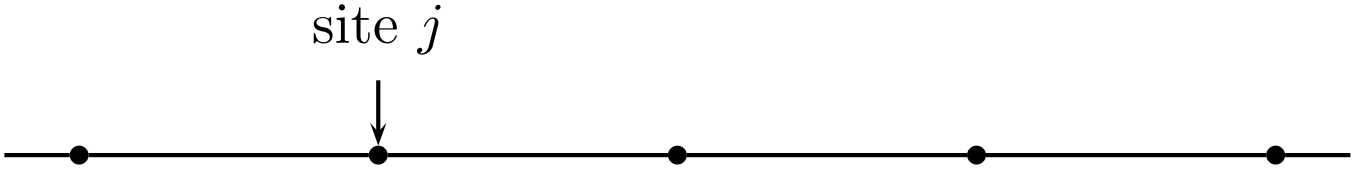


- Each site j is related to a matrix pair $A_j^{(0)}, A_j^{(1)} \in \mathbb{C}^{D_j \times D_{j+1}}$

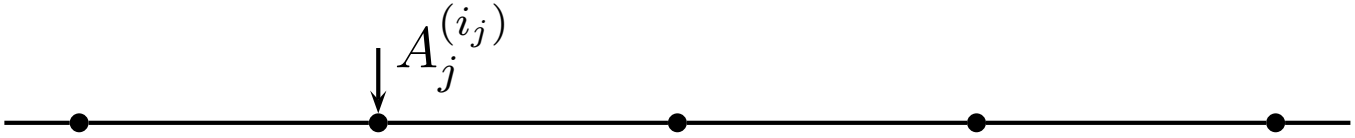


Matrix Product States (MPS)

- Consider again the linear 1D spin chain.

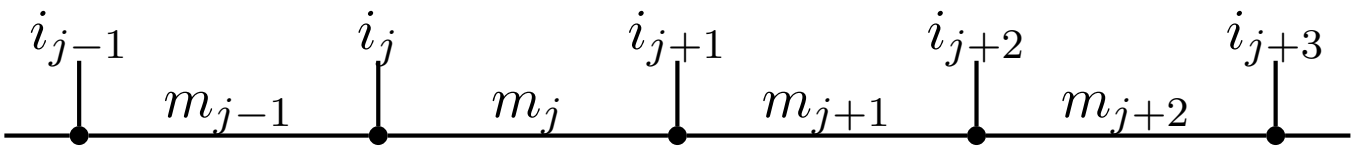


- Each site j is related to a matrix pair $A_j^{(0)}, A_j^{(1)} \in \mathbb{C}^{D_j \times D_{j+1}}$



- For $i = (i_1, i_2, \ldots, i_N)_2$, the vector component x_i is given by

$$x_i = x_{i_1 i_2 \ldots i_N} = \operatorname{tr}\left(A_1^{(i_1)} \cdot A_2^{(i_2)} \cdots A_N^{(i_N)}\right)$$



(1.✓)

Matrix Product States

- Thus, the vector x may be written as

$$x = \sum_{i=1}^{2^N} x_i e_i = \sum_{i_1,\ldots,i_p} \underbrace{\operatorname{tr}\left(A_1^{(i_1)} \cdot A_2^{(i_2)} \cdots A_N^{(i_N)}\right)}_{=x_i = x_{i_1,\ldots,i_N}} \underbrace{e_{i_1 i_2 \ldots i_N}}_{=e_i} .$$

- Binary version of the Tensor Train concept. (see [Oseledets and Tyrtyshnikov, 2009])

Matrix Product States

- Thus, the vector x may be written as

$$x = \sum_{i=1}^{2^N} x_i e_i = \sum_{i_1,\ldots,i_p} \underbrace{\operatorname{tr}\left(A_1^{(i_1)} \cdot A_2^{(i_2)} \cdots A_N^{(i_N)}\right)}_{=x_i = x_{i_1,\ldots,i_N}} \underbrace{e_{i_1 i_2 \ldots i_N}}_{=e_i} .$$

- Binary version of the Tensor Train concept. (see [Oseledets and Tyrtyshnikov, 2009])
- MPS representation requires $2ND^2$ instead of 2^N entries. (2.✓)

Matrix Product States

- Thus, the vector x may be written as

$$x = \sum_{i=1}^{2^N} x_i e_i = \sum_{i_1,\ldots,i_p} \underbrace{\operatorname{tr}\left(A_1^{(i_1)} \cdot A_2^{(i_2)} \cdots A_N^{(i_N)}\right)}_{=x_i = x_{i_1,\ldots,i_N}} \underbrace{e_{i_1 i_2 \ldots i_N}}_{=e_i} .$$

- Binary version of the Tensor Train concept. (see [Oseledets and Tyrtyshnikov, 2009])
- MPS representation requires $2ND^2$ instead of 2^N entries. (2.✓)
- Any state can be represented exactly by an MPS (but at the cost of exponentially growing matrix dimension D),
- For approximation problems, D has to scale only polynomially. (3.✓)

Matrix Product States

Example

- Consider the case $N = 2$ and $D = 1$:

$$\psi = \sum_{i_1,i_2=0}^{1} \underbrace{\left(a_1^{(i_1)} a_2^{(i_2)}\right)}_{=x_i = x_{i_1 i_2}} \underbrace{(e_{i_1} \otimes e_{i_2})}_{=e_i = e_{i_1 i_2}} = \begin{pmatrix} a_1^{(0)} a_2^{(0)} \\ a_1^{(0)} a_2^{(1)} \\ a_1^{(1)} a_2^{(0)} \\ a_1^{(1)} a_2^{(1)} \end{pmatrix} \stackrel{!}{=} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

Matrix Product States

Example

- Consider the case $N = 2$ and $D = 1$:

$$\psi = \sum_{i_1,i_2=0}^{1} \underbrace{\left(a_1^{(i_1)} a_2^{(i_2)}\right)}_{=x_i = x_{i_1 i_2}} \underbrace{(e_{i_1} \otimes e_{i_2})}_{=e_i = e_{i_1 i_2}} = \begin{pmatrix} a_1^{(0)} a_2^{(0)} \\ a_1^{(0)} a_2^{(1)} \\ a_1^{(1)} a_2^{(0)} \\ a_1^{(1)} a_2^{(1)} \end{pmatrix} \stackrel{!}{=} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

- For exact representation:

$$x_1 x_4 = x_2 x_3$$

Matrix Product States

Example

- Consider the case $N = 2$ and $D = 1$:

$$\psi = \sum_{i_1, i_2 = 0}^{1} \underbrace{\left(a_1^{(i_1)} a_2^{(i_2)} \right)}_{= x_i = x_{i_1 i_2}} \underbrace{(e_{i_1} \otimes e_{i_2})}_{= e_i = e_{i_1 i_2}} = \begin{pmatrix} a_1^{(0)} a_2^{(0)} \\ a_1^{(0)} a_2^{(1)} \\ a_1^{(1)} a_2^{(0)} \\ a_1^{(1)} a_2^{(1)} \end{pmatrix} \stackrel{!}{=} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

- For exact representation:

$$x_1 x_4 = x_2 x_3$$

  - e_1, e_2, e_3, e_4 can be represented,
  - However, $e_1 + e_4 = (1, 0, 0, 1)^{\mathrm{T}}$ **cannot** be represented.

Matrix Product States

Example

- Consider the case $N = 2$ and $D = 1$:

$$\psi = \sum_{i_1, i_2 = 0}^{1} \underbrace{\left(a_1^{(i_1)} a_2^{(i_2)} \right)}_{= x_i = x_{i_1 i_2}} \underbrace{(e_{i_1} \otimes e_{i_2})}_{= e_i = e_{i_1 i_2}} = \begin{pmatrix} a_1^{(0)} a_2^{(0)} \\ a_1^{(0)} a_2^{(1)} \\ a_1^{(1)} a_2^{(0)} \\ a_1^{(1)} a_2^{(1)} \end{pmatrix} \stackrel{!}{=} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

- For exact representation:

$$x_1 x_4 = x_2 x_3$$

  - e_1, e_2, e_3, e_4 can be represented,
  - However, $e_1 + e_4 = (1, 0, 0, 1)^{\mathrm{T}}$ **cannot** be represented.
- **MPS are not closed under addition!**

Computations with MPS

- Sum of two MPS vectors

$$x = \sum_{i_1,\ldots,i_N} \mathrm{tr}(A_1^{(i_1)} A_2^{(i_2)} \ldots A_N^{(i_N)}) e_{i_1 \ldots i_N}$$

$$y = \sum_{i_1,\ldots,i_N} \mathrm{tr}(B_1^{(i_1)} B_2^{(i_2)} \ldots B_N^{(i_N)}) e_{i_1 \ldots i_N}$$

Computations with MPS

- Sum of two MPS vectors

$$x = \sum_{i_1,\dots,i_N} \operatorname{tr}(A_1^{(i_1)} A_2^{(i_2)} \dots A_N^{(i_N)}) e_{i_1 \dots i_N}$$

$$y = \sum_{i_1,\dots,i_N} \operatorname{tr}(B_1^{(i_1)} B_2^{(i_2)} \dots B_N^{(i_N)}) e_{i_1 \dots i_N}$$

$$x + y = \sum_{i_1,\dots,i_N} \operatorname{tr}\left[\begin{pmatrix} A_1^{(i_1)} & 0 \\ 0 & B_1^{(i_1)} \end{pmatrix} \underbrace{\begin{pmatrix} A_2^{(i_2)} & 0 \\ 0 & B_2^{(i_2)} \end{pmatrix}}_{D' \times D'} \cdots \begin{pmatrix} A_N^{(i_N)} \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ B_N^{(i_N)} \end{pmatrix} \right] e_{i_1 \dots i_N}$$

Computations with MPS

- Sum of two MPS vectors

$$x = \sum_{i_1,\ldots,i_N} \operatorname{tr}(A_1^{(i_1)} A_2^{(i_2)} \ldots A_N^{(i_N)}) e_{i_1 \ldots i_N}$$

$$y = \sum_{i_1,\ldots,i_N} \operatorname{tr}(B_1^{(i_1)} B_2^{(i_2)} \ldots B_N^{(i_N)}) e_{i_1 \ldots i_N}$$

$$x + y = \sum_{i_1,\ldots,i_N} \operatorname{tr}\left[\begin{pmatrix} A_1^{(i_1)} & 0 \\ 0 & B_1^{(i_1)} \end{pmatrix} \underbrace{\begin{pmatrix} A_2^{(i_2)} & 0 \\ 0 & B_2^{(i_2)} \end{pmatrix}}_{D' \times D'} \cdots \begin{pmatrix} A_N^{(i_N)} \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ B_N^{(i_N)} \end{pmatrix} \right] e_{i_1 \ldots i_N}$$

- Summing MPS vectors leads to an **augmentation** of the matrix sizes $D \to D'$.
- Use compression techniques to keep the size of the MPS matrices constant.

Computations with MPS

- **Inner products**

$$x^{\mathrm{H}} y = \sum_{i_1,\ldots,i_N} \overbrace{\left(\operatorname{tr}\left(\bar{A}_1^{(i_1)} \cdots \bar{A}_N^{(i_N)}\right)\right)}^{\bar{x}_{i_1,\ldots,i_N}} \cdot \overbrace{\left(\operatorname{tr}\left(B_1^{(i_1)} \cdots B_N^{(i_N)}\right)\right)}^{y_{i_1,\ldots,i_N}}$$

$$= \sum_{i_j} \sum_{k_j} \bar{a}^{(i_1)}_{1;k_1,k_2} \cdots \bar{a}^{(i_N)}_{N;k_N,k_1} \sum_{m_j} b^{(i_1)}_{1;m_1,m_2} \cdots b^{(i_N)}_{N;m_N,m_1}$$

Find an efficient ordering of the summations!

- **Matrix $\times$ vector**

$$Hx = \left(\sum_{k=1}^{M} \alpha_k Q_1^{(k)} \otimes \cdots \otimes Q_N^{(k)}\right) \left(\sum_{i_1,\ldots,i_N} \operatorname{tr}(A_1^{(i_1)} \ldots A_N^{(i_N)}) e_{i_1 \ldots i_N}\right)$$

$$= \sum_{k=1}^{M} y_{MPS}^{(k)}$$

$\Longrightarrow$ 4. ✓

Uniqueness of MPS

- The MPS formalism is not unique:

$$\sum_{i_1,\ldots,i_N} \operatorname{tr}\left(A_1^{(i_1)} \cdot A_2^{(i_2)} \cdots A_N^{(i_N)}\right) e_{i_1,\ldots,i_N}$$

$$= \sum_{i_1,\ldots,i_N} \operatorname{tr}\left((A_1^{(i_1)} M_2^{-1})(M_2 A_2^{(i_2)} M_3^{-1}) \cdots (M_N A_N^{(i_N)})\right) e_{i_1,\ldots,i_N}$$

Uniqueness of MPS

- The MPS formalism is not unique:

$$\sum_{i_1,\ldots,i_N} \operatorname{tr}\left(A_1^{(i_1)} \cdot A_2^{(i_2)} \cdots A_N^{(i_N)}\right) e_{i_1,\ldots,i_N}$$

$$= \sum_{i_1,\ldots,i_N} \operatorname{tr}\left((A_1^{(i_1)} M_2^{-1})(M_2 A_2^{(i_2)} M_3^{-1}) \cdots (M_N A_N^{(i_N)})\right) e_{i_1,\ldots,i_N}$$

- Replace the matrices $A_j^{(0)}, A_j^{(1)}$ by parts of unitary matrices using the SVD:

$$\begin{pmatrix} A_j^{(0)} \\ A_j^{(1)} \end{pmatrix} = \begin{pmatrix} U_j^{(0)} \\ U_j^{(1)} \end{pmatrix} \Sigma_j V_j \ .$$

Uniqueness of MPS

- The MPS formalism is not unique:

$$\sum_{i_1,\ldots,i_N} \operatorname{tr}\left(A_1^{(i_1)} \cdot A_2^{(i_2)} \cdots A_N^{(i_N)}\right) e_{i_1,\ldots,i_N}$$

$$= \sum_{i_1,\ldots,i_N} \operatorname{tr}\left((A_1^{(i_1)} M_2^{-1})(M_2 A_2^{(i_2)} M_3^{-1}) \cdots (M_N A_N^{(i_N)})\right) e_{i_1,\ldots,i_N}$$

- Replace the matrices $A_j^{(0)}, A_j^{(1)}$ by parts of unitary matrices using the SVD:

$$\begin{pmatrix} A_j^{(0)} \\ A_j^{(1)} \end{pmatrix} = \begin{pmatrix} U_j^{(0)} \\ U_j^{(1)} \end{pmatrix} \Sigma_j V_j \ .$$

- Multiply the $\Sigma_j V_j$ parts on the right neighbour.

Uniqueness of MPS

- The MPS formalism is not unique:

$$\sum_{i_1,\ldots,i_N} \operatorname{tr}\left(A_1^{(i_1)} \cdot A_2^{(i_2)} \cdots A_N^{(i_N)}\right) e_{i_1,\ldots,i_N}$$

$$= \sum_{i_1,\ldots,i_N} \operatorname{tr}\left((A_1^{(i_1)} M_2^{-1})(M_2 A_2^{(i_2)} M_3^{-1}) \cdots (M_N A_N^{(i_N)})\right) e_{i_1,\ldots,i_N}$$

- Replace the matrices $A_j^{(0)}, A_j^{(1)}$ by parts of unitary matrices using the SVD:

$$\begin{pmatrix} A_j^{(0)} \\ A_j^{(1)} \end{pmatrix} = \begin{pmatrix} U_j^{(0)} \\ U_j^{(1)} \end{pmatrix} \Sigma_j V_j \ .$$

- Multiply the $\Sigma_j V_j$ parts on the right neighbour.
- Replace all A matrices by U matrices, up to one.
- This remaining factor is considered to be optimized.

Uniqueness of MPS

- The MPS formalism is not unique:

$$\sum_{i_1,\ldots,i_N} \operatorname{tr}\left(A_1^{(i_1)} \cdot A_2^{(i_2)} \cdots A_N^{(i_N)}\right) e_{i_1,\ldots,i_N}$$

$$= \sum_{i_1,\ldots,i_N} \operatorname{tr}\left((A_1^{(i_1)} M_2^{-1})(M_2 A_2^{(i_2)} M_3^{-1}) \cdots (M_N A_N^{(i_N)})\right) e_{i_1,\ldots,i_N}$$

- Replace the matrices $A_j^{(0)}, A_j^{(1)}$ by parts of unitary matrices using the SVD:

$$\begin{pmatrix} A_j^{(0)} \\ A_j^{(1)} \end{pmatrix} = \begin{pmatrix} U_j^{(0)} \\ U_j^{(1)} \end{pmatrix} \Sigma_j V_j \ .$$

- Multiply the $\Sigma_j V_j$ parts on the right neighbour.
- Replace all A matrices by U matrices, up to one.
- This remaining factor is considered to be optimized.
- Leads to better convergence properties and numerical stability.

Computation of the Ground State

How to calculate the smallest eigenvalue of the Hamiltonian H?

1. Variational ansatz:

$$\min_{x_{MPS}} \frac{x_{MPS}^{\mathrm{H}} H x_{MPS}}{x_{MPS}^{\mathrm{H}} x_{MPS}}$$

- Find optimal matrix pairs $A_j^{(i_j)}$ for all sites j
- Use Alternating Least Squares: Keep all matrix pairs up to one fixed and optimize the remaining one

Computation of the Ground State

How to calculate the smallest eigenvalue of the Hamiltonian H?

1. Variational ansatz:

$$\min_{x_{MPS}} \frac{x_{MPS}^{\mathrm{H}} H x_{MPS}}{x_{MPS}^{\mathrm{H}} x_{MPS}}$$

   - Find optimal matrix pairs $A_j^{(i_j)}$ for all sites j
   - Use Alternating Least Squares: Keep all matrix pairs up to one fixed and optimize the remaining one

2. New idea: Use subspace iteration

$$\min_{x \in U_m} \frac{x^{\mathrm{H}} H x}{x^{\mathrm{H}} x}$$

   - Find optimal solution in the subspace U_m
   - U_m is defined by MPS vectors

Minimization Algorithm in terms of MPS

- Start with initial guesses for the $A_j^{(i_j)}$ matrices.
- Replace $A_j^{(i_j)}$ by $U_j^{(i_j)}$ up to $A_1^{(i_1)}$.
- Consider all U matrices as fixed and optimize $X_1 = A_1$:

$$\min_{X_1} \frac{\left(\sum_{i_j} \mathrm{tr}\left(X_1^{(i_1)} U_2^{(i_2)} \cdots U_N^{(i_N)}\right) e_i\right)^{\mathrm{H}} H \left(\sum_{i_j} \mathrm{tr}\left(X_1^{(i_1)} U_2^{(i_2)} \cdots U_N^{(i_N)}\right) e_i\right)}{\left(\sum_{i_j} \mathrm{tr}\left(X_1^{(i_1)} U_2^{(i_2)} \cdots U_N^{(i_N)}\right) e_i\right)^{\mathrm{H}} \left(\sum_{i_j} \mathrm{tr}\left(X_1^{(i_1)} U_2^{(i_2)} \cdots U_N^{(i_N)}\right) e_i\right)}$$

$$= \min_{X_1} \frac{{X_1}^{\mathrm{H}} R_1 X_1}{{X_1}^{\mathrm{H}} X_1}$$

with effective Hamiltonian R_1

- $\Longrightarrow$ dense eigenvalue problem for the $(2D^2 \times 2D^2)$ matrix R_1.
- Orthogonalize the solution and continue with the right neighbor.

Modifications of this algorithm lead to the DMRG method.

Subspace iteration for the smallest eigenvalue

- Idea: Minimize the Rayleigh quotient $\frac{x^{\mathrm{H}} H x}{x^{\mathrm{H}} x}$ over a subspace (**!**) $U_m := \mathrm{span}\{q_1, \ldots, q_m\}$.
- Let $V_m = (q_1, \ldots, q_m) \in \mathbb{C}^{2^N \times m}$

Subspace iteration for the smallest eigenvalue

- Idea: Minimize the Rayleigh quotient $\frac{x^{\mathrm{H}} H x}{x^{\mathrm{H}} x}$ over a subspace (!) $U_m := \operatorname{span}\{q_1, \ldots, q_m\}$.
- Let $V_m = (q_1, \ldots, q_m) \in \mathbb{C}^{2^N \times m}$
- The minimization over U_m yields

$$\begin{aligned}
\min_{x \in U_m \setminus \{0\}} \frac{x^{\mathrm{H}} H x}{x^{\mathrm{H}} x} \quad \overset{x = V_m y}{=} \quad & \min_{y \in \mathbb{C}^m \setminus \{0\}} \frac{(V_m y)^{\mathrm{H}} H (V_m y)}{(V_m y)^{\mathrm{H}} V_m y} \\
= \quad & \min_{y \in \mathbb{C}^m \setminus \{0\}} \frac{y^{\mathrm{H}} \left(V_m^{\mathrm{H}} H V_m\right) y}{y^{\mathrm{H}} \left(V_m^{\mathrm{H}} V_m\right) y} \\
= \quad & \min_{y \in \mathbb{C}^m \setminus \{0\}} \frac{y^{\mathrm{H}} A y}{y^{\mathrm{H}} B y} \,,
\end{aligned}$$

a generalized eigenvalue problem $Ay = \lambda_{\min} By$ of size $m \times m$.

Subspace iteration for the smallest eigenvalue

- Idea: Minimize the Rayleigh quotient $\frac{x^{\mathrm{H}} H x}{x^{\mathrm{H}} x}$ over a subspace (!) $U_m := \operatorname{span}\{q_1, \ldots, q_m\}$.
- Let $V_m = (q_1, \ldots, q_m) \in \mathbb{C}^{2^N \times m}$
- The minimization over U_m yields

$$\begin{aligned} \min_{x \in U_m \setminus \{0\}} \frac{x^{\mathrm{H}} H x}{x^{\mathrm{H}} x} \quad & \overset{x = V_m y}{=} \quad \min_{y \in \mathbb{C}^m \setminus \{0\}} \frac{(V_m y)^{\mathrm{H}} H (V_m y)}{(V_m y)^{\mathrm{H}} V_m y} \\ & = \quad \min_{y \in \mathbb{C}^m \setminus \{0\}} \frac{y^{\mathrm{H}} \left(V_m^{\mathrm{H}} H V_m\right) y}{y^{\mathrm{H}} \left(V_m^{\mathrm{H}} V_m\right) y} \\ & = \quad \min_{y \in \mathbb{C}^m \setminus \{0\}} \frac{y^{\mathrm{H}} A y}{y^{\mathrm{H}} B y}, \end{aligned}$$

  a generalized eigenvalue problem $Ay = \lambda_{\min} By$ of size $m \times m$.
- $x = V_m y$ or restart with $q_1 = V_m y$

Subspace iteration for the smallest eigenvalue

- Idea: Minimize the Rayleigh quotient $\frac{x^{\mathrm{H}} H x}{x^{\mathrm{H}} x}$ over a subspace (!) $U_m := \operatorname{span}\{q_1, \ldots, q_m\}$.
- Let $V_m = (q_1, \ldots, q_m) \in \mathbb{C}^{2^N \times m}$
- The minimization over U_m yields

$$\begin{aligned} \min_{x \in U_m \setminus \{0\}} \frac{x^{\mathrm{H}} H x}{x^{\mathrm{H}} x} \quad & \overset{x = V_m y}{=} \quad \min_{y \in \mathbb{C}^m \setminus \{0\}} \frac{(V_m y)^{\mathrm{H}} H (V_m y)}{(V_m y)^{\mathrm{H}} V_m y} \\ & = \quad \min_{y \in \mathbb{C}^m \setminus \{0\}} \frac{y^{\mathrm{H}} \left(V_m^{\mathrm{H}} H V_m\right) y}{y^{\mathrm{H}} \left(V_m^{\mathrm{H}} V_m\right) y} \\ & = \quad \min_{y \in \mathbb{C}^m \setminus \{0\}} \frac{y^{\mathrm{H}} A y}{y^{\mathrm{H}} B y}, \end{aligned}$$

  a generalized eigenvalue problem $Ay = \lambda_{\min} By$ of size $m \times m$.
- $x = V_m y$ or restart with $q_1 = V_m y$
- How to choose an appropriate subspace U_m?

Subspace iteration for the smallest eigenvalue

- Choose the Krylov space $\mathcal{K}_m(H, b)$ as subspace:

$$U_m = \mathcal{K}_m(H, b) := \operatorname{span}\{b, Hb, \ldots, H^{m-1}b\}$$

Subspace iteration for the smallest eigenvalue

- Choose the Krylov space $\mathcal{K}_m(H, b)$ as subspace:

$$U_m = \mathcal{K}_m(H, b) := \operatorname{span}\{b, Hb, \ldots, H^{m-1}b\}$$

- This ansatz yields

$$A = V_m^{\mathrm{H}} H V_m = \begin{pmatrix} b^{\mathrm{H}}Hb & b^{\mathrm{H}}H^2b & \ldots & b^{\mathrm{H}}H^mb \\ b^{\mathrm{H}}H^2b & b^{\mathrm{H}}H^3b & \ldots & b^{\mathrm{H}}H^{m+1}b \\ \vdots & \vdots & \ddots & \vdots \\ b^{\mathrm{H}}H^mb & b^{\mathrm{H}}H^{m+1}b & \ldots & b^{\mathrm{H}}H^{2m-1}b \end{pmatrix}$$

$$B = V_m^{\mathrm{H}} V_m = \begin{pmatrix} b^{\mathrm{H}}b & b^{\mathrm{H}}Hb & \ldots & b^{\mathrm{H}}H^{m-1}b \\ b^{\mathrm{H}}Hb & b^{\mathrm{H}}H^3b & \ldots & b^{\mathrm{H}}H^mb \\ \vdots & \vdots & \ddots & \vdots \\ b^{\mathrm{H}}H^{m-1}b & b^{\mathrm{H}}H^mb & \ldots & b^{\mathrm{H}}H^{2m-2}b \end{pmatrix}$$

Subspace iteration in terms of MPS

- $b = b_{MPS}$ and
  $\mathcal{K}_m(H, b_{MPS}) = \text{span}\{b_{MPS}, Hb_{MPS}, \ldots, H^{m-1}b_{MPS}\}$:

Subspace iteration in terms of MPS

- $b = b_{MPS}$ and
  $\mathcal{K}_m(H, b_{MPS}) = \mathrm{span}\{b_{MPS}, Hb_{MPS}, \ldots, H^{m-1}b_{MPS}\}$:
- In principle the same proceeding, **BUT**

Subspace iteration in terms of MPS

- $b = b_{MPS}$ and
  $\mathcal{K}_m(H, b_{MPS}) = \text{span}\{b_{MPS}, Hb_{MPS}, \ldots, H^{m-1}b_{MPS}\}$:
- In principle the same proceeding, **BUT**
  - Computation of the matrix vector products:

$$H^j b_{MPS} = \left(\sum_{k=1}^{M} \alpha_k Q_1^{(k)} \otimes \cdots \otimes Q_N^{(k)} \right)^j b_{MPS}$$

$$= \sum_{k_1=1}^{M} \cdots \sum_{k_j=1}^{M} b_{MPS}^{(k_1,\ldots,k_j)}$$

Number of summands grows dramatically!

Subspace iteration in terms of MPS

- $b = b_{MPS}$ and
  $\mathcal{K}_m(H, b_{MPS}) = \mathrm{span}\{b_{MPS}, Hb_{MPS}, \ldots, H^{m-1}b_{MPS}\}$:
- In principle the same proceeding, **BUT**
  - Computation of the matrix vector products:

$$H^j b_{MPS} = \left(\sum_{k=1}^{M} \alpha_k Q_1^{(k)} \otimes \cdots \otimes Q_N^{(k)} \right)^j b_{MPS}$$
$$= \sum_{k_1=1}^{M} \cdots \sum_{k_j=1}^{M} b_{MPS}^{(k_1,\ldots,k_j)}$$

    Number of summands grows dramatically!
  - Application of the back-transformation:

$$V_m y_{\min} = \sum_{j=1}^{m} y_{\min,j} H^{j-1} b_{MPS}$$

    does **not** lie in the MPS space.

Subspace iteration in terms of MPS

- Solution: Application of a projection $\mathcal{P}$ onto the MPS space:

$$\mathcal{P} : \sum_{k=1}^{M} y_{MPS}^{(k)} \mapsto \text{argmin}_{x_{MPS}} \left\| \sum_{k=1}^{M} y_{MPS}^{(k)} - x_{MPS} \right\|$$

Subspace iteration in terms of MPS

- Solution: Application of a projection $\mathcal{P}$ onto the MPS space:

$$\mathcal{P} : \sum_{k=1}^{M} y_{MPS}^{(k)} \mapsto \text{argmin}_{x_{MPS}} \left\| \sum_{k=1}^{M} y_{MPS}^{(k)} - x_{MPS} \right\|$$

- Solve the optimization problem

$$\min_{A} \left\| \sum_{i_1,\ldots,i_N} \text{tr} \left(\sum_{k=1}^{M} (B_1^{(i_1;k)} \cdots B_N^{(i_N;k)}) - A_1^{(i_1)} \cdots A_N^{(i_N)} \right) e_{i_1 \ldots i_N} \right\|$$

Subspace iteration in terms of MPS

- Solution: Application of a projection $\mathcal{P}$ onto the MPS space:

$$\mathcal{P} : \sum_{k=1}^{M} y_{MPS}^{(k)} \mapsto \text{argmin}_{x_{MPS}} \left\| \sum_{k=1}^{M} y_{MPS}^{(k)} - x_{MPS} \right\|$$

- Solve the optimization problem

$$\min_{A} \left\| \sum_{i_1,\ldots,i_N} \text{tr} \left(\sum_{k=1}^{M} (B_1^{(i_1;k)} \cdots B_N^{(i_N;k)}) - A_1^{(i_1)} \cdots A_N^{(i_N)} \right) e_{i_1 \ldots i_N} \right\|$$

- Two possible projection methods:
  - SVD-based truncation
  - ALS-based optimization

Subspace iteration in terms of MPS

- Solution: Application of a projection $\mathcal{P}$ onto the MPS space:

$$\mathcal{P} : \sum_{k=1}^{M} y_{MPS}^{(k)} \mapsto \text{argmin}_{x_{MPS}} \left\| \sum_{k=1}^{M} y_{MPS}^{(k)} - x_{MPS} \right\|$$

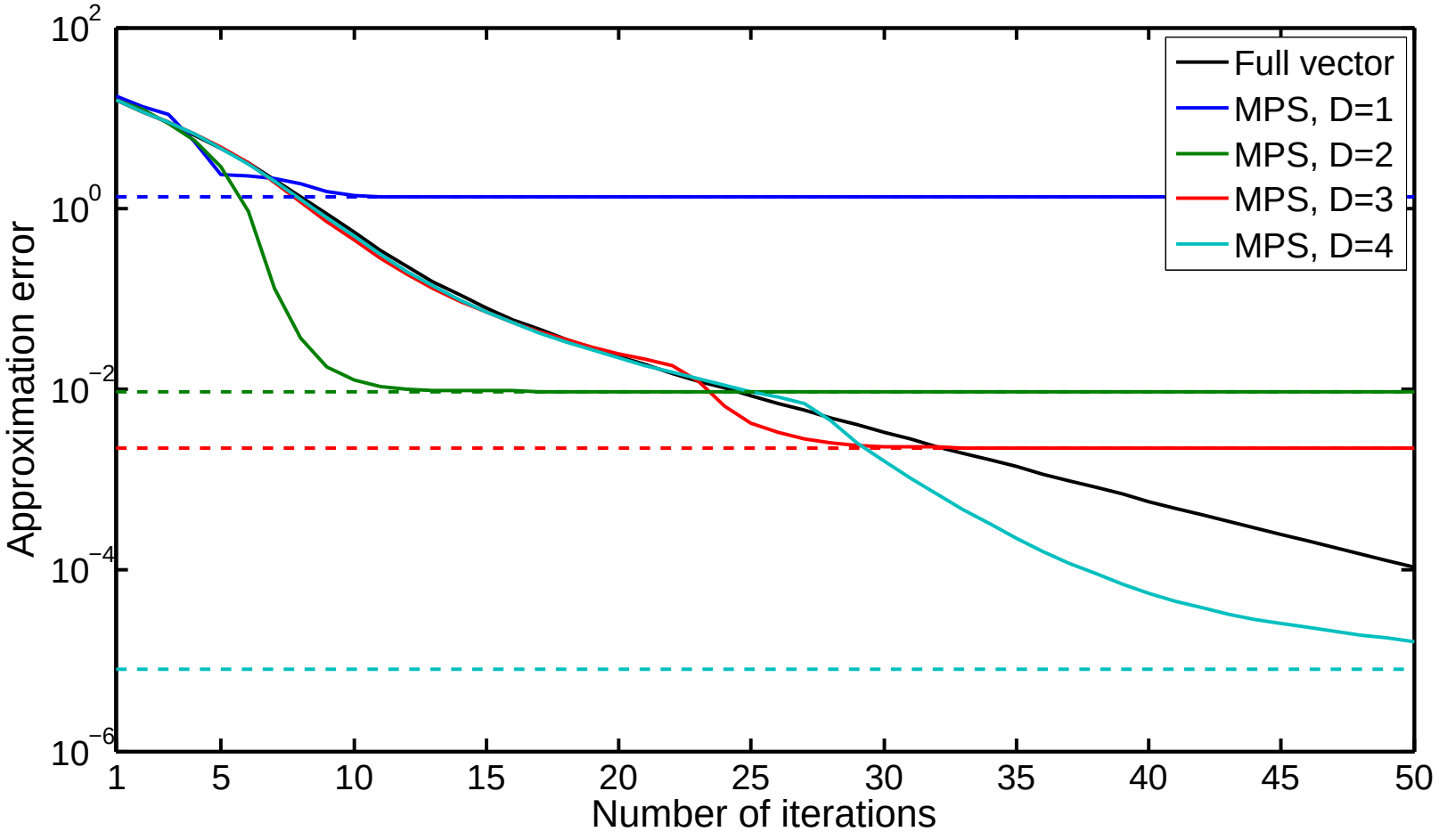
- Solve the optimization problem

$$\min_{A} \left\| \sum_{i_1,\ldots,i_N} \text{tr} \left(\sum_{k=1}^{M} (B_1^{(i_1;k)} \cdots B_N^{(i_N;k)}) - A_1^{(i_1)} \cdots A_N^{(i_N)} \right) e_{i_1 \ldots i_N} \right\|$$

- Two possible projection methods:
  - SVD-based truncation
  - ALS-based optimization
- Use ALS to find optimal matrix pair $A_r^{(0)}, A_r^{(1)}$
- Forcing the derivative to be zero leads to a linear system.

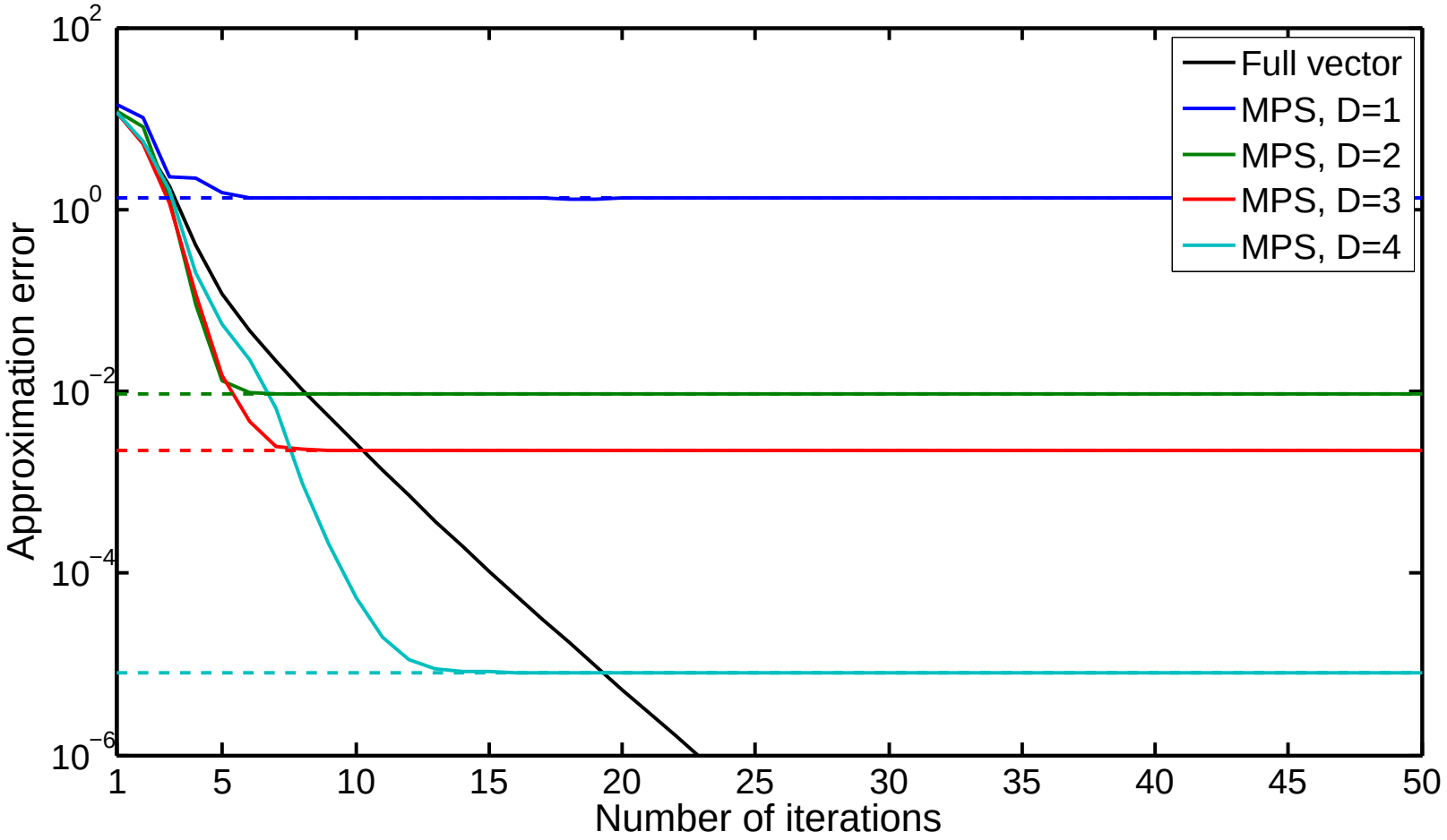
Numerical results

Ising-type Hamiltonian, $N = 10$, $dim = 2$



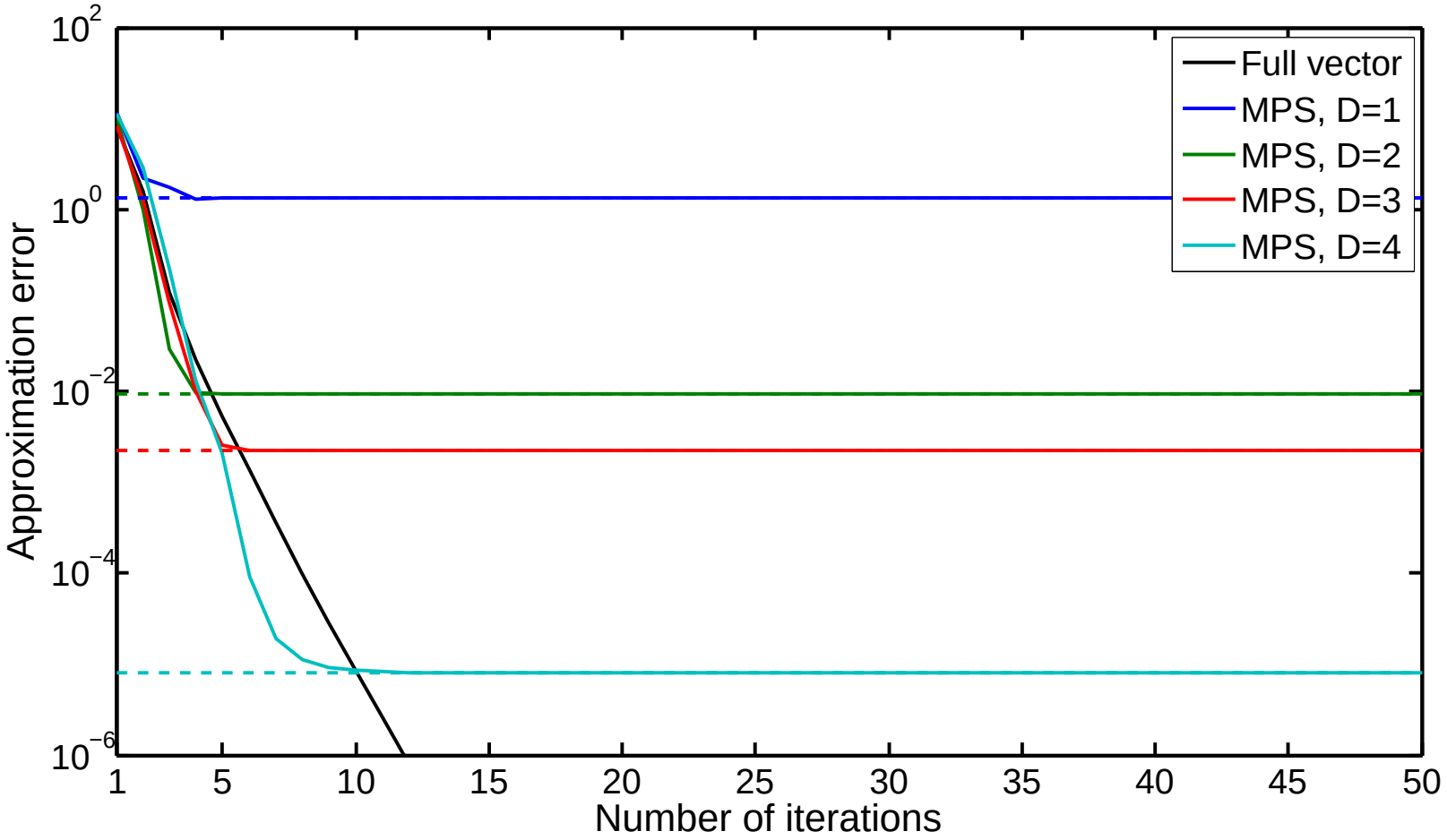
Numerical results

Ising-type Hamiltonian, $N = 10$, $dim = 3$



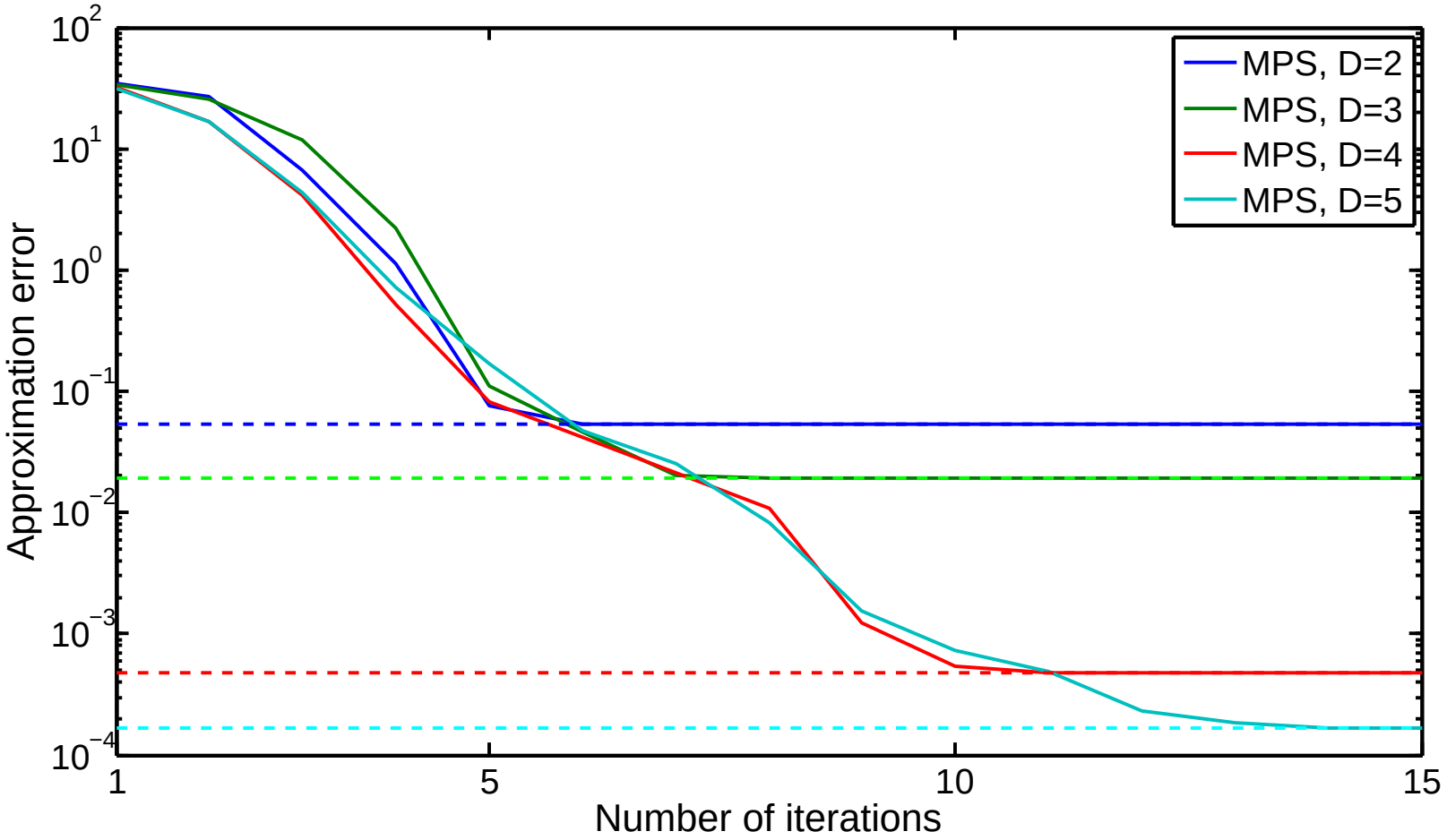
Numerical results

Ising-type Hamiltonian, $N = 10$, $dim = 4$



Numerical results

Ising-type Hamiltonian, $N = 25$, $dim = 5$



Conclusions

- Matrix Product States
  - data-sparse representation format
  - represent ground states faithfully
  - DMRG in terms of MPS

Conclusions

- Matrix Product States
  - data-sparse representation format
  - represent ground states faithfully
  - DMRG in terms of MPS
- Subspace methods in terms of MPS
  - more flexibility (choice of subspace dimension)
  - approximation not only of the lowest eigenvalue
  - projection required to keep the number of summands limited
  - projection can even improve the approximation
  - convergence to DMRG solution
  - convergence properties of subspace iteration and approximation properties of MPS

Conclusions

- Matrix Product States
  - data-sparse representation format
  - represent ground states faithfully
  - DMRG in terms of MPS
- Subspace methods in terms of MPS
  - more flexibility (choice of subspace dimension)
  - approximation not only of the lowest eigenvalue
  - projection required to keep the number of summands limited
  - projection can even improve the approximation
  - convergence to DMRG solution
  - convergence properties of subspace iteration and approximation properties of MPS

Conclusions

- Matrix Product States
  - data-sparse representation format
  - represent ground states faithfully
  - DMRG in terms of MPS
- Subspace methods in terms of MPS
  - more flexibility (choice of subspace dimension)
  - approximation not only of the lowest eigenvalue
  - projection required to keep the number of summands limited
  - projection can even improve the approximation
  - convergence to DMRG solution
  - convergence properties of subspace iteration and approximation properties of MPS

Thank you for your attention.