Comprehensive Notes: Matrix Product States (MPS)

Annwoy Roy Choudhury

BS-MS Student in Physics (2022-2027)
Indian Institute of Science Education and Research (IISER), Mohali, Punjab, India

Document created during a research internship (May-July 2025)
at the Center for Optical Quantum Technologies,
University of Hamburg, Hamburg, Germany

Contents

1 The Basics: What is an MPS? 4
1.1 Gauge Freedom: The Source of Ambiguity . . . 4

2 Generalised Tensor Decompositions by Mode 4
2.1 Generalised QR Decomposition . . . 5
2.1.1 The Mode . . . 5
2.1.2 The Goal . . . 5
2.1.3 The Mathematics . . . 5
2.1.4 The Algorithm . . . 5
2.1.5 Physical Significance . . . 6
2.1.6 Testing and Verification . . . 6
2.1.7 Diagrams . . . 7
2.2 Generalised SVD . . . 7
2.2.1 The Goal . . . 7
2.2.2 The Mathematics . . . 7
2.2.3 The Algorithm . . . 8
2.2.4 Physical Significance . . . 8
2.2.5 Testing and Verification . . . 9
2.2.6 Diagrams . . . 9

3 The Building Blocks I: Tensor Contraction 9
3.1 Optimal Contraction Order and Cost . . . 9
3.2 The Zipper Algorithm: Cost Analysis . . . 10

4 The Building Blocks II: Tensor Decomposition 10
4.1 Singular Value Decomposition (SVD) . . . 10
4.2 QR and RQ Decompositions . . . 11

5 The Goal of Canonicalization: Revealing the Schmidt Decomposition 11

6 Left-Canonical Form 11

7 Right-Canonical Form 12

8 Mixed-Canonical Form and the Orthogonality Center 12

9 Rank, Entanglement, and the Structure of States 13

10 Bond Dimension and Entanglement 14

11 Constructing an MPS from a Quantum State 14

12 Why Bond Dimensions Change During Canonicalization 15
12.1 The Physics: Bond Dimension as an Information Channel . . . 15
12.2 The Mathematics: How SVD-based Canonicalization Discovers the Profile . . . 16
12.2.1 Step 1: Canonicalize Site 0 . . . 16
12.2.2 Step 2: Canonicalize Site 1 . . . 16
12.3 Conclusion . . . 17

13 The Full Canonicalization Workflow: From Random MPS to Vidal Gauge **17**
13.1 Step 1: Left-Canonicalization 17
13.2 Step 2: Right-Canonicalization 18
13.3 Step 3: Mixed-Canonicalization 18
13.4 Step 4: Transformation to Vidal Gauge (Γ-Λ Form) 19

14 SVD Compression: The Fast, Greedy Approach **20**
14.1 The Core Idea: Entanglement and the Schmidt Decomposition 20
14.2 The Algorithm: A Stable Two-Sweep Procedure 20
14.3 SVD for Optimal Truncation and Cost 21
14.4 Summary: Advantages and Disadvantages 22

15 Variational Compression: The Optimal Approach **22**
15.1 The Core Idea: Direct Minimization of Global Error 22
15.2 The Algorithm: One-Site Optimization 22
15.3 Comparison with Schollwöck's Formalism 22
15.4 The Full Sweeping Algorithm 23
15.5 Summary: Advantages and Disadvantages 24

16 Compression Conclusion: Which Method to Use? **24**

17 The Cost of MPS Algorithms and the Role of D **25**

18 QR vs. SVD for Canonicalization **25**

19 Matrix Product Operators (MPOs) **25**

20 Critical Systems and Finite Entanglement Scaling **26**

21 Graphical Notation: The Rules of the Game **27**

1 The Basics: What is an MPS?

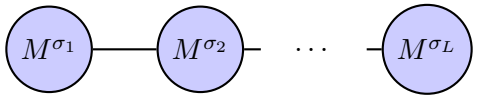
A general quantum state for a system of L sites is written as:

$$|\psi\rangle = \sum_{\sigma_1,\ldots,\sigma_L} c_{\sigma_1\ldots\sigma_L} |\sigma_1 \ldots \sigma_L\rangle \tag{1}$$

The number of coefficients c grows exponentially with L. An MPS is a way to decompose these coefficients into a product of small matrices, one for each site.

$$c_{\sigma_1\ldots\sigma_L} = M^{\sigma_1}_{a_0 a_1} M^{\sigma_2}_{a_1 a_2} \cdots M^{\sigma_L}_{a_{L-1} a_L} \tag{2}$$

where the internal "virtual" indices are contracted (summed over). This structure is often visualized as a chain:



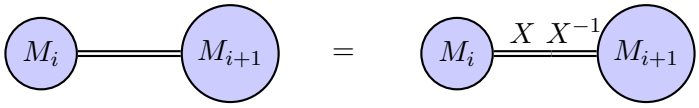
- **Physical Indices (σ_i):** Represent the state at each site (e.g., spin-up/down). Their dimension is d.
- **Virtual/Bond Indices (a_i):** Internal indices that are contracted. They do not correspond to any physical particle but carry the entanglement information. Their dimension D is the **bond dimension**.

1.1 Gauge Freedom: The Source of Ambiguity

A "random" MPS, where the M matrices have no special properties, is not unique. For any invertible matrix X, we can insert the identity XX^{-1} on a bond between sites i and $i+1$ without changing the final state $|\psi\rangle$.

$$\cdots M^{\sigma_i} M^{\sigma_{i+1}} \cdots = \cdots M^{\sigma_i} (XX^{-1}) M^{\sigma_{i+1}} \cdots = \cdots (M^{\sigma_i} X)(X^{-1} M^{\sigma_{i+1}}) \cdots \tag{3}$$

This freedom to transform the tensors, $\tilde{M}^{\sigma_i} = M^{\sigma_i} X$ and $\tilde{M}^{\sigma_{i+1}} = X^{-1} M^{\sigma_{i+1}}$, is known as **gauge freedom**. This ambiguity makes a random MPS difficult to work with.



Q: What are we canonicalizing in a random MPS?

A: We are not changing the physical state $|\psi\rangle$. We are using the gauge freedom to change the individual matrices M into a special, standard form (A or B matrices) with useful orthogonality properties. This makes physical properties like the norm and entanglement easy to calculate.

2 Generalised Tensor Decompositions by Mode

A tensor can be decomposed into simpler, structured components by reshaping it into a matrix and applying standard linear algebra factorizations. While this is often done by partitioning a tensor's indices into two contiguous groups, a more powerful and flexible approach is to isolate a single, specific index (or "mode") and decompose the tensor with respect to it. This section details the mathematics, algorithm, and physical significance of performing QR and Singular Value Decompositions (SVD) on an arbitrary tensor by selecting a single mode to act as the column space of the effective matrix.

2.1 Generalised QR Decomposition

2.1.1 The Mode

For a given tensor of rank-N, say $A_{i_1 i_2 \ldots i_N}$, the `mode` is an integer $m \in [0, N-1]$ that specifies which index we wish to isolate. For instance, in a rank-4 tensor A_{ijkl}, if we choose `mode`=1, we are isolating the index j.

2.1.2 The Goal

The goal is to decompose the tensor A into two parts:

1. An isometry Q. This tensor retains all original indices except for the chosen `mode`, which is replaced by a new internal "bond" index, x.
2. A smaller matrix R (often called the remainder or triangular part) which connects the new bond index x to the original `mode` index.

The tensor Q will satisfy an isometric property: contracting it with its conjugate over all of its original indices yields an identity matrix on the new bond indices.

2.1.3 The Mathematics

For a rank-4 tensor A_{ijkl} and a chosen `mode` of j (axis 1), the decomposition takes the form of a tensor contraction:

$$A_{ijkl} = \sum_x Q_{ixkl} R_{xj} \tag{4}$$

where x is the new internal bond index. The dimensions of the tensors are:

- A: (d_i, d_j, d_k, d_l)
- Q: (d_i, d_x, d_k, d_l)
- R: (d_x, d_j)

The defining property of the isometry Q is given by:

$$\sum_{i,k,l} Q_{ixkl} Q^*_{iykl} = \delta_{xy} \tag{5}$$

where Q^* is the complex conjugate of Q.

2.1.4 The Algorithm

The algorithm achieves this decomposition without ambiguity by a series of controlled permutations.

1. **Permute to Isolate:** The `mode` axis is moved to the final position by a sequence of one-step swaps with its right neighbor. The sequence of swaps performed is recorded.
   - Example for A_{ijkl} and `mode`=j:
     $A_{ijkl} \xrightarrow{\text{swap } j \leftrightarrow k} A_{ikjl} \xrightarrow{\text{swap } j \leftrightarrow l} A_{iklj}$
   - The recorded swaps are: $[(1,2),(2,3)]$.
2. **Matricize:** The resulting permuted tensor (A'_{iklj}) is reshaped (matricized) into a 2D matrix. All axes except the last (the isolated `mode` axis) are flattened to form the rows.

$$A'_{iklj} \quad \longrightarrow \quad M_{(ikl)j}$$

3. **Decompose:** A standard matrix QR decomposition is performed on M.

$$M_{(ikl)j} = \sum_x (Q_{mat})_{(ikl)x} (R_{mat})_{xj}$$

The matrix R_{mat} is the final R_{xj} tensor and is complete.

4. **Reshape Isometry:** The matrix Q_{mat} is reshaped back into a tensor, Q', whose shape corresponds to the flattened row indices plus the new bond index.

$$(Q_{mat})_{(ikl)x} \quad \longrightarrow \quad Q'_{iklx}$$

5. **Inverse Permute:** The recorded sequence of swaps is applied in *reverse order* to the new tensor Q'. This moves the new bond index x from the last position back into the original position of the `mode` index.

   - Example: Applying reverse swaps $[(2,3),(1,2)]$ to Q'_{iklx}:
     $Q'_{iklx} \xrightarrow{\text{swap axis } 2\leftrightarrow 3} Q'_{ikxl} \xrightarrow{\text{swap axis } 1\leftrightarrow 2} Q'_{ixkl}$

   The final tensor is the desired isometry Q_{ixkl}.

2.1.5 Physical Significance

This decomposition is a fundamental tool in tensor network theory.

- **Gauge Fixing:** It allows one to impose a specific "gauge" on a tensor, making one of its legs orthonormal with respect to the others. This is the core operation for bringing tensors in an MPO or PEPS into a canonical form.
- **Environment Calculation:** In algorithms like DMRG, this is used to efficiently calculate and update the environment of a local site. The isometry property ensures that large parts of the network contract to identity, simplifying calculations.
- **Projectors:** The tensor Q can be viewed as a projector from the original Hilbert space of the `mode` index to the smaller subspace spanned by the new bond index.

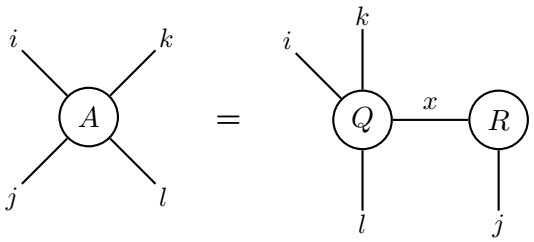
2.1.6 Testing and Verification

The correctness of the algorithm is verified through three key tests:

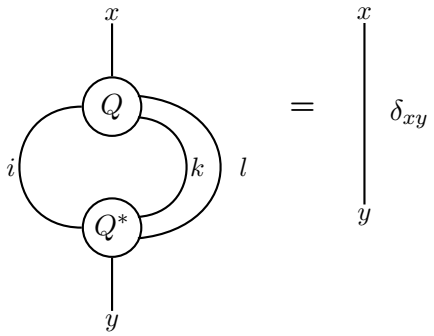
1. **Isometry Check:** The primary property of the output tensor Q must be verified. The contraction of Q with its conjugate Q^* over all original indices must yield an identity matrix on the bond indices, as shown in the diagram below.
2. **Bond Position Check:** The shape of the output tensor Q must be correct. Its rank must be the same as the input tensor A. The dimension of the axis corresponding to the original `mode` must be equal to the new bond dimension. All other axes must have their original dimensions.
3. **Reconstruction Check:** Contracting the outputs Q and R along the new bond index must perfectly reconstruct the original tensor A.

2.1.7 Diagrams

The decomposition of a rank-4 tensor A_{ijkl} with respect to mode j can be visualized as splitting the tensor into an isometry Q and a remainder matrix R. The index j is separated, while all other indices (i, k, l) remain with the main tensor body, which becomes Q. A new internal index x is created to connect them.



The isometry condition for the resulting tensor Q_{ixkl} states that if we contract Q with its complex conjugate Q^* over all of their shared original indices (i, k, l), the result is an identity operator on the new bond index x. This means the network effectively "vanishes" into a simple line representing the Kronecker delta.



2.2 Generalised SVD

2.2.1 The Goal

The goal of the generalised SVD is similar to QR but provides a more detailed decomposition. It factorizes a tensor A into:

1. An isometry U, which is identical in structure to the Q from the QR decomposition.
2. A diagonal matrix of non-negative singular values S.
3. A unitary matrix $V^\dagger$.

The pair $(S \cdot V^\dagger)$ collectively plays the role of the R matrix from the QR decomposition.

2.2.2 The Mathematics

For a rank-4 tensor A_{ijkl} and a chosen `mode` of j (axis 1), the SVD takes the form:

$$A_{ijkl} = \sum_{x,y} U_{ixkl} S_{xy} V^\dagger_{yj} \tag{6}$$

Since S is a diagonal matrix ($S_{xy} = s_x \delta_{xy}$), this is more commonly written as:

$$A_{ijkl} = \sum_x U_{ixkl} (S \cdot V^\dagger)_{xj} \tag{7}$$

The dimensions of the components are:

- A: (d_i, d_j, d_k, d_l)
- U: (d_i, d_x, d_k, d_l)
- S: (d_x, d_x)
- $V^\dagger$: (d_x, d_j)

The tensor U has the same isometry property as Q from the QR decomposition.

2.2.3 The Algorithm

The algorithm is identical to the Generalised QR algorithm, except for the decomposition step.

1. **Permute to Isolate:** Identical to the QR algorithm. The `mode` axis is moved to the end via sequential swaps, and the swaps are recorded.
2. **Matricize:** Identical. The permuted tensor is reshaped into a matrix M.
3. **Decompose:** A standard matrix SVD is performed on M.

   $$M_{(ikl)j} = \sum_x (U_{mat})_{(ikl)x} s_x (V^\dagger_{mat})_{xj}$$

   The matrices S (from the singular values s_x) and $V^\dagger_{mat}$ are the final S and $V^\dagger$ and are complete.
4. **Reshape Isometry:** The matrix U_{mat} is reshaped back into a tensor, U', with the new bond index at the end.

   $$(U_{mat})_{(ikl)x} \quad \longrightarrow \quad U'_{iklx}$$
5. **Inverse Permute:** The recorded sequence of swaps is applied in reverse order to U', moving the new bond index x into the original position of the `mode`. This produces the final isometry U_{ixkl}.

2.2.4 Physical Significance

The generalised SVD is arguably the most important tool in tensor network analysis.

- **Quantifying Correlation:** The singular values in the matrix S directly measure the amount of quantum entanglement (for pure states) or classical correlation between the degrees of freedom of the `mode` index and all other indices combined. A spectrum of rapidly decaying singular values indicates weak correlation.
- **Optimal Compression:** SVD provides the mathematically optimal way to compress a tensor. By keeping only the k largest singular values and discarding the rest, one obtains the best possible rank-k approximation of the original tensor (Eckart-Young theorem). This is the foundation of all MPS compression and truncation algorithms (like TEBD and DMRG).
- **Finding Canonical Forms:** The SVD is the central operation used to construct the Vidal canonical form of an MPS, where the singular values are made explicit on the bonds.

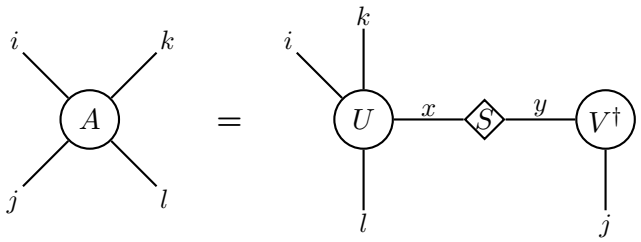
2.2.5 Testing and Verification

Verification is analogous to the QR case.

1. **Isometry Check:** The tensor U must satisfy the same isometry condition as Q: contracting U and U^* over their original indices must yield an identity matrix.
2. **Bond Position Check:** The shape of the output tensor U must have the new bond dimension at the correct `mode` axis, with all other dimensions matching the original tensor.
3. **Reconstruction Check:** Contracting U, S, and $V^\dagger$ must perfectly reconstruct the original tensor A.

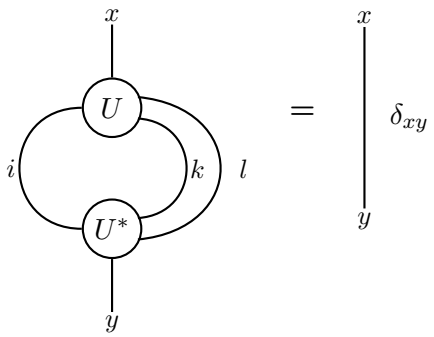
2.2.6 Diagrams

The SVD provides a more detailed factorization. The tensor A is split into an isometry U, a diagonal matrix of singular values S, and a unitary matrix $V^\dagger$. The pair $(S, V^\dagger)$ serves the same role as the R matrix in the QR decomposition.



Note: Since S is diagonal ($S_{xy} = s_x \delta_{xy}$), the indices x and y on its legs are the same. It is often drawn as a diamond or simply as a label on the bond.

The isometry condition for the tensor U_{ixkl} is identical to the one for Q in the QR decomposition. Contracting U with its conjugate U^* over the shared original indices results in an identity operator on the bond.



3 The Building Blocks I: Tensor Contraction

All MPS algorithms rely on contracting networks of tensors. The efficiency of this process depends critically on the order of contractions.

3.1 Optimal Contraction Order and Cost

A naive contraction by summing over all internal indices at once is computationally expensive. It is almost always more efficient to break the process down into a sequence of pairwise contractions.

Consider contracting four matrices to get a scalar: $F = \sum_{i,j,k} A_{ij} B_{jk} C_{ki}$. Assume all indices have dimension D.

- **Bad Order (Naive):** Summing over all indices simultaneously. The cost is the number of terms in the sum times the number of multiplications per term.

$$\text{Cost} = (\text{terms } D^3) \times (\text{mults } 2) = \mathcal{O}(D^3)$$

- **Good Order (Pairwise):** Contract pairwise, e.g., $E_{ik} = \sum_j A_{ij}B_{jk}$, then $F = \sum_{i,k} E_{ik}C_{ki}$.

$$\begin{aligned}\text{Cost(E)} &= (\text{output } D^2) \times (\text{sum } D) = \mathcal{O}(D^3)\\ \text{Cost(F)} &= (\text{output } 1) \times (\text{sum } D^2) = \mathcal{O}(D^2)\\ \text{Total Cost} &= \mathcal{O}(D^3) + \mathcal{O}(D^2) = \mathcal{O}(D^3)\end{aligned}$$

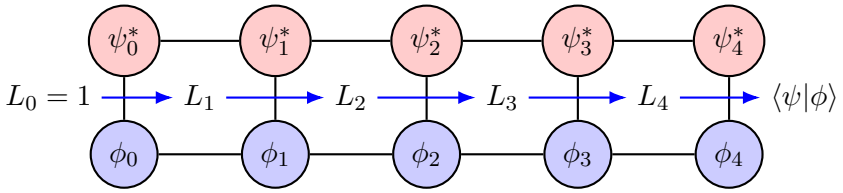
While the leading cost is the same here, for longer chains the pairwise method is vastly superior.

3.2 The Zipper Algorithm: Cost Analysis

The most common contraction is the inner product $\langle\psi|\phi\rangle$. The "zipper" algorithm contracts the network from one end to the other with a cost that scales **linearly** with the system size L.

The key operation is updating the environment tensor: $L_{i+1} = \text{contract}(L_i, \psi_i^*, \phi_i)$.

- L_i has shape (D, D).
- ψ_i^* has shape (D, d, D).
- ϕ_i has shape (D, d, D).
- The contraction involves summing over one physical index (dim d) and two virtual indices (dim D).
- The cost of one update step is $\mathcal{O}(D^3 d)$.
- Since we repeat this L times, the total cost is $\mathcal{O}(LD^3 d)$. This is a polynomial cost, a massive improvement over the exponential cost of contracting the full state vectors.



The left environment L_i is updated at each step by contracting it with the next pair of tensors: $L_{i+1} = \text{contract}(L_i, \psi_i^*, \phi_i)$.

4 The Building Blocks II: Tensor Decomposition

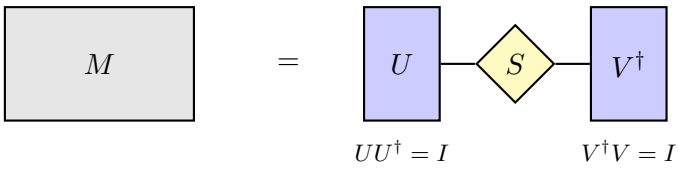
Decomposition is the process of splitting a single, large tensor into a product of smaller ones. This is fundamental to creating and manipulating MPS.

4.1 Singular Value Decomposition (SVD)

SVD is the most important decomposition because it is "rank-revealing." It tells us about the internal structure of a tensor. For any matrix M, SVD gives:

$$M = USV^\dagger \tag{8}$$

where U and V are isometric (or unitary) matrices, and S is a diagonal matrix of non-negative real numbers called the **singular values**. These singular values are the Schmidt coefficients (s_α) for the cut represented by the matrix partition.



4.2 QR and RQ Decompositions

When we only need to create an orthonormal tensor and do not need the singular values (e.g., during a canonicalization sweep without truncation), we can use the computationally faster QR or RQ decompositions.

- **QR Decomposition**: Splits $M = QR$, where Q is isometric and R is upper-triangular. Used for left-to-right sweeps.
- **RQ Decomposition**: Splits $M = RQ$, where Q is isometric and R is lower-triangular. Used for right-to-left sweeps.

The cost for both QR and RQ also scales as $\mathcal{O}(mn^2)$, but with a smaller prefactor than SVD, making them numerically faster.

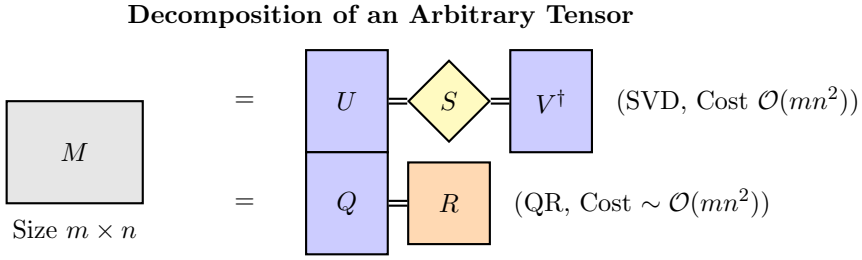


Figure 1: Visual representation of SVD and QR decompositions. A tensor M is reshaped into a matrix and then decomposed. SVD yields two isometries $(U, V^\dagger)$ and a diagonal matrix (S), while QR yields one isometry (Q) and a triangular matrix (R).

5 The Goal of Canonicalization: Revealing the Schmidt Decomposition

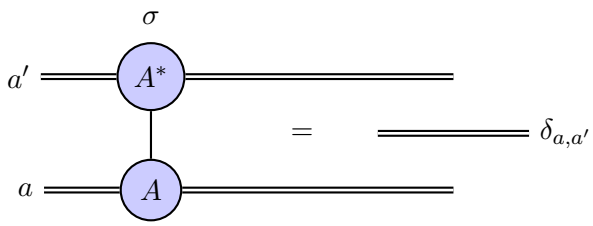
The core idea is to make the MPS explicitly represent the Schmidt decomposition across any cut. For a cut between a left part (A) and a right part (B), any state can be written as:

$$|\psi\rangle = \sum_\alpha s_\alpha \, |\phi_\alpha\rangle_A \, |\chi_\alpha\rangle_B \tag{9}$$

where $\{|\phi_\alpha\rangle_A\}$ and $\{|\chi_\alpha\rangle_B\}$ are two sets of orthonormal basis states. Canonicalization forces the MPS tensors to build these orthonormal bases.

6 Left-Canonical Form

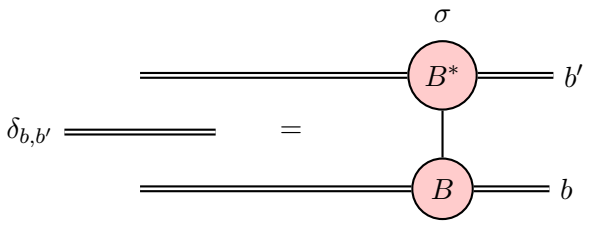
- **Goal:** To build an orthonormal basis from left to right.
- **Property:** A set of matrices $\{A^{\sigma_i}\}$ is left-normalized if $\sum_{\sigma_i} (A^{\sigma_i})^\dagger A^{\sigma_i} = I$ (Identity matrix).



- **Intuition:** The tensor $A^{(i)}$ acts as an isometry that takes the orthonormal basis of the block $(1 \dots i-1)$, adds site i, and produces a new orthonormal basis for the block $(1 \dots i)$.
- **Procedure:** A left-to-right sweep using QR or SVD decompositions at each site, absorbing the leftover R or $SV^\dagger$ part into the tensor to the right before decomposing it.
- **Resulting MPS:** $A^{(1)} - A^{(2)} - \cdots - A^{(L-1)} - C^{(L)}$. All A matrices are left-normalized. The last tensor $C^{(L)}$ is not normalized and contains the norm information.

7 Right-Canonical Form

- **Goal:** To build an orthonormal basis from right to left.
- **Property:** A set of matrices $\{B^{\sigma_i}\}$ is right-normalized if $\sum_{\sigma_i} B^{\sigma_i}(B^{\sigma_i})^\dagger = I$.



- **Intuition:** The tensor $B^{(i)}$ takes the orthonormal basis of the block $(i+1 \dots L)$, adds site i, and produces a new orthonormal basis for the block $(i \dots L)$.
- **Procedure:** A right-to-left sweep.
- **Resulting MPS:** $C^{(1)} - B^{(2)} - \cdots - B^{(L)}$. All B matrices are right-normalized. The first tensor $C^{(1)}$ is not normalized.

8 Mixed-Canonical Form and the Orthogonality Center

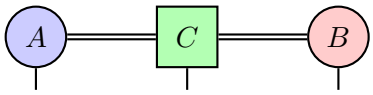
This is the most useful form, combining the two above.

- **Definition:** An MPS has an **orthogonality center** when all tensors to the left of the center are left-canonical (A's) and all tensors to the right are right-canonical (B's). The center itself is not normalized.

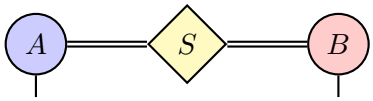
Q: Does the orthogonality center correspond to a physical part of the lattice?

A: Yes, and there are three common types:

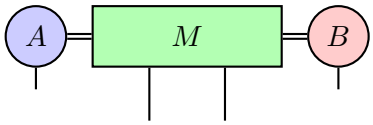
- **Site-Centered (at site l):** $\cdots A^{(l-1)} - C^{(l)} - B^{(l+1)} \cdots$
  Useful for single-site updates. The right part starts at $l+1$.
- **Bond-Centered (at bond $(l, l+1)$):** $\cdots A^{(l)} - S^{(l)} - B^{(l+1)} \cdots$
  The most explicit form of the Schmidt decomposition. The right part starts at $l+1$.
- **Two-Site Centered (at sites $(l, l+1)$):** $\cdots A^{(l-1)} - M^{(l,l+1)} - B^{(l+2)} \cdots$
  Essential for two-site DMRG. The right part starts at $l+2$.



Site-Centered



Bond-Centered



Two-Site Centered

Q: For a fully left-canonical state, where does the entanglement of the original quantum state go? Does all the entanglement reside in just the S matrix?

A: Yes. For that specific bipartition, the A and B blocks just build orthonormal bases (which are "unentangled" with each other), and the S matrix contains the complete information about how these bases are entangled to form the true state.

Q: For a fully left-canonical state, where does the entanglement of the original quantum state go?

A: It has not vanished. It has been implicitly encoded into the structure of the A matrices themselves. A fully canonical MPS is a representation of the entire state's interwoven complexity. To see the entanglement at a specific cut, you must perform a partial sweep to move the orthogonality center to that cut, which makes a new S matrix emerge, revealing the entanglement that was previously implicit.

9 Rank, Entanglement, and the Structure of States

Q: What does it mean for a matrix to have full rank? Why must a product state have rank 1?
A:

- **Rank:** The number of linearly independent rows or columns. It's the "true" dimensionality of the information.
- **Coefficient Matrix C:** For a two-part system A and B, we can arrange the state's coefficients c_{ij} into a matrix C. The rank of this matrix is the Schmidt rank.

- **Product State (Rank 1):** A state $|\psi\rangle = |\psi_A\rangle \otimes |\psi_B\rangle$ has a coefficient matrix where every row is a multiple of every other row. They are linearly dependent, so the rank is 1. No entanglement.

- **Entangled State (Rank ¿ 1):** The rows of the C matrix are linearly independent. You cannot factor the state. "Full rank" means the rank is maximal, corresponding to maximal entanglement.

Q: If a random state is a sum of product basis states, where does the entanglement come from?
A: The entanglement is not in the individual basis states (which are unentangled). It arises from the specific **pattern of the coefficients** c in the superposition. For a random state, these coefficients are random, leading to a coefficient matrix C that has full rank, signifying maximal entanglement between the two halves.

10 Bond Dimension and Entanglement

Q: What happens to the bond dimension during canonicalization? Does it change?
A: *Pure* canonicalization (just changing the gauge) **does not** change the bond dimension. However, canonicalization is almost always paired with **truncation**. The SVD step reveals the singular values, and if many are negligibly small, we discard them and the corresponding basis vectors, actively reducing the bond dimension. This turns the MPS into a controlled approximation.

Q: What does the bond dimension represent?
A:

- The **bond dimension** D is the **capacity** for entanglement at a bond. It's the maximum Schmidt rank the MPS can represent there.

- The **actual entanglement** is the **Schmidt rank** (number of non-zero singular values).

Q: Why aren't extra singular values exactly zero for a physical ground state?
A: The Area Law often means entanglement is an infinite sum with exponentially decaying terms, not a finite one. The true Schmidt rank might be huge or infinite, but the *effective* rank needed for a given precision is small and finite. We truncate the "entanglement tail" and accept a tiny error for a massive speedup.

Q: At the ends of the chain, should the bond dimension be 1?
A: This is a crucial point of terminology.

- The **dummy virtual bond** that terminates the chain (e.g., to the left of site 1) has dimension **1**.

- The **first physical virtual bond** (e.g., between site 1 and 2) has a dimension $D_1 > 1$ (typically d). It must carry the entanglement between site 1 and the rest of the chain.

11 Constructing an MPS from a Quantum State

This is done via an iterative SVD procedure, peeling off one site at a time.

1. **Start:** Take the full coefficient tensor $c_{\sigma_1 \dots \sigma_L}$.

2. **Reshape & SVD:** Reshape it into a matrix M separating site 1 from the rest. SVD it: $M = USV^\dagger$.

3. **Identify Tensor:** U becomes the first MPS tensor $A(1)$. It is guaranteed to be left-canonical.

4. **Remainder:** The remainder $SV^\dagger$ is passed to the next step.

5. **Iterate:** Reshape the remainder to separate site 2, do another SVD, get $A(2)$, and so on.

- If the original state is a **product state**, this procedure will automatically produce an MPS with all bond dimensions equal to 1.

- If the original state is **entangled**, the SVDs will reveal the necessary Schmidt ranks, and the procedure will produce an MPS with the exact, non-trivial bond dimensions required to represent the state perfectly.

Q: If we keep bond dimensions at $2^{L/2}$, can we represent a random state?
A: Yes, theoretically. An MPS with exponential bond dimension has the capacity to represent *any* state. However, this completely defeats the purpose of MPS, as the computational cost ($\sim D^3$) would be even worse than working in the full Hilbert space. The success of MPS relies on the fact that physical ground states in 1D have low entanglement (Area Law) and can be accurately represented with a small, *finite* bond dimension.

12 Why Bond Dimensions Change During Canonicalization

A common source of confusion when working with Matrix Product States (MPS) is observing that the bond dimensions of the tensors change during the canonicalization procedure, even though canonicalization is often described as a simple "gauge transformation". This apparent contradiction is resolved by understanding that SVD-based canonicalization is a powerful process that actively discovers and sets the **minimum necessary bond dimension** required at each bond to represent the state.

12.1 The Physics: Bond Dimension as an Information Channel

The virtual bond connecting any two sites in an MPS acts as an information channel that must be "wide" enough to carry all the quantum correlations (entanglement) between the part of the chain to its left (Block A) and the part to its right (Block B).

The absolute minimum width required for this channel is a precise physical quantity called the **Schmidt rank**. This is the number of terms, D_i, in the Schmidt decomposition for the cut at bond i:

$$|\psi\rangle = \sum_{\alpha=1}^{D_i} s_\alpha |\phi_\alpha\rangle_A |\chi_\alpha\rangle_B \tag{10}$$

Crucially, the Schmidt rank D_i is a property of the state itself and is generally **different for every cut** in the chain.

- **At the Ends:** The Schmidt rank for a cut between site 1 and site 2, denoted D_1, can be at most d, where d is the dimension of the physical state at a single site. For a spin-1/2 system, $d = 2$.

- **In the Middle:** The entanglement is typically maximal in the middle of the chain. The Schmidt rank $D_{L/2}$ will be at its largest value here.

- **The Profile:** Therefore, the physically necessary bond dimension is not constant. It starts small at the edges of the chain, grows towards the center, and then shrinks again towards the other end. For an open boundary MPS, this profile is a fundamental feature.

12.2 The Mathematics: How SVD-based Canonicalization Discovers the Profile

The left canonicalization algorithm automatically discovers this physical profile of bond dimensions by using the Singular Value Decomposition (SVD), which is a **rank-revealing decomposition**. Let's trace the first two steps, starting from a random MPS where all internal bonds have been initialized with a large, uniform dimension $D_{\max}$ (e.g., 16).

12.2.1 Step 1: Canonicalize Site 0

1. **Tensor and Shape:** We take the first tensor, $M^{(0)}$, which has shape $(d, 1, D_{\max})$. The left bond dimension is 1 due to the open boundary.
2. **Reshape for SVD:** We combine the physical and left indices to form a matrix.
$$\text{matrix}^{(0)} = \text{reshape}(M^{(0)}, \quad (d \times 1, \quad D_{\max}))$$
For a spin-1/2 system ($d = 2$) with $D_{\max} = 16$, this matrix has shape $(2, 16)$.
3. **Perform SVD:** We decompose this matrix: $\text{matrix}^{(0)} = USV^\dagger$. The rank of a 2×16 matrix can be at most 2. The SVD will find that there are only $D_1 = \min(d, D_{\max}) = 2$ non-zero singular values. The resulting matrices will have shapes:
   - U: $(2, 2)$
   - S: $(2,)$
   - $V^\dagger$: $(2, 16)$
4. **Create New Tensor $A^{(0)}$:** The new, left-canonical tensor $A^{(0)}$ is formed from the isometric part, U. We reshape it back to the standard tensor format:
$$A^{(0)} = \text{reshape}(U, \quad (d, 1, D_1)) = \text{reshape}(U, \quad (2, 1, 2))$$
The right bond dimension of the first tensor has now been automatically set to the minimum necessary value, $D_1 = 2$. The algorithm has discovered that a dimension of 16 was unnecessary here.
5. **Pass the Remainder:** The non-isometric part, $R^{(1)} = \text{diag}(S) \cdot V^\dagger$, which has shape $(2, 16)$, is passed to the right and contracted with the next tensor, $M^{(1)}$. The new temporary tensor $M'^{(1)}$ now has a *left* bond dimension of 2.

12.2.2 Step 2: Canonicalize Site 1

1. **Tensor and Shape:** We take the temporary tensor $M'^{(1)}$, which now has shape $(d, D_1, D_{\max}) = (2, 2, 16)$.
2. **Reshape for SVD:**
$$\text{matrix}^{(1)} = \text{reshape}(M'^{(1)}, \quad (d \times D_1, \quad D_{\max}))$$
This matrix has shape $(4, 16)$.
3. **Perform SVD:** The SVD will find $D_2 = \min(d \cdot D_1, D_{\max}) = 4$ non-zero singular values.
4. **Create New Tensor $A^{(1)}$:** The new tensor is formed from the resulting U' matrix of shape $(4, 4)$:
$$A^{(1)} = \text{reshape}(U', \quad (d, D_1, D_2)) = \text{reshape}(U', \quad (2, 2, 4))$$
The bond dimension between sites 1 and 2 has now grown to $D_2 = 4$.

12.3 Conclusion

Canonicalization is more than just a gauge transformation; it is an active process of **discovery and compression**. The SVD at each step interrogates the local structure of the state and determines the true Schmidt rank for that cut. It naturally compresses the representation by discarding any "unnecessary" dimensions from the initial (potentially oversized) bonds, automatically shaping the MPS to reflect the physical entanglement profile of the quantum state.

13 The Full Canonicalization Workflow: From Random MPS to Vidal Gauge

This section details the complete, step-by-step process for transforming a general, non-canonical Matrix Product State (MPS) into any of the standard canonical forms, culminating in the Vidal gauge (Γ-Λ) representation. We start with a random MPS for a 4-site chain, denoted by the tensors $M^{(i)}$.

Initial State: A Random MPS

We begin with a list of tensors with no special properties, where the bond dimensions may be oversized.

$$|\psi\rangle \propto \quad M^{(0)}—M^{(1)}—M^{(2)}—M^{(3)} \tag{11}$$

13.1 Step 1: Left-Canonicalization

Goal: To transform the MPS so all tensors except the last are left-normalized (A-tensors), placing the orthogonality center at the last site. This is achieved with a left-to-right sweep.

1.1: Canonicalize Site 0

1. **Decomposition:** Reshape and perform an SVD on the first tensor: $M^{(0)} = U_0 S_0 V_0^\dagger$.
2. **Physics:** This is the Schmidt decomposition between site 0 and the rest of the chain $\{1, 2, 3\}$.
3. **Action:**
   - The new left-canonical tensor is $A^{(0)} = U_0$. By the properties of SVD, this is guaranteed to be an isometry satisfying $\sum_{\sigma_0} (A^{(0)})^\dagger A^{(0)} = I$.
   - The remainder, $R^{(1)} = S_0 V_0^\dagger$, contains the entanglement and norm information. It is passed to the right by contracting it with the next tensor: $M'^{(1)} = R^{(1)} M^{(1)}$.
4. **Result:** The chain is now $A^{(0)}—M'^{(1)}—M^{(2)}—M^{(3)}$. Site 0 is canonical.

1.2: Sweep to the Right

We repeat this process for each subsequent site. For site 1, we decompose $M'^{(1)} = U_1 S_1 V_1^\dagger$. The new tensor is $A^{(1)} = U_1$, and the new remainder $R^{(2)} = S_1 V_1^\dagger$ is absorbed into $M^{(2)}$. This continues until the last bond.

Final Result: Left-Canonical Form

After sweeping through to the last bond, the MPS has the form:

$$A^{(0)}—A^{(1)}—A^{(2)}—C^{(3)} \tag{12}$$

where $A^{(0)}, A^{(1)}, A^{(2)}$ are all left-normalized. The final tensor $C^{(3)}$ is not normalized and contains the entire norm of the state. The **orthogonality center is at site 3**.

13.2 Step 2: Right-Canonicalization

Goal: To transform the result from Step 1 into a fully right-canonical MPS, where the orthogonality center is at site 0. This requires a right-to-left sweep.

2.1: Canonicalize Site 3

1. **Decomposition:** Take the current center tensor, $C^{(3)}$, and perform an SVD: $C^{(3)} = U_3 S_3 V_3^\dagger$.
2. **Action:**
   - The new right-canonical tensor is $B^{(3)} = V_3^\dagger$. SVD guarantees this is an isometry satisfying $\sum_{\sigma_3} B^{(3)}(B^{(3)})^\dagger = I$.
   - The remainder, $L^{(2)} = U_3 S_3$, is passed to the *left*: $M'^{(2)} = A^{(2)} L^{(2)}$.
3. **Result:** The chain is now $A^{(0)}—A^{(1)}—M'^{(2)}—B^{(3)}$. The orthogonality center has moved to site 2.

Final Result: Right-Canonical Form

Continuing this right-to-left sweep until we reach site 0, the final MPS has the form:

$$C^{(0)}—B^{(1)}—B^{(2)}—B^{(3)} \tag{13}$$

where $B^{(1)}, B^{(2)}, B^{(3)}$ are all right-normalized. The first tensor $C^{(0)}$ is now the orthogonality center.

13.3 Step 3: Mixed-Canonicalization

Goal: To move the orthogonality center to a specific site, e.g., site 2. We start from the right-canonical form ‘C(0)—B(1)—B(2)—B(3)‘.

This is achieved by a partial left-to-right sweep that stops at the desired site. This is precisely the logic of the `update_orthogonality_center` function.

1. **Move center $0 \to 1$:** Decompose $C^{(0)} = U_0 S_0 V_0^\dagger$. Set $A^{(0)} = U_0$. Absorb the remainder into $B^{(1)}$ to create a new center, $C^{(1)} = (S_0 V_0^\dagger) B^{(1)}$. The chain is now $A^{(0)}—C^{(1)}—B^{(2)}—B^{(3)}$.
2. **Move center $1 \to 2$:** Decompose $C^{(1)} = U_1 S_1 V_1^\dagger$. Set $A^{(1)} = U_1$. Absorb the remainder into $B^{(2)}$ to create the final center, $C^{(2)} = (S_1 V_1^\dagger) B^{(2)}$.

Final Result: Mixed-Canonical Form

The sweep stops. The final MPS has the form:

$$A^{(0)}—A^{(1)}—C^{(2)}—B^{(3)} \tag{14}$$

This state is left-canonical to the left of site 2, right-canonical to the right of site 2, with the orthogonality center at site 2.

13.4 Step 4: Transformation to Vidal Gauge (Γ-Λ Form)

Goal: To transform the mixed-canonical MPS into a representation where all site tensors (Γ) are isometries and the singular values (Λ) live on the bonds.

4.1: Find all Lambda (Λ) Matrices

The singular values on bond $(i, i+1)$ are, by definition, found by SVD'ing the center tensor when the center is at site i.

1. Take the input mixed-canonical MPS, e.g., $A^{(0)}$—$A^{(1)}$—$C^{(2)}$—$B^{(3)}$.
2. **To get Λ_1:** Move the center to site 0. SVD the center tensor $C^{(0)}$. The singular values S give Λ_1.
3. **To get Λ_2:** Move the center to site 1. SVD the center tensor $C^{(1)}$. The singular values S' give Λ_2.
4. **To get Λ_3:** Move the center to site 2. SVD the center tensor $C^{(2)}$. The singular values S'' give Λ_3.
5. We also define dummy matrices $\Lambda_0 = \Lambda_4 = [1]$.

4.2: Calculate all Gamma (Γ) Tensors

With all Λ matrices known, we use the original mixed-canonical MPS and the defining relations to solve for each Γ.

- For an A-tensor (e.g., site 1): $A^{(1)} = \Gamma^{(1)}\Lambda_2 \implies \Gamma^{(1)} = A^{(1)}(\Lambda_2)^{-1}$.
- For a B-tensor (e.g., site 3): $B^{(3)} = \Lambda_3\Gamma^{(3)} \implies \Gamma^{(3)} = (\Lambda_3)^{-1}B^{(3)}$.
- For the C-tensor (site 2): $C^{(2)} = \Lambda_2\Gamma^{(2)}\Lambda_3 \implies \Gamma^{(2)} = (\Lambda_2)^{-1}C^{(2)}(\Lambda_3)^{-1}$.

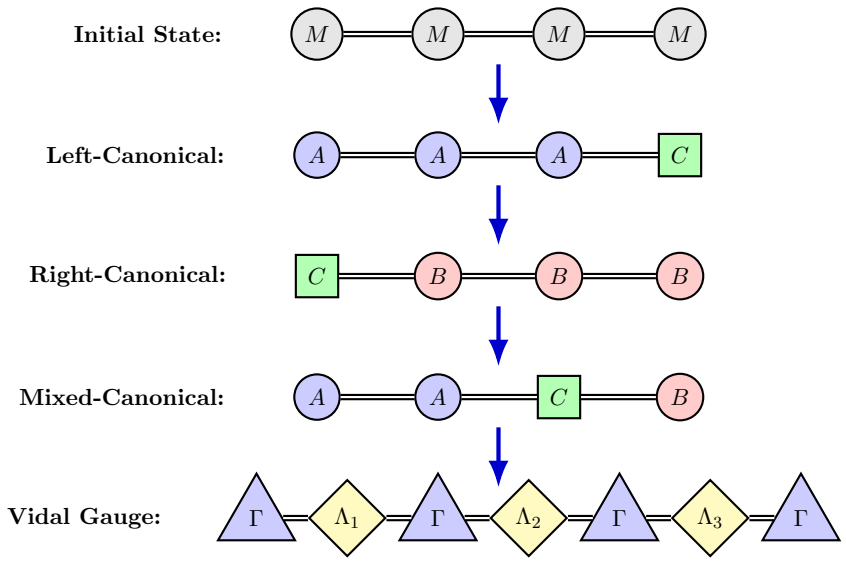


Figure 2: The full workflow of transforming a random MPS into various canonical forms.

Final Result: Vidal Gauge

The final representation explicitly separates the isometric tensors from the entanglement information on the bonds:

$$\Lambda_0—\Gamma^{(0)}—\Lambda_1—\Gamma^{(1)}—\Lambda_2—\Gamma^{(2)}—\Lambda_3—\Gamma^{(3)}—\Lambda_4 \tag{15}$$

14 SVD Compression: The Fast, Greedy Approach

SVD compression is a direct, non-iterative algorithm for reducing the bond dimension of an MPS. It operates by making a series of locally optimal decisions at each bond, which makes it fast but not globally optimal.

14.1 The Core Idea: Entanglement and the Schmidt Decomposition

The physics behind SVD compression is rooted in the **Schmidt decomposition**. For any quantum system partitioned into two subsystems, A and B, the state can be written as:

$$|\psi\rangle = \sum_{\alpha=1}^{r} s_\alpha |\alpha\rangle_A |\alpha\rangle_B \tag{16}$$

where s_α are the real, non-negative Schmidt coefficients, and r is the Schmidt rank. The magnitude of s_α^2 represents the probability of finding the system in the corresponding state $|\alpha\rangle_A |\alpha\rangle_B$. Truncating the state by discarding the components with the smallest Schmidt coefficients provides the best possible low-rank approximation for that specific A-B cut.

The crucial link to MPS is the **mixed-canonical form**. When an MPS is in mixed-canonical form with the orthogonality center placed on the bond between sites l and $l+1$, the state is explicitly in the Schmidt decomposition form:

$$|\psi\rangle = \sum_{\alpha=1}^{D'} (S^{(l)})_{\alpha\alpha} |\alpha\rangle_{1\ldots l} |\alpha\rangle_{l+1\ldots L} \tag{17}$$

The matrix of singular values, $S^{(l)}$, on that bond directly contains the Schmidt coefficients. The SVD compression algorithm leverages this fact to identify and discard the least important components of the entanglement.

14.2 The Algorithm: A Stable Two-Sweep Procedure

A robust implementation of SVD compression involves two sweeps to ensure stability and proper normalization.

1. **Preparation Sweep (Right-to-Left):**
   (a) Take the initial, large-bond-dimension MPS $|\psi\rangle$.
   (b) Bring it into a fully normalized, **right-canonical form** by performing a sweep from right to left using RQ decompositions. This is achieved by calling a function like your `make_right_canon`.
   (c) This preparatory step "sweeps" all the norm and non-orthonormal parts of the state to the leftmost tensor, ψ_0, which is then normalized. The result is a numerically stable MPS, ready for truncation.
2. **Truncation Sweep (Left-to-Right):**
   (a) Start at the leftmost site, $i = 0$. The orthogonality center is now at site 0.

(b) Perform an SVD on the tensor ψ_i, splitting it into $U, S, V^\dagger$. The diagonal of S contains the Schmidt coefficients for the bond $(i, i+1)$.

(c) **Decision:** Based on the chosen strategy, determine the new, smaller bond dimension, d_{new}. This can be a fixed maximum value (`max_bond_dim`) or determined adaptively by a truncation error tolerance, ϵ_{tol}.

(d) **Truncate:** Slice the matrices $U, S, V^\dagger$ to the new dimension d_{new}, creating $\tilde{U}, \tilde{S}, \tilde{V}^\dagger$. The sum of the squares of the discarded singular values gives the local truncation error for that bond.

(e) **Re-normalize $\tilde{S}$:** Normalize the truncated singular value matrix, $\tilde{S}$, so its Frobenius norm is 1. This is a vital step to preserve the overall unit norm of the state as we sweep.

(f) **Absorb and Update:**

- The new tensor for site i becomes the truncated, left-orthonormal $\tilde{U}$.
- The product $(\tilde{S}\tilde{V}^\dagger)$ is absorbed into the next tensor to the right: $\psi_{i+1} \leftarrow (\tilde{S}\tilde{V}^\dagger) \cdot \psi_{i+1}$. This moves the orthogonality center to site $i+1$.

(g) Repeat for all sites $i = 0, \ldots, N-2$. The final MPS is left-canonical and compressed.

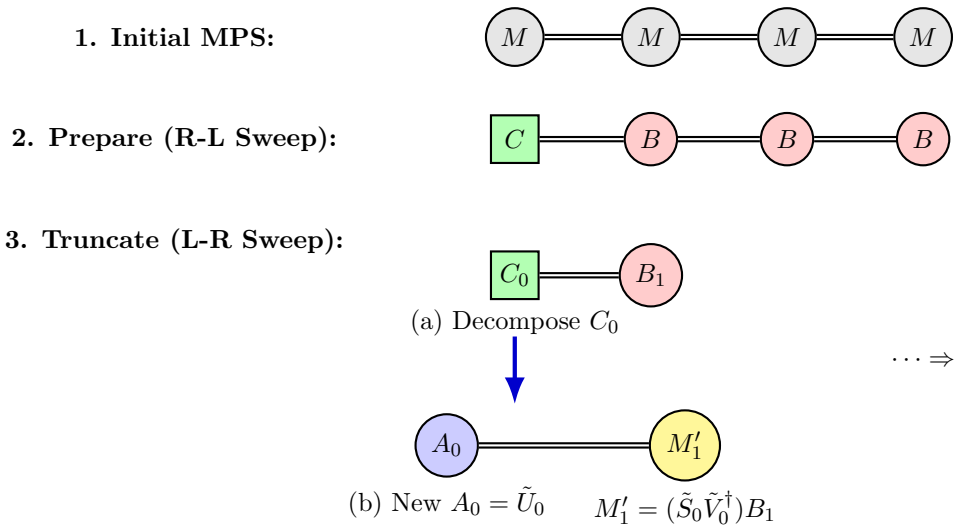
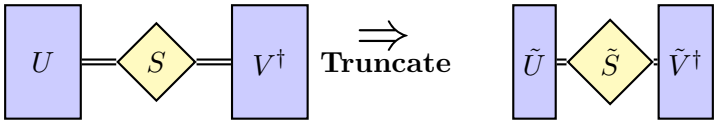


Figure 3: The two-sweep process of SVD Compression.

14.3 SVD for Optimal Truncation and Cost

The power of SVD lies in its use for compression. The Eckart-Young theorem guarantees that the best rank-d_{new} approximation of a matrix is found by keeping the d_{new} largest singular values and discarding the rest. The computational cost for decomposing an $m \times n$ matrix (with $m \geq n$) scales as $\mathcal{O}(mn^2)$.



The error introduced by this truncation is precisely the sum of the squares of the discarded singular values: $\epsilon^2 = \sum_{i>d_{\text{new}}} s_i^2$.

The cost of QR and RQ decomposition for an $m \times n$ matrix is also $\mathcal{O}(mn^2)$, but with a smaller prefactor, making them numerically faster than SVD.

14.4 Summary: Advantages and Disadvantages

SVD Compression Summary

+ **Fast and Direct:** It is a non-iterative, single-pass (or two-pass) algorithm.

+ **Locally Optimal:** At each bond, the truncation is guaranteed by the Eckart-Young theorem to be the best possible low-rank approximation.

- **Not Globally Optimal:** The algorithm is "greedy." The truncation decision at one bond affects the basis for all subsequent truncations. These local errors accumulate, and the final state is not guaranteed to be the globally best approximation.

- **Costly for Very Large D':** The cost of the SVD scales as $\mathcal{O}((D')^3)$, making it slow if the initial bond dimension is extremely large.

15 Variational Compression: The Optimal Approach

Variational compression directly optimizes the global fidelity between the original and compressed states, making it the most accurate method. It is an iterative algorithm that conceptually mirrors the logic of DMRG.

15.1 The Core Idea: Direct Minimization of Global Error

The goal is to find the compressed MPS $|\phi\rangle$ that minimizes the global error $\||\psi\rangle - |\phi\rangle\|^2$. Expanding this expression, we find that minimizing the error is equivalent to maximizing the overlap (fidelity) $\mathrm{Re}(\langle\psi|\phi\rangle)$.

The overlap is a complex function of all tensors in $|\phi\rangle$ simultaneously. The key insight of the variational approach is to optimize the tensors one at a time, keeping all others fixed, and repeating this process in sweeps until the overlap converges.

15.2 The Algorithm: One-Site Optimization

The heart of the algorithm is optimizing a single tensor, ϕ_i, while all its neighbors $\{\phi_j\}_{j\neq i}$ are held constant.

1. **The Linear Dependence:** When all other tensors are frozen, the global overlap $\langle\psi|\phi\rangle$ becomes a *linear function* of the elements of the single tensor ϕ_i^*.

2. **The Environment Tensor:** We can define an "environment tensor" E_i by contracting the entire network with a "hole" at site i. E_i encapsulates the influence of the target state $|\psi\rangle$ and all the frozen parts of our guess $|\phi\rangle$. The overlap then becomes a simple contraction:
$$\langle\psi|\phi\rangle = \mathrm{contract}(E_i, \phi_i^*) \tag{18}$$

3. **The Optimal Solution:** To maximize this linear overlap, the optimal choice for the new tensor is simply proportional to the environment tensor itself: $\phi_i^{\mathrm{new}} \propto E_i$.

15.3 Comparison with Schollwöck's Formalism

This procedure is a practical implementation of the formalism in Schollwöck's review (Sec. 4.5.2).

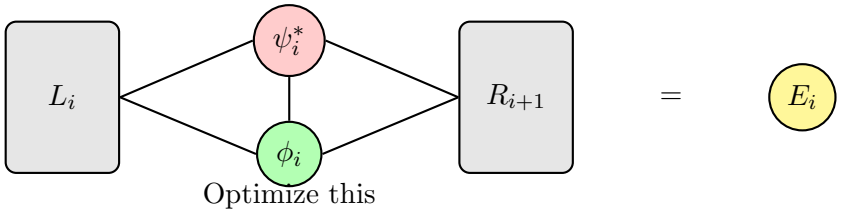
- Schollwöck derives the solution by setting the derivative of the error with respect to ϕ_i^* to zero. This leads to the general linear system $P \cdot v = b$ (his Eq. 141).

- In this equation, v corresponds to our unknown tensor ϕ_i, and b is precisely our environment tensor E_i.
- The matrix P represents the environment of site i contracted *within the guess state* $|\phi\rangle$ *itself.*
- The crucial simplification arises from using the **canonical form**. When $|\phi\rangle$ is in mixed-canonical form with the orthogonality center at site i, the environment matrix P is guaranteed to be the identity matrix, $P = I$.
- This reduces the difficult linear system $P \cdot v = b$ to the trivial solution $v = b$, which is exactly $\phi_i = E_i$ (his Eq. 147).

The sweeping algorithm, which re-normalizes with QR/RQ at each step, is the efficient method that ensures $P = I$ is always true at the site being optimized.

15.4 The Full Sweeping Algorithm

1. **Initialization:**
   - Start with the target MPS $|\psi\rangle$ and an initial guess $|\phi\rangle$ (ideally from SVD compression).
   - Normalize $|\phi\rangle$ and prepare environment containers.
2. **Sweeping Procedure (repeated until convergence):**
   (a) **Prepare Right Environments:** Sweep once from right to left, calculating and storing all right environment blocks $[R_i]$.
   (b) **Forward Sweep (L-R):** For $i = 0, \ldots, N-1$:
      - Calculate the environment E_i using the stored left block L_i and right block R_{i+1}.
      - Update $\phi_i \leftarrow E_i$.
      - Re-normalize ϕ_i with a QR decomposition, absorbing the R matrix into ϕ_{i+1} to move the orthogonality center.
      - Update the left environment block L_{i+1}.
   (c) **Backward Sweep (R-L):** For $i = N-1, \ldots, 0$:
      - Calculate the environment E_i using L_i and R_{i+1}.
      - Update $\phi_i \leftarrow E_i$.
      - Re-normalize ϕ_i with an RQ decomposition, absorbing the R matrix into ϕ_{i-1}.
      - Update the right environment block R_i.
3. **Convergence Check:** After each full two-way sweep, check if the fidelity has stopped increasing.



At each step, we optimize a single tensor ϕ_i by calculating its environment E_i. The environment is the contraction of the target MPS $|\psi\rangle$ with all frozen tensors from the guess MPS $|\phi\rangle$. The optimal update is then $\phi_i^{\text{new}} \propto E_i$.

Figure 4: The local optimization step in variational compression.

15.5 Summary: Advantages and Disadvantages

Variational Compression Summary

+ **Globally Optimal:** Finds a state with higher (or equal) fidelity than SVD compression. It finds a local minimum of the *global* error function, which is typically the true optimum.

+ **Highly Accurate:** This is the state-of-the-art method when precision is the top priority.

- **Iterative and Slower:** Requires multiple sweeps to converge, making it slower than the direct SVD method.

- **Requires Good Initialization:** While it can start from a random guess, it converges much faster and more reliably if initialized with the result from SVD compression.

16 Compression Conclusion: Which Method to Use?

The choice of compression algorithm is a trade-off between speed and accuracy.

- For applications where speed is critical and a small amount of sub-optimality is acceptable (e.g., some real-time evolution algorithms), **SVD Compression** is often used.

- For applications where the highest possible accuracy is required for a given bond dimension (e.g., finding ground states, or after adding/applying operators), **Variational Compression** is the superior method.

A common and robust strategy in modern tensor network codes is to use **SVD compression to get a good initial guess, and then refine it with a few sweeps of variational compression** to achieve the true optimum.

Extra Topics

17 The Cost of MPS Algorithms and the Role of D

While we established that MPS provides an efficient compression, it's crucial to understand how the cost scales. Most MPS algorithms (like DMRG, TEBD) involve contracting local networks.

- **The Key Operation:** A typical operation is contracting an MPS tensor with an operator and another MPS tensor. This forms a small local network. For example, calculating the expectation value $\langle\psi|\hat{O}_i|\psi\rangle$ in a canonical MPS involves contracting $A(i)^\dagger$, $A(i)$, and the operator O_i.
- **Cost Scaling:** The computational cost of these contractions is dominated by matrix multiplications involving the virtual indices. The cost scales polynomially with the bond dimension D and the local dimension d. A standard two-site DMRG update scales as $\mathcal{O}(D^3d^2)$.
- **The D^3 Bottleneck:** The cubic scaling with bond dimension is the main computational bottleneck. This is why truncating D from 500 to 64 provides such a massive speedup (a factor of $(500/64)^3 \approx 475$). Doubling the bond dimension makes the calculation roughly 8 times slower. This highlights the trade-off between accuracy (higher D) and speed (lower D).

18 QR vs. SVD for Canonicalization

In the procedure for canonicalization, we mentioned using QR or SVD. They have different trade-offs.

- **QR Decomposition ($M = QR$):**
  - **Faster:** Numerically less expensive than SVD.
  - **"Dumb":** It produces a left-orthogonal matrix Q but tells you nothing about the entanglement structure. The leftover matrix R is just a triangular matrix.
  - **Use Case:** Ideal when you simply need to enforce the canonical form without needing to know or change the bond dimension. For example, just re-centering the orthogonality center without truncation.
- **Singular Value Decomposition ($M = USV^\dagger$):**
  - **Slower:** More computationally demanding.
  - **"Smart":** It provides a complete diagnosis of the bond. The S matrix contains the singular values, which are the Schmidt coefficients.
  - **Use Case:** Essential whenever you need to **truncate** the bond dimension. You *must* use SVD to identify which states are the least important (those with the smallest singular values) so you can safely discard them. This is the core of DMRG's truncation step.

19 Matrix Product Operators (MPOs)

Just as an MPS is a way to efficiently represent a *state vector*, a **Matrix Product Operator (MPO)** is an efficient way to represent a *linear operator* (like a Hamiltonian).

- **Structure:** An MPO has the same tensor network structure as an MPS, but each tensor has *two* physical indices: one "input" and one "output" leg.

$$O_{\sigma_1 \dots \sigma_L, \sigma'_1 \dots \sigma'_L} = W^{\sigma_1, \sigma'_1} W^{\sigma_2, \sigma'_2} \cdots W^{\sigma_L, \sigma'_L} \tag{19}$$

- **Hamiltonians:** Hamiltonians with only local interactions (e.g., nearest-neighbor) can be represented *exactly* by an MPO with a very small, finite bond dimension (typically 3 to 5 for a simple spin chain).

- **Application:** Applying a Hamiltonian to a state ($H|\psi\rangle$) becomes an efficient operation. You simply place the MPO "on top" of the MPS and contract the physical indices. The result is a new MPS whose bond dimension is the product of the original MPS and MPO bond dimensions ($D_{\text{new}} = D_{\text{MPS}} \times D_{\text{MPO}}$). This new, higher-bond-dimension MPS must then be compressed/truncated back down. This apply-MPO-then-truncate cycle is the basis of many advanced algorithms like time evolution (TEBD).

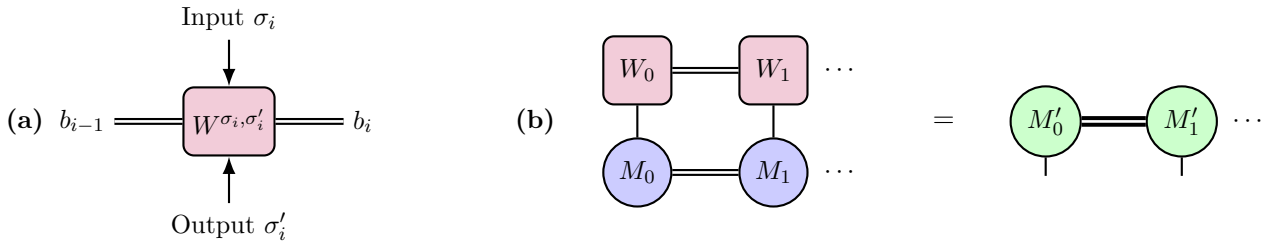


Figure 5: Graphical representation of MPOs. (a) A single MPO tensor with two physical indices (input and output) and two virtual indices. (b) Applying an MPO to an MPS by contracting their physical indices, resulting in a new MPS with a larger bond dimension.

20 Critical Systems and Finite Entanglement Scaling

- **Gapped vs. Critical Systems:** Our "Area Law" discussion primarily applies to **gapped** systems (systems with a finite energy gap above the ground state). In these systems, correlations decay exponentially, and a finite D can represent the state perfectly in the thermodynamic limit.

- **Critical Systems:** At a quantum phase transition, the energy gap closes, and correlations decay as a power law, not exponentially. The entanglement in 1D no longer saturates to a constant but grows **logarithmically** with the system size:

$$S \propto \log(L) \tag{20}$$

- **Consequence for MPS:** This means that to perfectly represent a critical state, the bond dimension D must also grow as a power law of the system size L. This is more demanding than a gapped system but still vastly more efficient than the exponential requirement for a random state.

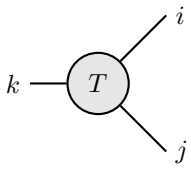
- **Finite Entanglement Scaling:** In practice, for a critical system, we fix a finite D. This finite D introduces an effective correlation length into our simulation. The MPS can only capture the true power-law physics up to this length scale. By studying how physical quantities change as we systematically increase D, we can extrapolate to the true $D \to \infty$ critical behavior.

21 Graphical Notation: The Rules of the Game

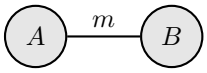
A quick summary of the visual language:

- **Tensors:** Shapes (circles, squares).
- **Indices:** Lines or "legs" coming out of the shapes.
- **Contraction (Summation):** A line connecting two tensors means the corresponding indices are summed over.
- **Tensor Product:** Simply placing two tensors near each other with no connecting line.
- **MPS:** A 1D chain of tensors connected by virtual bonds.
- **Canonical Form ($A^\dagger A = I$):** Graphically, this means if you take a tensor, flip it vertically to represent its conjugate ($\dagger$), and connect the physical and left virtual legs, the entire object "annihilates" and becomes a single line (an identity matrix).

Mastering this graphical language allows you to reason about complex tensor network manipulations without writing down pages of equations with messy indices. It's an incredibly powerful tool for both developing algorithms and understanding them.

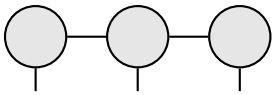


(a) Tensor and Indices

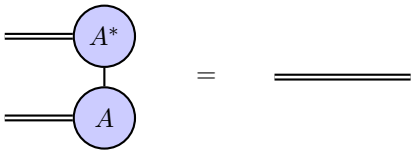


(b) Contraction ($\sum_m$)

(c) Tensor Product ($C \otimes D$)



(d) Matrix Product State



(e) Canonical Form ($A^\dagger A = I$)

Figure 6: A visual summary of the rules of graphical notation for tensor networks.