Simulation of Quantum Systems using Matrix Product States

Chase Hodges-Heilmann

Table of Contents

Bond Dimension:

Tensor Notation

An order-N tensor is an N-dimensional array. Scalars, vectors, and matrices are special cases of tensors:

- **Order 0 (scalar)**: m
- **Order 1 (vector)**: $\mathbf{m}$
- **Order 2 (matrix)**: $\mathbf{M}$
- **Order $N > 2$ (tensor)**: $\mathcal{M}$

Ballard˙Kolda˙2025

Tensor Notation

An order-N tensor is an N-dimensional array. Scalars, vectors, and matrices are special cases of tensors:

- **Order 0 (scalar)**: m
- **Order 1 (vector)**: $\mathbf{m}$
- **Order 2 (matrix)**: $\mathbf{M}$
- **Order $N > 2$ (tensor)**: $\mathcal{M}$

Element notation:

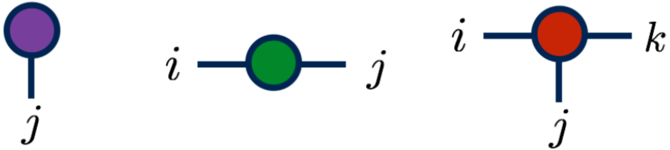
$$m_{i_1,i_2,\ldots,i_N} = \mathcal{M}(i_1, i_2, \ldots, i_N)$$

Ballard˙Kolda˙2025

Tensor Diagrams

Tensor network diagrams represent tensors as nodes with edges (legs) for each index:

- Vector (order 1): One leg
- Matrix (order 2): Two legs
- Order-3 tensor: Three legs



Vector $\mathbf{v}(i)$ Matrix $\mathbf{M}(i, j)$ Tensor $\mathfrak{T}(i, j, k)$

https://tensornetwork.org/

Tensor Contraction

Tensor contraction is a generalization of matrix multiplication, summing over shared indices:

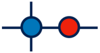
- Examples include:
  - Matrix–Vector product

$$= \sum_j M_{ij} v_j$$

  - Matrix–Matrix product

$$= A_{ij} B_{jk} = AB$$

  - Tensor–Matrix product



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

https://tensornetwork.org/

Tensor Train/Matrix Product States

A quantum state with N qubits can be written as

$$|\psi\rangle = \sum_{\sigma_1,\ldots,\sigma_N} c_{\sigma_1,\ldots,\sigma_N} |\sigma_1\rangle \otimes \ldots \otimes |\sigma_N\rangle. \qquad (1)$$

Ballard˙Kolda˙2025

Tensor Train/Matrix Product States

A quantum state with N qubits can be written as

$$|\psi\rangle = \sum_{\sigma_1,\ldots,\sigma_N} c_{\sigma_1,\ldots,\sigma_N} |\sigma_1\rangle \otimes \ldots \otimes |\sigma_N\rangle. \tag{1}$$

$c_{\sigma_1,\ldots,\sigma_N}$ is an entry from the N order tensor $\mathfrak{C} \in \mathbb{C}^{\sigma_1 \times \sigma_2 \times \ldots \times \sigma_N}$.

Ballard˙Kolda˙2025

Tensor Train/Matrix Product States

A quantum state with N qubits can be written as

$$|\psi\rangle = \sum_{\sigma_1,\ldots,\sigma_N} c_{\sigma_1,\ldots,\sigma_N} |\sigma_1\rangle \otimes \ldots \otimes |\sigma_N\rangle. \tag{1}$$

$c_{\sigma_1,\ldots,\sigma_N}$ is an entry from the N order tensor $\mathfrak{C} \in \mathbb{C}^{\sigma_1 \times \sigma_2 \times \ldots \times \sigma_N}$.
A tensor train (or matrix product state) expresses a high-order tensor $\mathfrak{C}$ as a product of lower-rank tensors:

$$\mathfrak{C} = \mathbf{M}_1 \cdot \mathfrak{M}_2 \cdot \cdots \cdot \mathfrak{M}_{N-1} \cdot \mathbf{M}_N \tag{2}$$

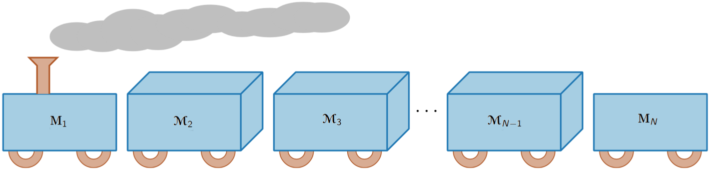


Figure 1: A Tensor "Train", each train car is called a core.

Ballard˙Kolda˙2025

Tensor Train / Matrix Product States

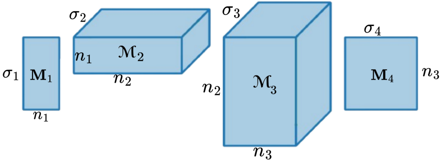


Figure 2: Dimensions of tensors in 4 site MPS, the n_i dimensions are called bond dimensions.

Ballard˙Kolda˙2025

Tensor Train / Matrix Product States

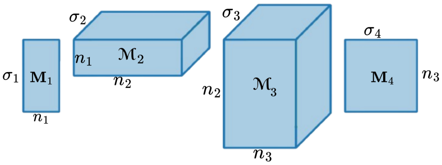
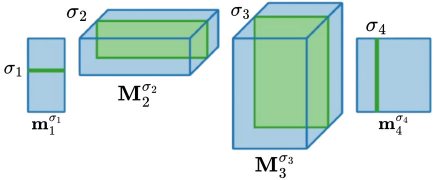


Figure 2: Dimensions of tensors in 4 site MPS, the n_i dimensions are called bond dimensions.



Ballard˙Kolda˙2025

MPS Diagram

We can write our quantum state now as

$$|\psi\rangle = \sum_{\sigma_1,\ldots\sigma_N} \mathbf{m}_1^{\sigma_1}\mathbf{M}_2^{\sigma_2}\ldots\mathbf{M}_{N-1}^{\sigma_{N-1}}\mathbf{m}_N^{\sigma_N}|\sigma_1\rangle \otimes \ldots \otimes |\sigma_N\rangle \qquad (3)$$

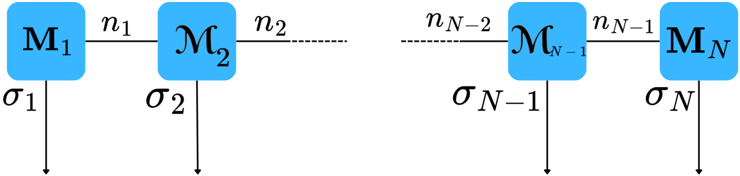
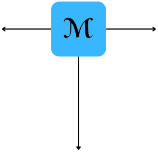
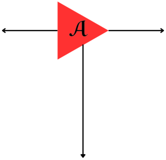


Figure 3: Tensor Diagram of MPS

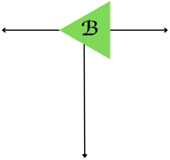
Tensor Orthogonality



Non-Orthogonal



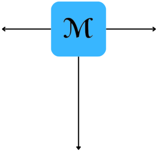
Left-Orthogonal



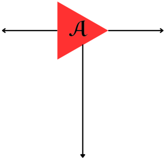
Right-Orthogonal

PAECKEL2019167998

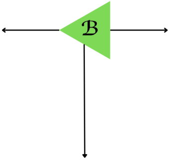
Tensor Orthogonality



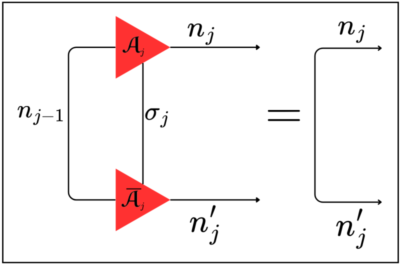
Non-Orthogonal



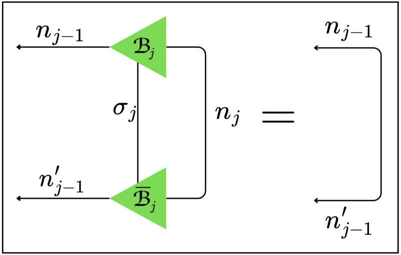
Left-Orthogonal



Right-Orthogonal



Left-Orthogonality:
$\sum_{\sigma_j} (\mathbf{A}_j^{\sigma_j})^\dagger \mathbf{A}_j^{\sigma_j} = I$

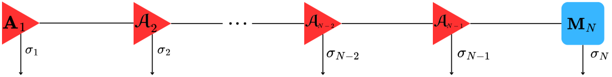


Right-Orthogonality:
$\sum_{\sigma_j} \mathbf{B}_j^{\sigma_j} (\mathbf{B}_j^{\sigma_j})^\dagger = I$

PAECKEL2019167998

Matrix Product State Canonical Forms

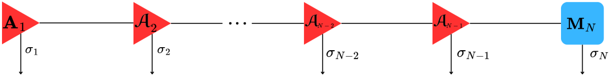
- Left-Orthogonal canonical form:



$$|\psi\rangle = \sum_{\boldsymbol{\sigma}} \mathbf{a}_1^{\sigma_1}\mathbf{A}_2^{\sigma_2}\ldots\mathbf{A}_{N-1}^{\sigma_{N-1}}\mathbf{m}_N^{\sigma_N}|\boldsymbol{\sigma}\rangle,$$

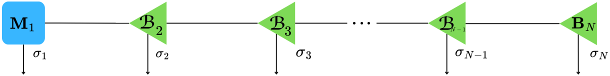
Matrix Product State Canonical Forms

- Left-Orthogonal canonical form:



$$|\psi\rangle = \sum_{\boldsymbol{\sigma}} \mathbf{a}_1^{\sigma_1} \mathbf{A}_2^{\sigma_2} \ldots \mathbf{A}_{N-1}^{\sigma_{N-1}} \mathbf{m}_N^{\sigma_N} |\boldsymbol{\sigma}\rangle,$$

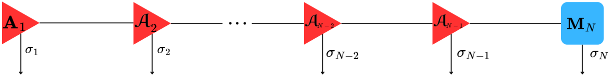
- Right-Orthogonal canonical form:



$$|\psi\rangle = \sum_{\boldsymbol{\sigma}} \mathbf{m}_1^{\sigma_1} \mathbf{B}_2^{\sigma_2} \ldots \mathbf{B}_{N-1}^{\sigma_{N-1}} \mathbf{b}_N^{\sigma_N} |\boldsymbol{\sigma}\rangle,$$

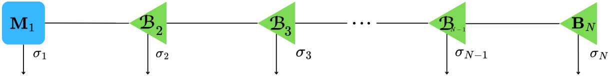
Matrix Product State Canonical Forms

- Left-Orthogonal canonical form:



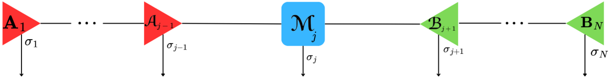
$$|\psi\rangle = \sum_{\boldsymbol{\sigma}} \mathbf{a}_1^{\sigma_1} \mathbf{A}_2^{\sigma_2} \ldots \mathbf{A}_{N-1}^{\sigma_{N-1}} \mathbf{m}_N^{\sigma_N} |\boldsymbol{\sigma}\rangle,$$

- Right-Orthogonal canonical form:



$$|\psi\rangle = \sum_{\boldsymbol{\sigma}} \mathbf{m}_1^{\sigma_1} \mathbf{B}_2^{\sigma_2} \ldots \mathbf{B}_{N-1}^{\sigma_{N-1}} \mathbf{b}_N^{\sigma_N} |\boldsymbol{\sigma}\rangle,$$

- Mixed-Orthogonal canonical form:



$$|\psi\rangle = \sum_{\boldsymbol{\sigma}} \mathbf{a}_1^{\sigma_1} \ldots \mathbf{A}_{\ell-1}^{\sigma_{\ell-1}} \mathbf{M}_{\ell}^{\sigma_\ell} \mathbf{B}_{\ell+1}^{\sigma_{\ell+1}} \ldots \mathbf{b}_N^{\sigma_N} |\boldsymbol{\sigma}\rangle.$$

Usefulness of Canonical Forms

Let's take the inner product of a quantum state with itself, $\langle\psi|\psi\rangle$, as a tensor diagram we have:

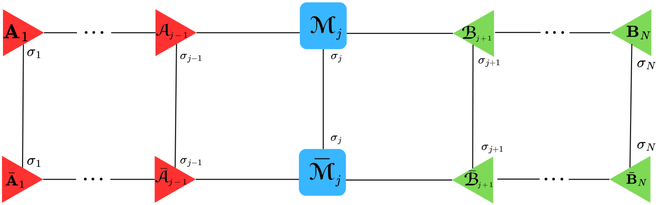


Figure 4: Inner Product of $\langle\psi|\psi\rangle$

Usefulness of Canonical Forms

Let's take the inner product of a quantum state with itself, $\langle\psi|\psi\rangle$, as a tensor diagram we have:

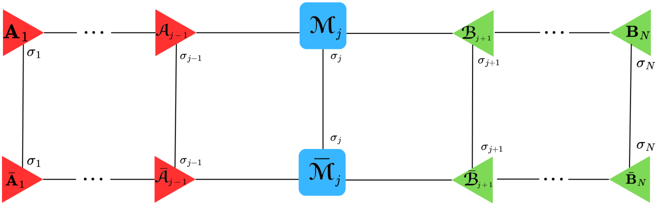


Figure 4: Inner Product of $\langle\psi|\psi\rangle$

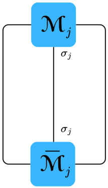
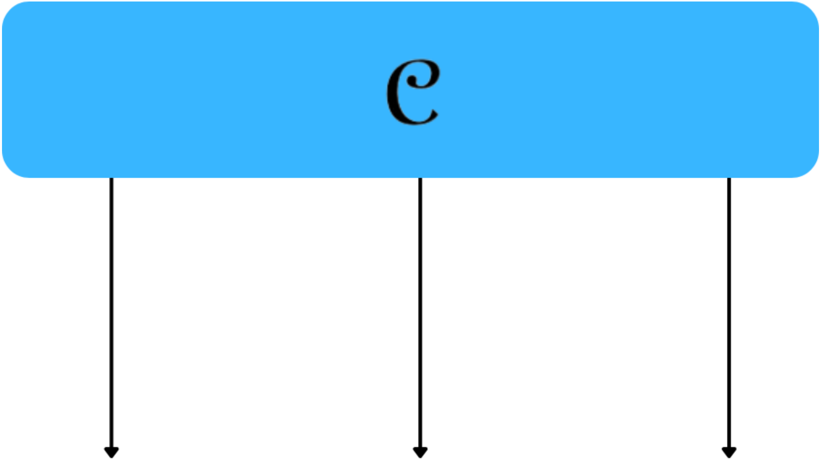


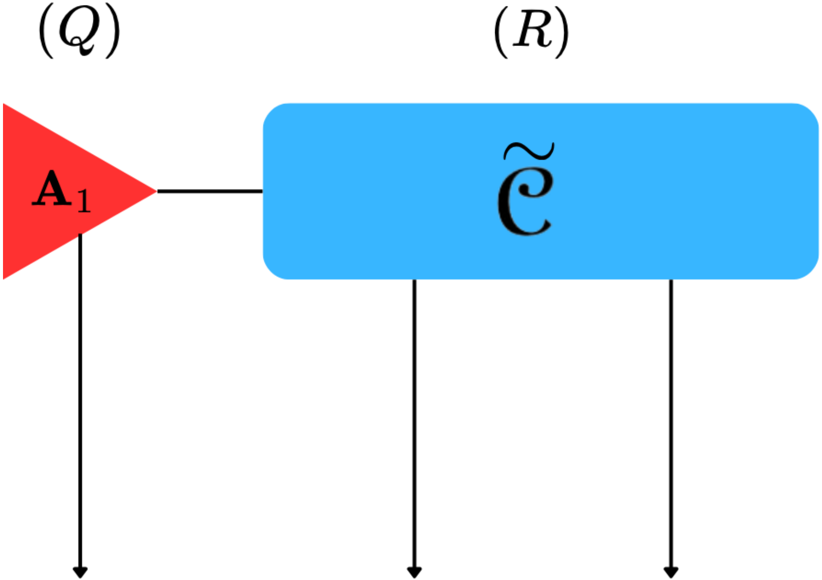
Figure 5: $\langle\psi|\psi\rangle = ||M_j||_F^2$

Calculating the MPS



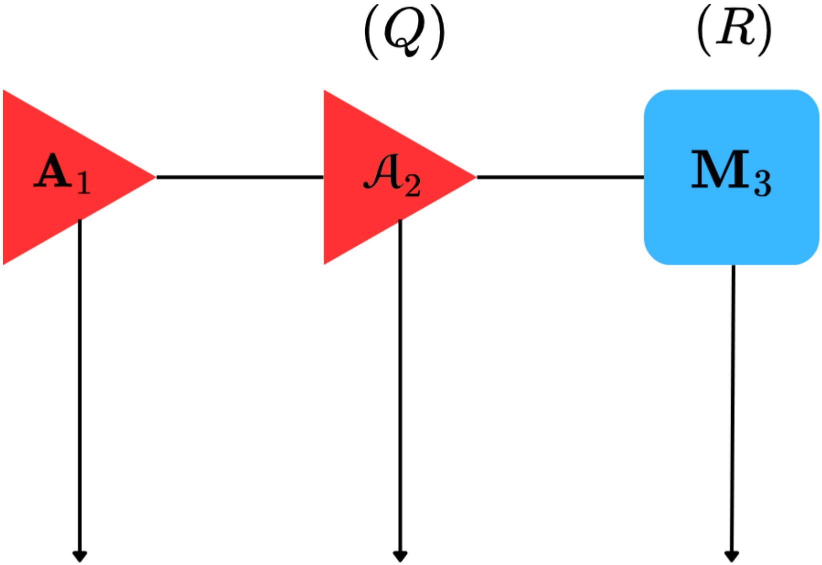
Initial Tensor

Calculating the MPS



Performing QR decomposition

Calculating the MPS



Final State as MPS

Matrix Product State (MPS) Storage

For N qubits:

Vector Storage	**MPS Storage**
Storage exponential with N	Storage linear in N
	May require large bond dimension

SCHOLLWOCK201196

Matrix Product State (MPS) Storage

For N qubits:

Vector Storage	**MPS Storage**
Storage exponential with N	Storage linear in N
	May require large bond dimension
2^N complex numbers stored	$> 2^{N+1}$ complex numbers stored

SCHOLLWOCK201196

Matrix Product State (MPS) Storage

For N qubits:

Vector Storage	**MPS Storage**
Storage exponential with N	Storage linear in N May require large bond dimension
2^N complex numbers stored	$> 2^{N+1}$ complex numbers stored

Key idea: truncate matrix dimensions via SVD

- High entanglement $\Rightarrow$ Larger bond dimension
- Low entanglement $\Rightarrow$ Smaller bond dimension

SCHOLLWOCK201196

MPS Truncation

A separable state (e.g., $|0\rangle \otimes |0\rangle \otimes ... \otimes |0\rangle$ can be represented with much smaller storage:

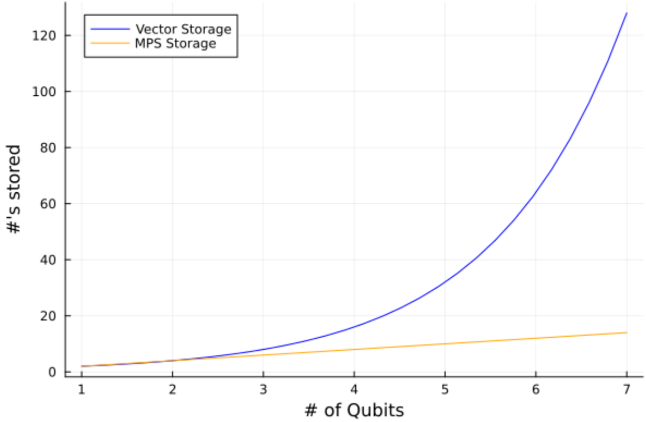


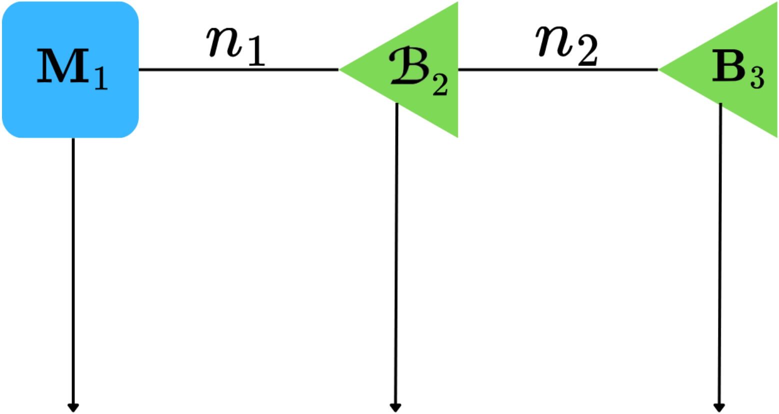
Figure 6: Separable State Storage

Truncating MPS via SVD

- To control bond dimensions, we truncate tensors by performing an SVD of one of the core tensors.
- Truncate by keeping only largest χ singular values: $\lambda_1 \geq \cdots \geq \lambda_\chi$
- Truncation error: $\epsilon_{\text{trunc}} = \sqrt{\sum_{s>\chi} \lambda_s^2}$

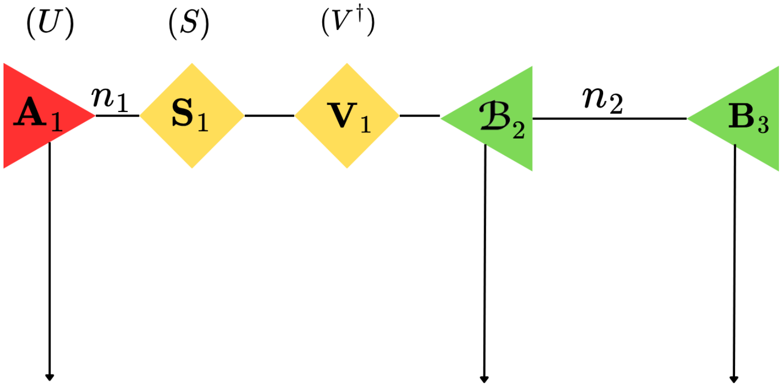
SCHOLLWOCK201196

SVD Truncation



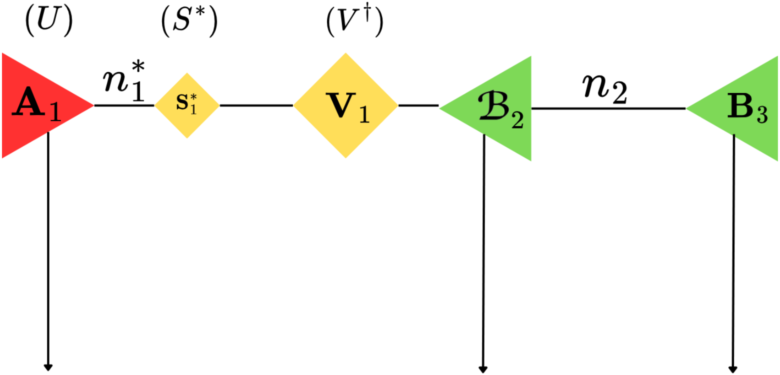
Initial state with bond dimensions n_1, n_2

SVD Truncation



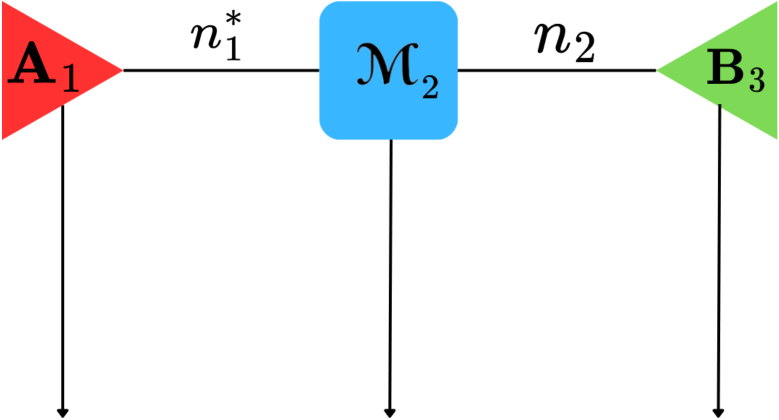
SVD performed on $\mathbf{M}_1$

SVD Truncation



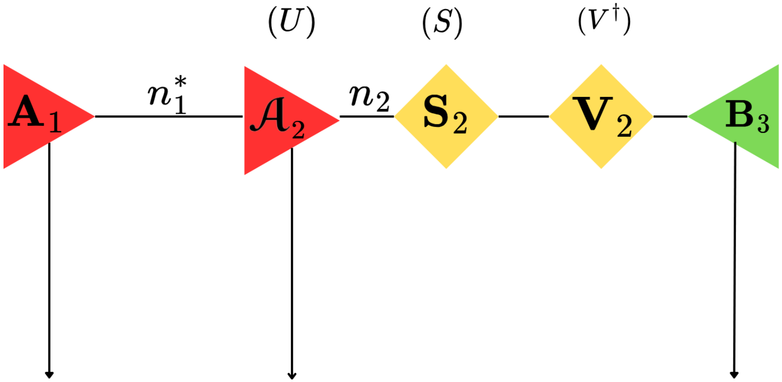
Truncation of Singular value matrix, $\mathbf{S}_1$

SVD Truncation



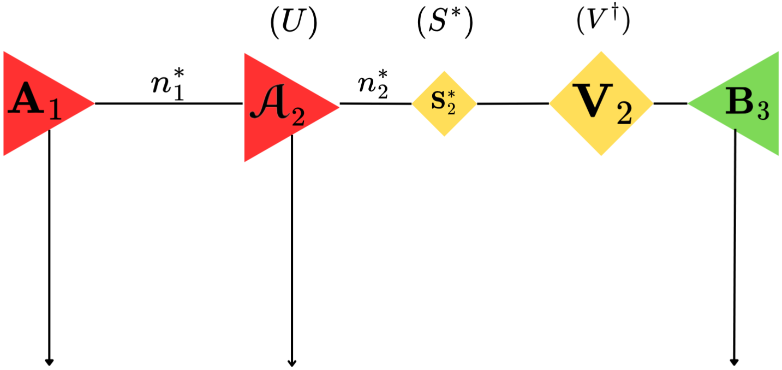
Tensor contraction of $\mathbf{S}_1^*, \mathbf{V}_1, \mathcal{B}_2$

SVD Truncation



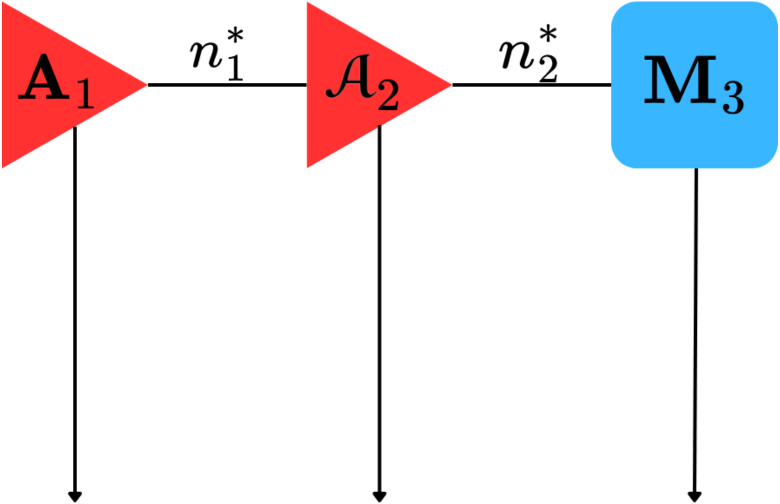
SVD of $\mathcal{M}_2$

SVD Truncation



Truncation of singular value matrix, $\mathbf{S}_2$

SVD Truncation



Tensor contraction of $\mathbf{S}_2^*, \mathbf{V}_2, \mathbf{B}_3$

Matrix Product Operator (MPO)

To evolve an MPS under Schrödinger's equation,

$$|\dot{\psi}\rangle = -iH|\psi\rangle,$$

we express the Hamiltonian H as a Matrix Product Operator (MPO)—a tensor train of order-3 and order-4 tensors. As a tensor diagram:

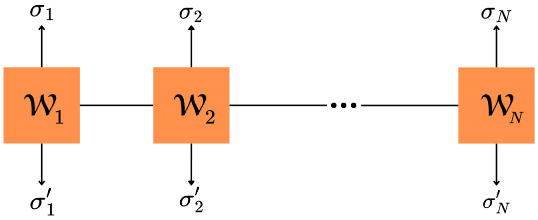


Figure 7: Diagram notation of an MPO

SCHOLLWOCK201196

MPS/MPO Operations and Bond Dimension Growth

Key operations that increase bond dimension:

- **MPS Addition:** Adding two MPS increases bond dimension:

$$\ell_i = m_i + n_i$$

- **MPO Application:** Applying an MPO to an MPS multiplies bond dimensions:

$$\ell_i = w_i \cdot n_i$$

Both operations lead to higher computational cost due to bond dimension growth.

SCHOLLWOCK201196

Table of Contents

MPS Time Evolution Methods

Method	Pros	Cons
TEBD	Easy to implement for local Hamiltonians	Not effective for long-range Hamiltonians
$W^{I,II}$	Allows long-range Hamiltonians, typically smaller bond dimension than TEBD	Evolution not unitary
Krylov Method	High accuracy, flexible for different Hamiltonians	Quickly increases bond dimension
TDVP	Preservation of norm, Fixed Bond Dimension	Fixed Bond Dimension
TDVP2	Allows bond dimension growth, handles entanglement better	More computationally expensive, doesn't necessarily preserve norm

PAECKEL2019167998

TDVP Evolution Picture

The Time-Dependent Variational Principle (TDVP) evolves Schrödinger's equation within the manifold of MPS with fixed bond dimension by projecting onto the tangent space:

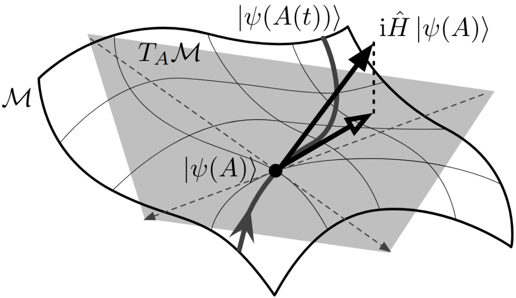


Figure 8: Projection

PhysRevLett.107.070601

TDVP Evolution

$$|\dot{\psi}\rangle = -iP_{T,|\psi\rangle} H|\psi\rangle \tag{4}$$

PhysRevLett.107.070601

TDVP Evolution

$$|\dot{\psi}\rangle = -iP_{T,|\psi\rangle} H|\psi\rangle \tag{4}$$

The projector $P_{T,|\psi\rangle}$ decomposes into a sum of local projectors:

$$P_{T,|\psi\rangle} = \sum_{i=1}^{N} P_i^{+} - \sum_{i=1}^{N-1} P_i^{-}$$

$$P_{T,|\psi\rangle} = P_1^{+} - P_1^{-} + P_2^{+} - P_2^{-} + ... - P_{N-1}^{-} + P_N^{+}$$

PhysRevLett.107.070601

TDVP Evolution

$$|\dot{\psi}\rangle = -iP_{T,|\psi\rangle}H|\psi\rangle \tag{4}$$

The projector $P_{T,|\psi\rangle}$ decomposes into a sum of local projectors:

$$P_{T,|\psi\rangle} = \sum_{i=1}^{N} P_i^+ - \sum_{i=1}^{N-1} P_i^-$$

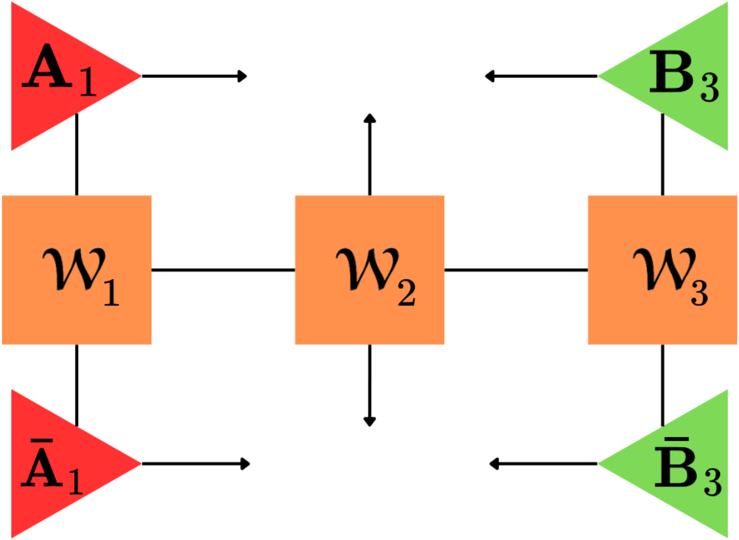
$$P_{T,|\psi\rangle} = P_1^+ - P_1^- + P_2^+ - P_2^- + \ldots - P_{N-1}^- + P_N^+$$

The evolution is performed using a Lie-Trotter splitting scheme, solving one-site and zero-site ODEs:

Projector Step	**Tensor ODE**	**Vectorized Form**
$\lvert\dot{\psi}\rangle = -iP_i^+ H\lvert\psi\rangle$	$\dot{\mathcal{M}}_i = -i\mathcal{H}^i_{\text{eff}}\mathcal{M}_i$	$\dot{\boldsymbol{m}} = -i\mathbf{H}^i_{\text{eff}}\boldsymbol{m}$
$\lvert\dot{\psi}\rangle = iP_i H^- \lvert\psi\rangle$	$\dot{\mathrm{C}}_i = i\mathcal{K}^i_{\text{eff}}\mathrm{C}_i$	$\dot{\boldsymbol{c}} = i\mathbf{K}^i_{\text{eff}}\boldsymbol{c}$

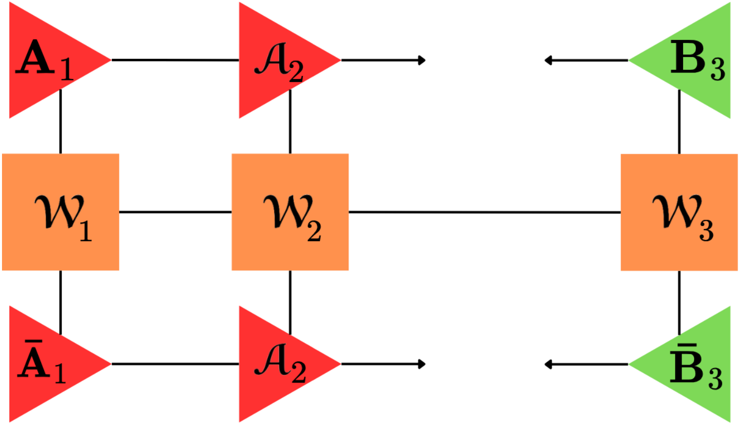
PhysRevLett.107.070601

Effective Hamiltonian



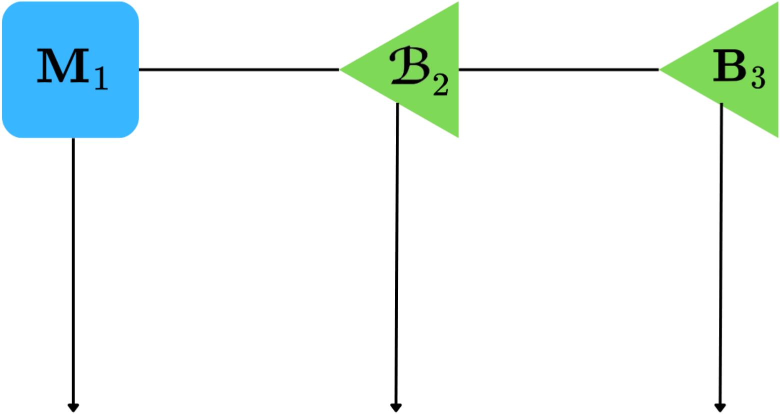
Tensor Diagram of $\mathcal{H}^2_{\text{eff}}$.

Effective Hamiltonian

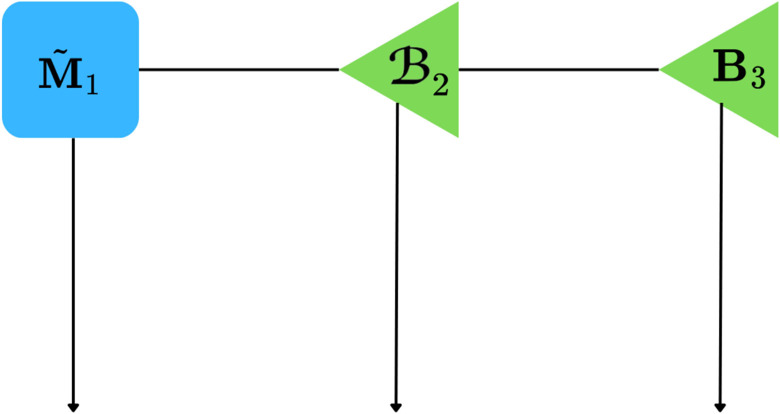


Tensor Diagram of $\mathcal{K}_{\text{eff}}^2$.

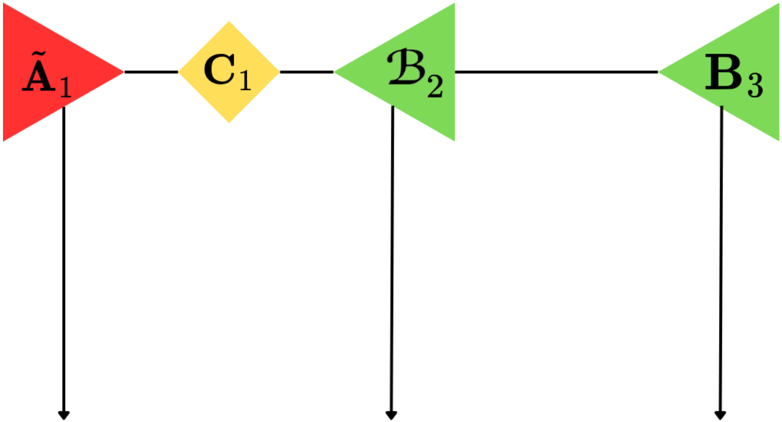
TDVP Evolution Tensor Diagram



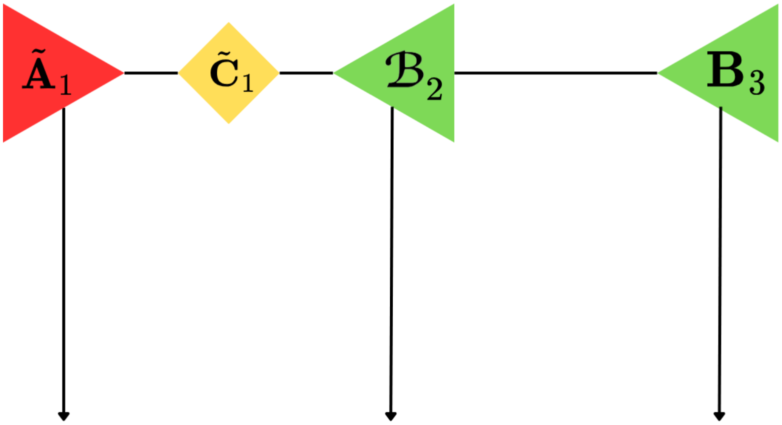
TDVP Evolution Tensor Diagram



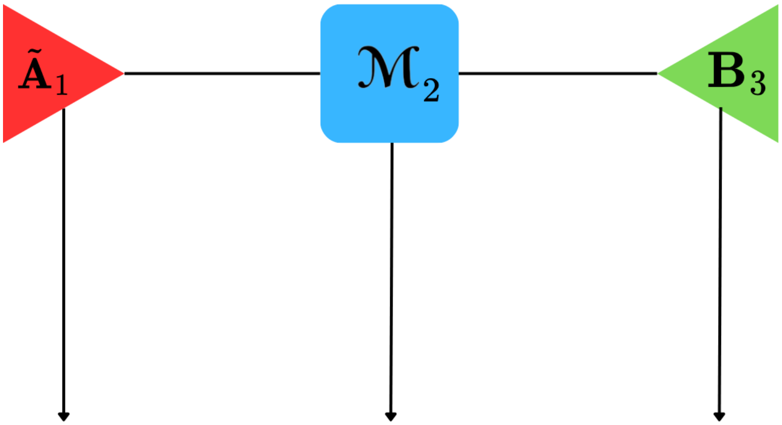
TDVP Evolution Tensor Diagram



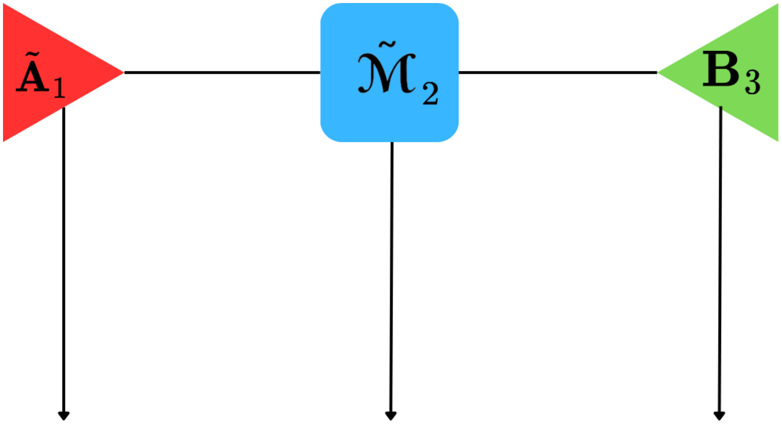
TDVP Evolution Tensor Diagram



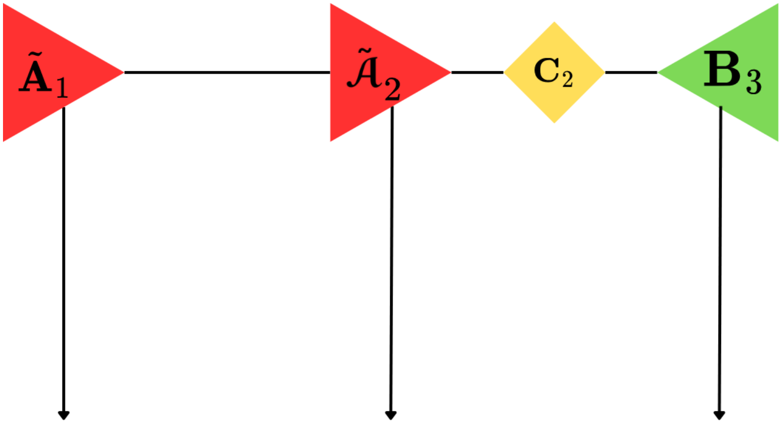
TDVP Evolution Tensor Diagram



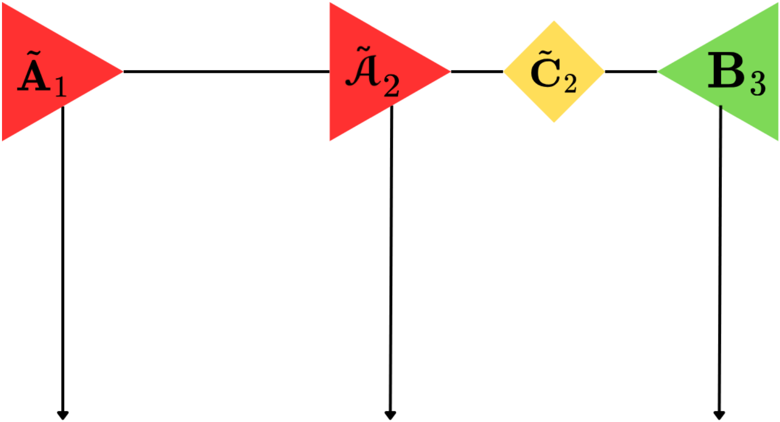
TDVP Evolution Tensor Diagram



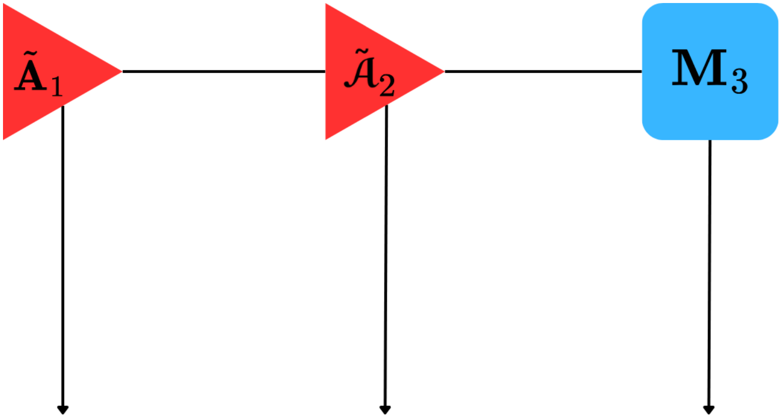
TDVP Evolution Tensor Diagram



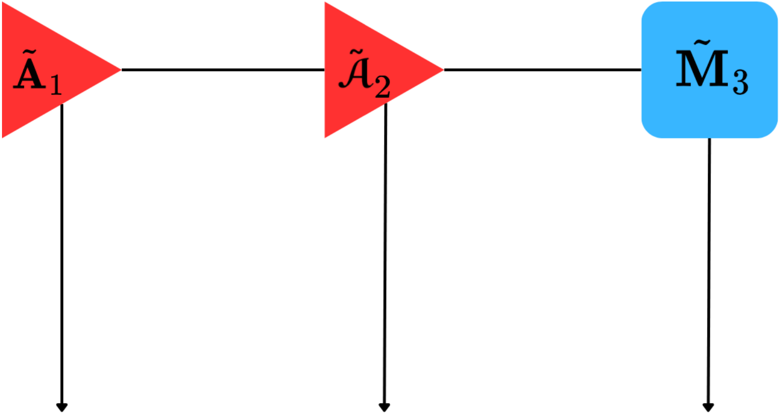
TDVP Evolution Tensor Diagram



TDVP Evolution Tensor Diagram



TDVP Evolution Tensor Diagram



TDVP Evolution Example: CNOT gate

The CNOT quantum gate (short for Controlled NOT), swaps the states $|10\rangle$ and $|11\rangle$, and leaves the states $|00\rangle$ and $|01\rangle$ unchanged.

$$CNOT|10\rangle = |11\rangle, \quad CNOT|11\rangle = |10\rangle.$$

TDVP Evolution Example: CNOT gate

The CNOT quantum gate (short for Controlled NOT), swaps the states $|10\rangle$ and $|11\rangle$, and leaves the states $|00\rangle$ and $|01\rangle$ unchanged.

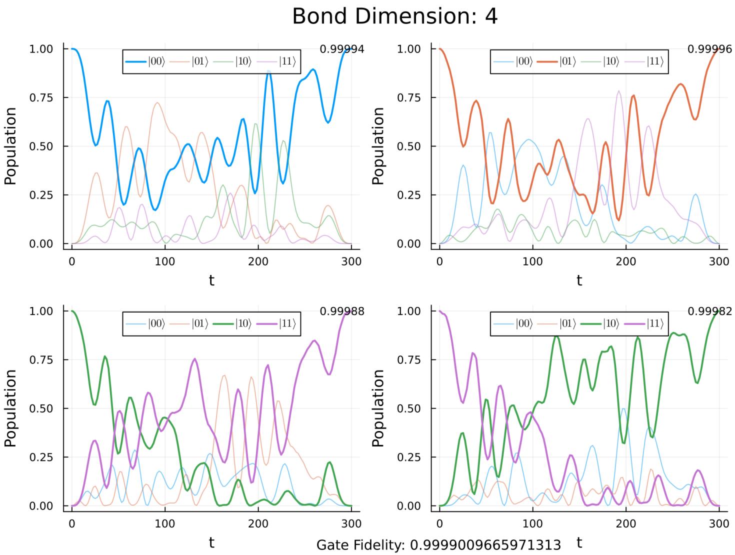
$$CNOT|10\rangle = |11\rangle, \quad CNOT|11\rangle = |10\rangle.$$

As a matrix:

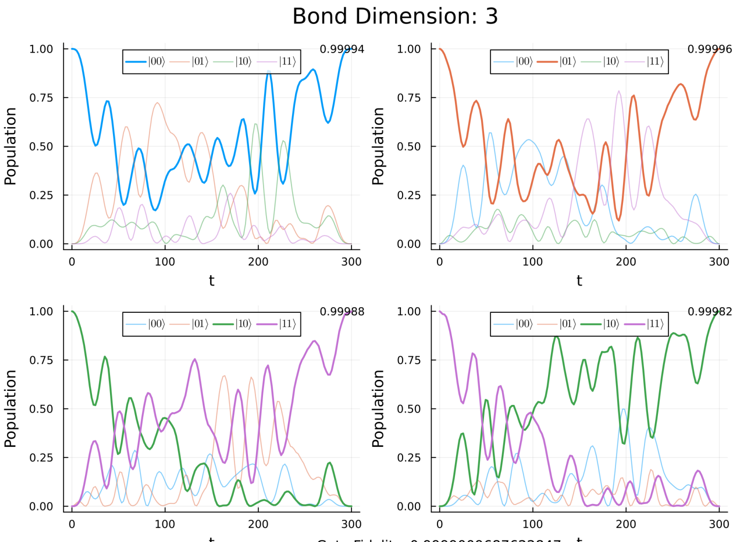
$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

The CNOT gate control pulses were computed using the quantum control package *quandary*.

TDVP Evolution Example: CNOT gate

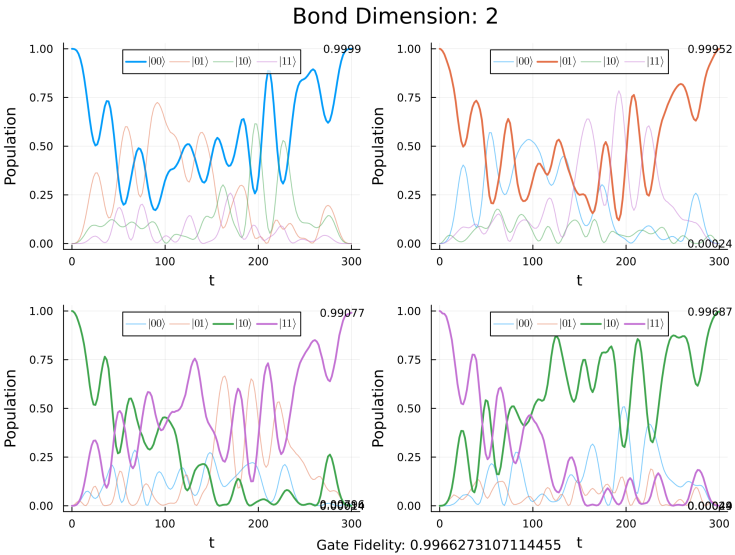


TDVP Evolution Example: CNOT gate

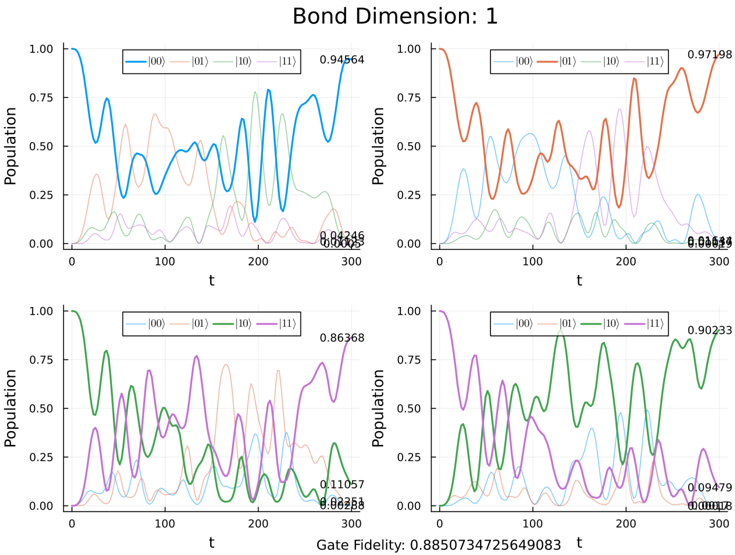


Gate Fidelity: 0.9999009687622847

TDVP Evolution Example: CNOT gate



TDVP Evolution Example: CNOT gate



2 Site TDVP Evolution

$$|\dot{\psi}\rangle = -iP_{T_2,|\psi\rangle} H|\psi\rangle \tag{5}$$

PhysRevLett.107.070601

2 Site TDVP Evolution

$$|\dot{\psi}\rangle = -iP_{T_2,|\psi\rangle} H|\psi\rangle \tag{5}$$

2 site TDVP Evolution (TDVP2) evolves Schrödinger's equation, but by evolving two sites at one time, then splitting them up via SVD. This allows bond dimension growth.

The projector $P_{T_2,|\psi\rangle}$ is a sum of projectors on individual sites of the MPS.

$$\begin{aligned} P_{T_2,|\psi\rangle} &= \sum_{i=1}^{N-1} P_i^+ - \sum_{i=2}^{N-2} P_i^-, \\ P_{T_2,|\psi\rangle} &= P_1^+ - P_2^- + P_2^+ - P_3^- + \ldots - P_{N-2}^- + P_{N-1}^+. \end{aligned} \tag{6}$$

PhysRevLett.107.070601

2 Site TDVP Evolution

$$|\dot{\psi}\rangle = -iP_{T_2,|\psi\rangle} H|\psi\rangle \tag{5}$$

2 site TDVP Evolution (TDVP2) evolves Schrödinger's equation, but by evolving two sites at one time, then splitting them up via SVD. This allows bond dimension growth.
The projector $P_{T_2,|\psi\rangle}$ is a sum of projectors on individual sites of the MPS.

$$P_{T_2,|\psi\rangle} = \sum_{i=1}^{N-1} P_i^+ - \sum_{i=2}^{N-2} P_i^-,$$
$$P_{T_2,|\psi\rangle} = P_1^+ - P_2^- + P_2^+ - P_3^- + \dots - P_{N-2}^- + P_{N-1}^+. \tag{6}$$

The evolution is performed using a Lie-Trotter splitting scheme, solving two-site and one-site ODEs:

Projector Step	**Tensor ODE**	**Vectorized Form**
$\lvert\dot{\psi}\rangle = -iP_i^+ H\lvert\psi\rangle$	$\dot{\mathcal{M}}_{i,i+1} = -i\mathcal{H}_{\text{eff}}^{i,i+1}\mathcal{M}_{i,i+1}$	$\dot{\boldsymbol{m}} = -i\mathbf{H}_{\text{eff}}^{i,i+1}\boldsymbol{m}$
$\lvert\dot{\psi}\rangle = iP_i^- H\lvert\psi\rangle$	$\dot{\mathcal{M}}_i = i\mathcal{H}_{\text{eff}}^i\mathcal{M}_i$	$\dot{\boldsymbol{m}} = i\mathbf{H}_{\text{eff}}^i\boldsymbol{m}$

PhysRevLett.107.070601

2 site Effective Hamiltonian

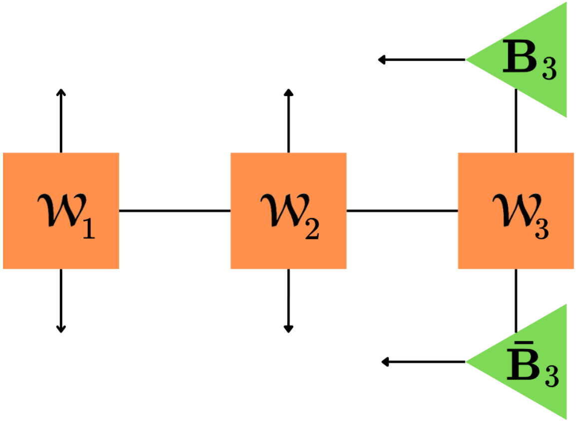
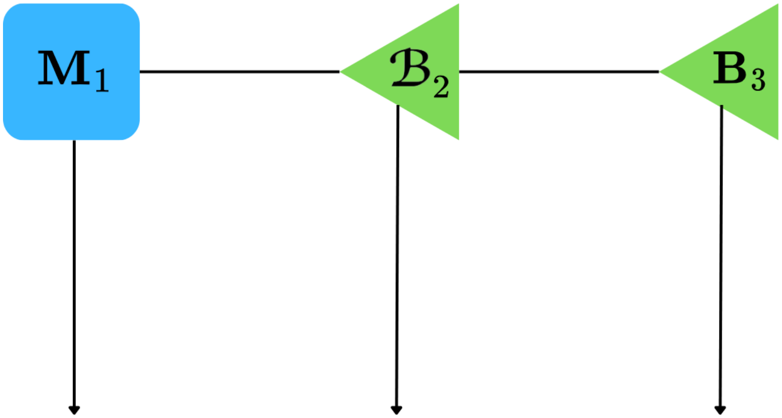
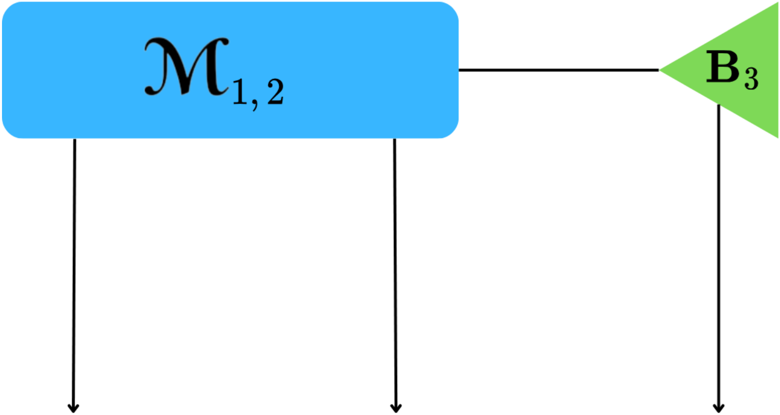


Figure 9: Tensor Diagram of $\mathcal{H}_{\text{eff}}^{1,2}$.

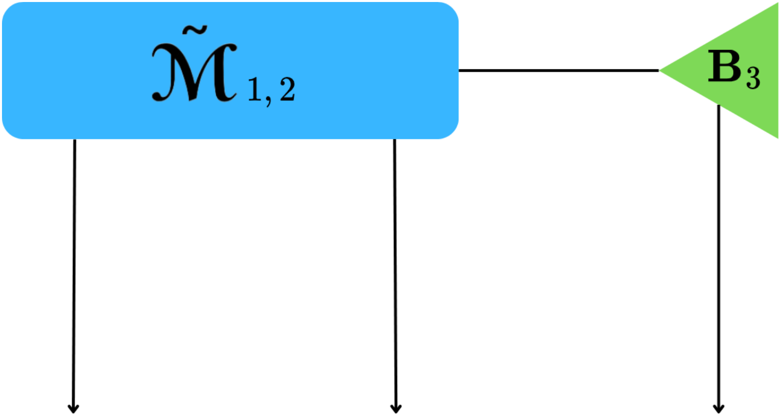
TDVP2 Evolution Tensor Diagram



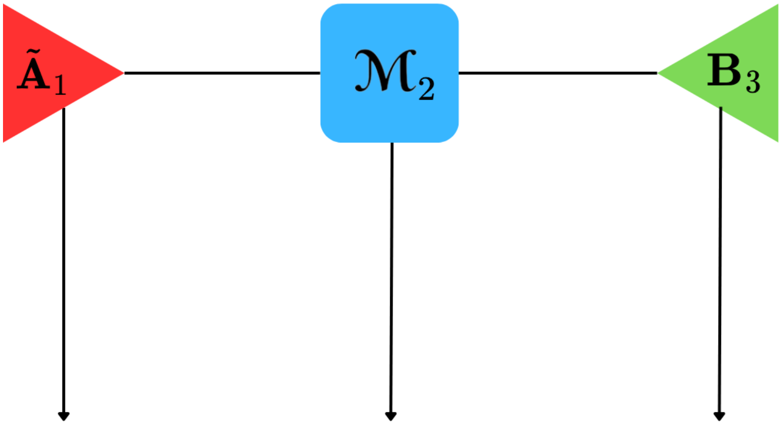
TDVP2 Evolution Tensor Diagram



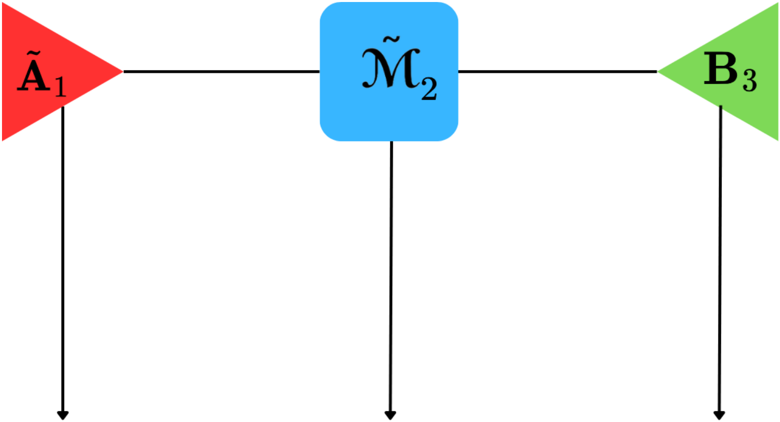
TDVP2 Evolution Tensor Diagram



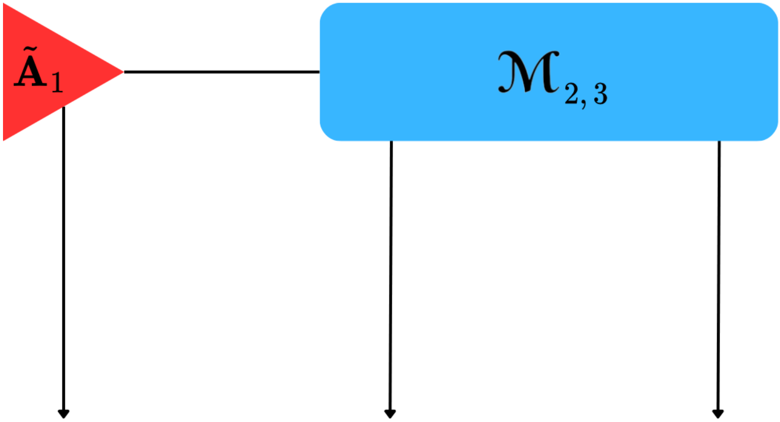
TDVP2 Evolution Tensor Diagram



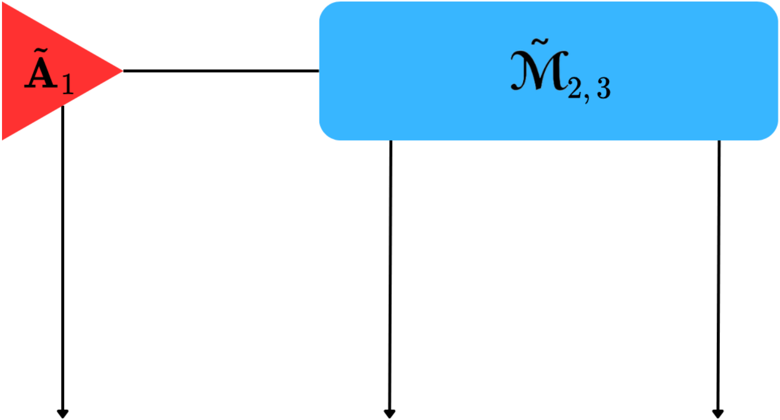
TDVP2 Evolution Tensor Diagram



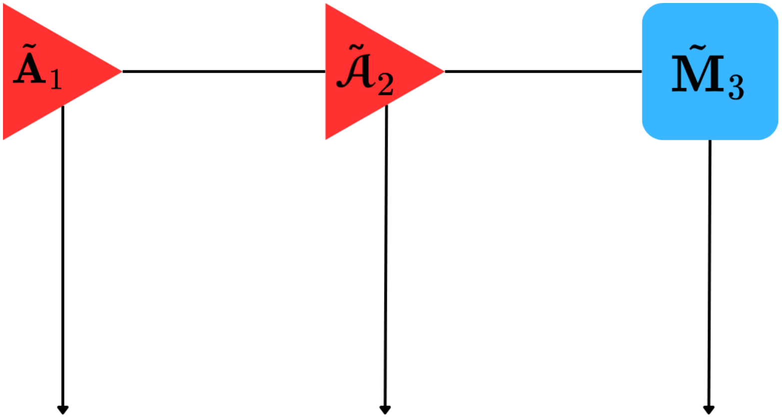
TDVP2 Evolution Tensor Diagram



TDVP2 Evolution Tensor Diagram



TDVP2 Evolution Tensor Diagram



TDVP2 Evolution

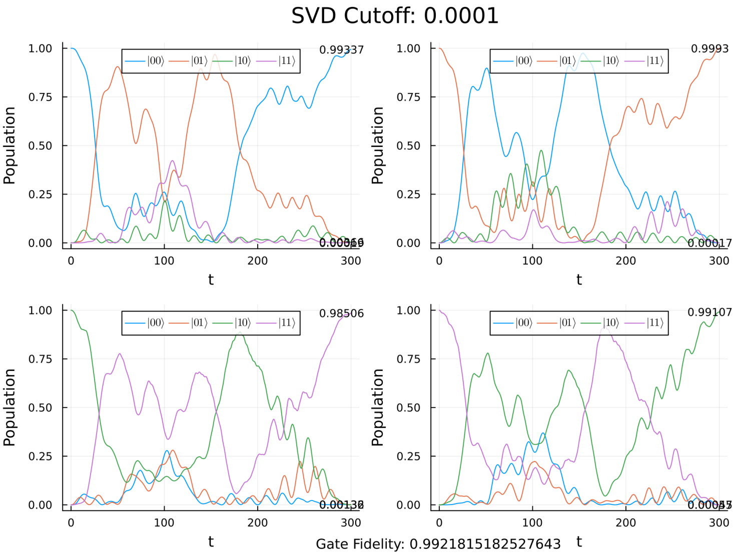


Figure 10: TDVP2 Evolution with SVD cutoff of 1E-4

TDVP2 Evolution

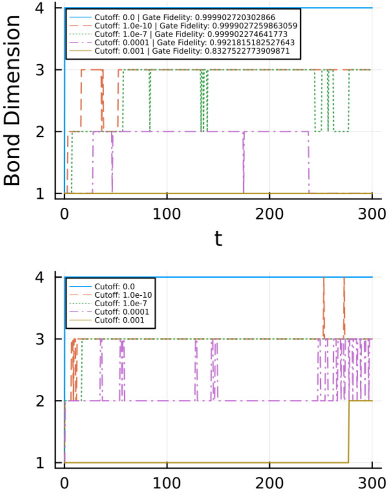


Figure 11: Change in bond dimension for different SVD cutoff values for TDVP2.

TDVP Error

Three Sources of Error:

- Projection Error

$$||H|\psi\rangle - P_{T,|\psi\rangle} H|\psi\rangle||_2$$

- Splitting Error from ODE splitting Method
- Truncation Error (for TDVP2)

PhysRevLett.107.070601

Table of Contents

Develop Method for continuous time Hamiltonian

We solve the equation

$$|\dot{\psi}\rangle = -iH(t)|\psi\rangle,$$

with

$$H(t) = H_s + H_c(t).$$

Develop Method for continuous time Hamiltonian

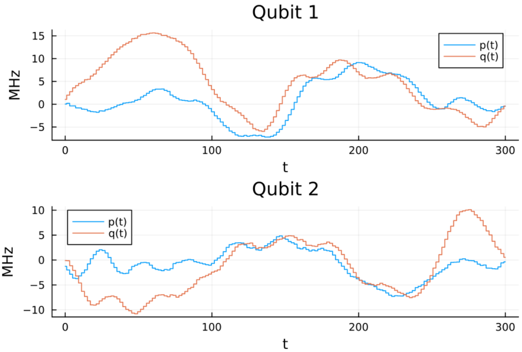
We solve the equation

$$|\dot{\psi}\rangle = -iH(t)|\psi\rangle,$$

with

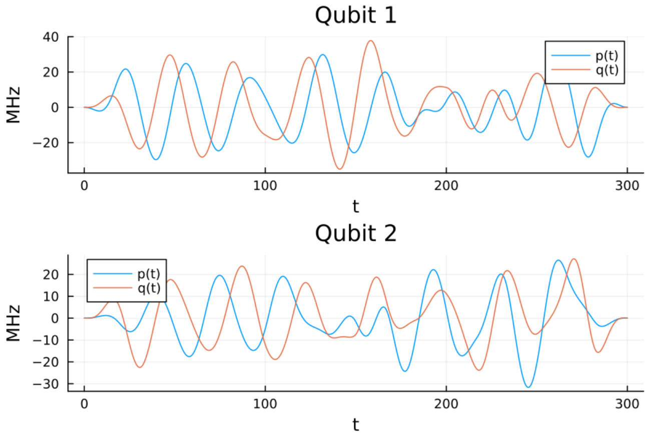
$$H(t) = H_s + H_c(t).$$

The pulses implemented for the CNOT gate were done as piecewise constant pulses.



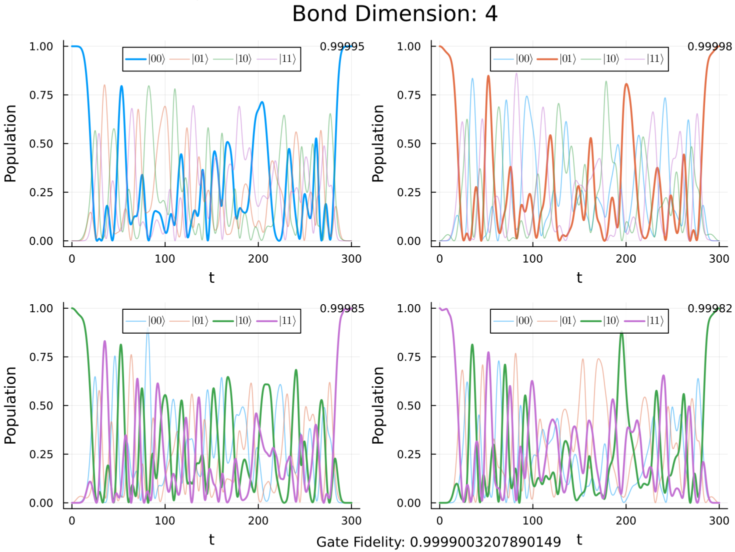
Develop Method for continuous time Hamiltonian

Additionally it is valuable to develop a method that doesn't work for just pulses that aren't just piecewise-constant, for example:



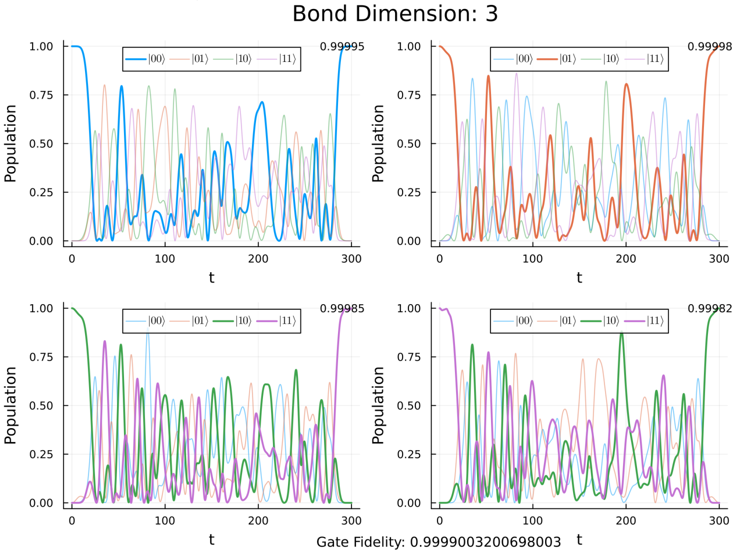
Continuous time Hamiltonian Method

Simulation using b-splines



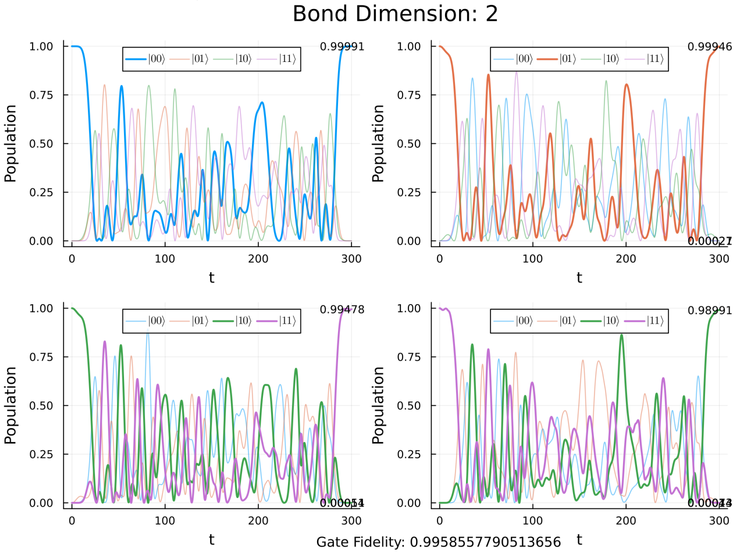
Continuous time Hamiltonian Method

Simulation using b-splines



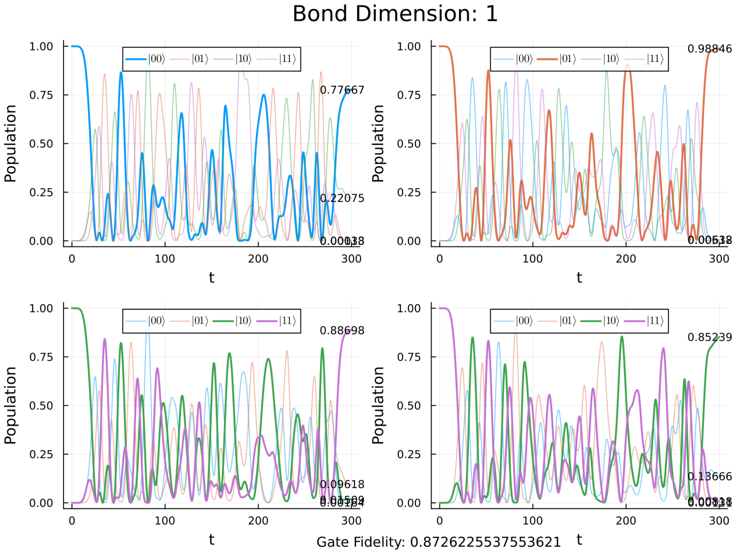
Continuous time Hamiltonian Method

Simulation using b-splines



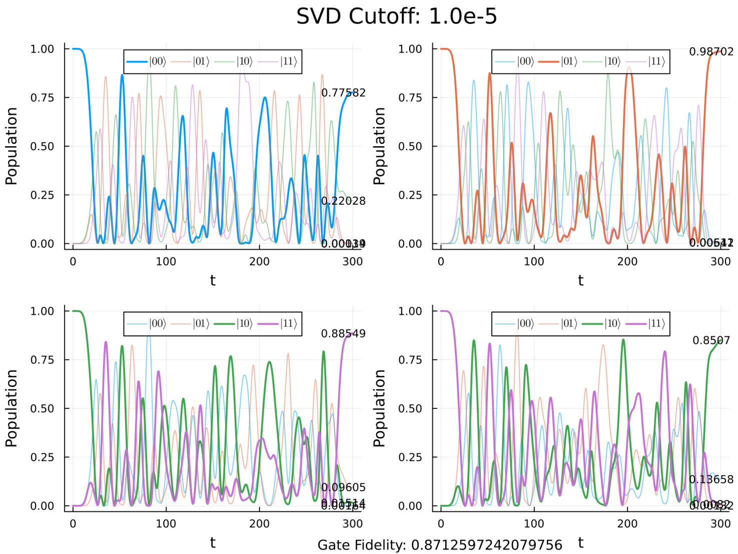
Continuous time Hamiltonian Method

Simulation using b-splines



Develop Method for Continuous time Hamiltonian

We can also use the TDVP2 method with this pulse, still just using an exponential solver.



Combine time-evolution with Optimal Control

The pulses shown in previous slide were obtained from quandary.

- Gradient Information obtained by evolving

$$\begin{cases} \dot{\boldsymbol{y}} = A(t)\boldsymbol{y} \\ \boldsymbol{y}(0) = \boldsymbol{\eta} \end{cases},$$

$$\begin{cases} -\dot{\boldsymbol{\lambda}} - A(t)^{\dagger}\boldsymbol{\lambda} = \boldsymbol{a}(t) \\ \boldsymbol{\lambda}(T) = \boldsymbol{w}. \end{cases}.$$

Combine time-evolution with Optimal Control

The pulses shown in previous slide were obtained from quandary.

- Gradient Information obtained by evolving

$$\begin{cases} \dot{\boldsymbol{y}} = A(t)\boldsymbol{y} \\ \boldsymbol{y}(0) = \boldsymbol{\eta} \end{cases},$$

$$\begin{cases} -\dot{\boldsymbol{\lambda}} - A(t)^\dagger \boldsymbol{\lambda} = \boldsymbol{a}(t) \\ \boldsymbol{\lambda}(T) = \boldsymbol{w}. \end{cases}.$$

- Quandary evolves the equation using the Implicit Midpoint Rule.

Combine time-evolution with Optimal Control

The pulses shown in previous slide were obtained from quandary.

- Gradient Information obtained by evolving

$$\begin{cases} \dot{\boldsymbol{y}} = A(t)\boldsymbol{y} \\ \boldsymbol{y}(0) = \boldsymbol{\eta} \end{cases},$$

$$\begin{cases} -\dot{\boldsymbol{\lambda}} - A(t)^\dagger \boldsymbol{\lambda} = \boldsymbol{a}(t) \\ \boldsymbol{\lambda}(T) = \boldsymbol{w}. \end{cases}.$$

- Quandary evolves the equation using the Implicit Midpoint Rule.
- For a large numbers of qubits storing vector becomes impossible.

Combine time-evolution with Optimal Control

The pulses shown in previous slide were obtained from quandary.

- Gradient Information obtained by evolving

$$\begin{cases} \dot{\boldsymbol{y}} = A(t)\boldsymbol{y} \\ \boldsymbol{y}(0) = \boldsymbol{\eta} \end{cases},$$

$$\begin{cases} -\dot{\boldsymbol{\lambda}} - A(t)^\dagger \boldsymbol{\lambda} = \boldsymbol{a}(t) \\ \boldsymbol{\lambda}(T) = \boldsymbol{w}. \end{cases}.$$

- Quandary evolves the equation using the Implicit Midpoint Rule.
- For a large numbers of qubits storing vector becomes impossible.
- Evolving forward using MPS can reduce the storage requirements of this evolution.

Beyond Dissertation

- Schrodinger's equation evolves a quantum state in a closed system.

Beyond Dissertation

- Schrodinger's equation evolves a quantum state in a closed system.
- An open quantum system allows for interaction with the environment.

Beyond Dissertation

- Schrodinger's equation evolves a quantum state in a closed system.
- An open quantum system allows for interaction with the environment.
- An example model of an open quantum system is the **Lindblad Master Equation**,

$$\dot{\rho} = -i(H\rho - \rho H) + \sum_j \gamma_j L_j \rho L_j^\dagger - \frac{1}{2}\left(L_j^\dagger L_j \rho + \rho L_j^\dagger L_j\right)$$

Beyond Dissertation

- Schrodinger's equation evolves a quantum state in a closed system.
- An open quantum system allows for interaction with the environment.
- An example model of an open quantum system is the **Lindblad Master Equation**,

$$\dot{\rho} = -i(H\rho - \rho H) + \sum_j \gamma_j L_j \rho L_j^\dagger - \frac{1}{2}\left(L_j^\dagger L_j \rho + \rho L_j^\dagger L_j\right)$$

- This equation evolves a matrix ρ as opposed to a vector.

References I