PASQAL

Anton Quelle

CINECA

Practical Quantum Computing 2022

Efficient Quantum simulation using Tensor Network states

Contents

- Quantum complexity
- Tensors
- Tensor Network representation
- Contraction order
- Example: Hardware efficient ansatz QVC
- If time permits: Controlling bond dimension

PASQAL

Quantum complexity

$$|\psi\rangle_1 = \sum_i A_{i_1} |i_1\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle =$$

$$\alpha_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \alpha_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}$$

Qbits	Type	Parameters
1	A_i	2

PASQAL

Quantum complexity

$$|\psi\rangle_1 = \sum_i A_{i_1}|i_1\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$$

$$|\psi\rangle_2 = \sum_i A_{i_1 i_2}|i_1\, i_2\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

$$\alpha_0 \begin{pmatrix}1\\0\\0\\0\end{pmatrix} + \alpha_1 \begin{pmatrix}0\\1\\0\\0\end{pmatrix} + \alpha_2 \begin{pmatrix}0\\0\\1\\0\end{pmatrix} + \alpha_3 \begin{pmatrix}0\\0\\0\\1\end{pmatrix} = \begin{pmatrix}\alpha_0\\\alpha_1\\\alpha_2\\\alpha_3\end{pmatrix}$$

Qbits	Type	Parameters
1	A_i	2
2	A_{ij}	4

α_0	α_1
α_2	α_3

The same information can be represented in vector or matrix form

PASQAL

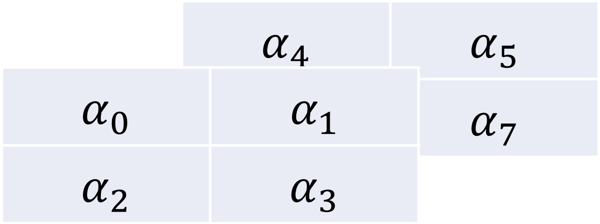
Quantum complexity

$$|\psi\rangle_1 = \sum_i A_{i_1}|i_1\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$$

$$|\psi\rangle_2 = \sum_i A_{i_1 i_2}|i_1\, i_2\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

$$|\psi\rangle_N = \sum_i A_{i_1 \dots i_N}|i_1 \dots i_N\rangle$$

Qbits	Type	Parameters
1	A_i	2
2	A_{ij}	4
3	A_{ijk}	8
N	$A_{i_1 \dots i_N}$	2^N



For 3 qubits the quantum state can be encoded as a rank 3 tensor

PASQAL

Quantum complexity

$$|\psi\rangle_1 = \sum_i A_{i_1} |i_1\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$$

$$|\psi\rangle_2 = \sum_i A_{i_1 i_2} |i_1\, i_2\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

$$|\psi\rangle_N = \sum_i A_{i_1 \dots i_N} |i_1 \dots i_N\rangle$$

Qbits	Type	Parameters
1	A_i	2
2	A_{ij}	4
3	A_{ijk}	8
N	$A_{i_1 \dots i_N}$	2^N

$A_{i_1 \dots i_N}$ is an object with N indices, i.e. A rank N tensor

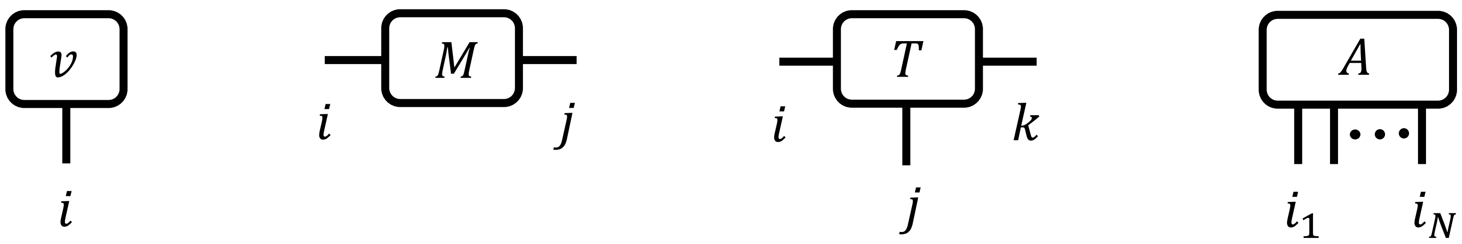
\# parameter for N = 300 is $\approx 10^{90}$
\# of particles in the observable universe *vigintillion* $\approx 10^{80}$

Array of coefficients for 40 qbits requires 1 Tb of memory.
How to store it in a computer ?

PASQAL

Tensors

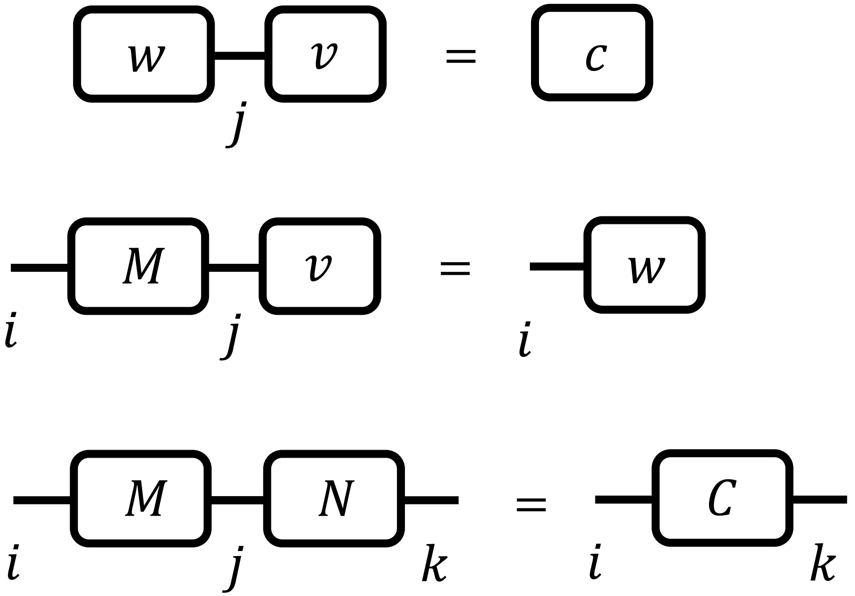
vector | matrix | 3d tensor | N d tensor

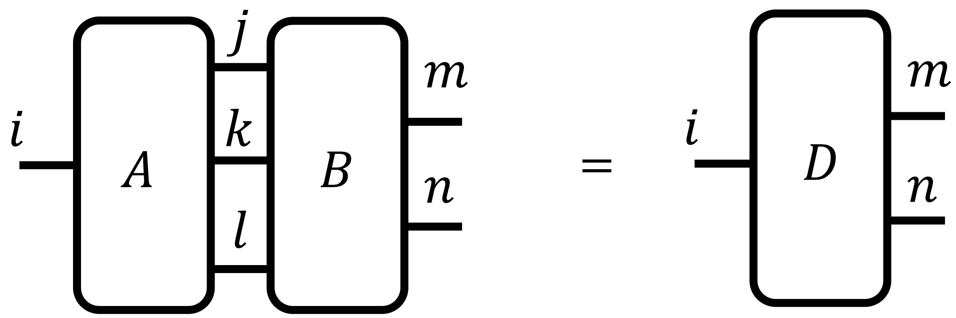


$$\sum_j w_j \, v_j =$$

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

$$\sum_j M_{ij} \, N_{jk} =$$



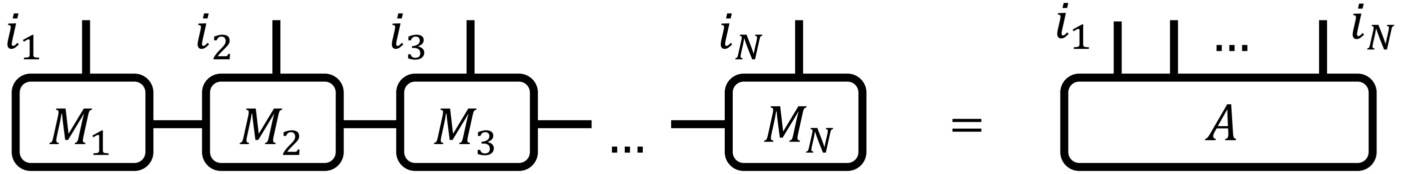


PASQAL

Tensor Network representation

We represent a high rank tensor as a product of low rank tensors.

$$|\psi\rangle_N = \sum_i A_{i_1 \ldots i_N} |i_1 \ldots i_N\rangle$$



Matrix Product State/ Tensor Train

Assume each index of A has dim d, then the rank of $M_i <= d^{N/2}$
For relevant states, the rank of M_d is usually significantly smaller

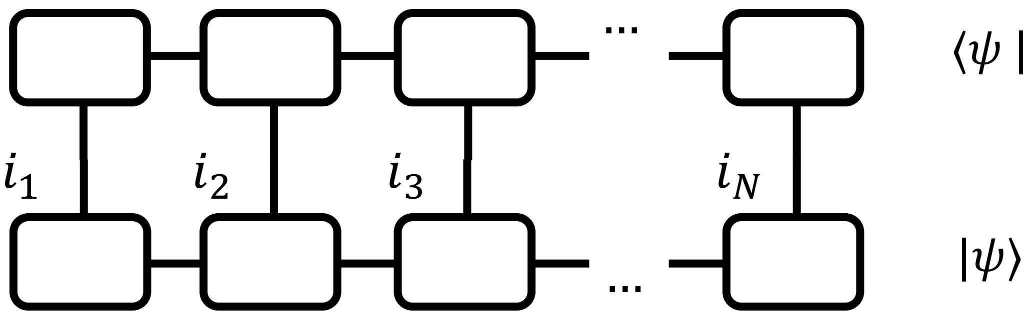
Let rank $M_i = m$ for convenience, then the tensor network is defined by $2Nm^2$ for a 2 level system.

Matrix Product State(MPS)/ Tensor Train

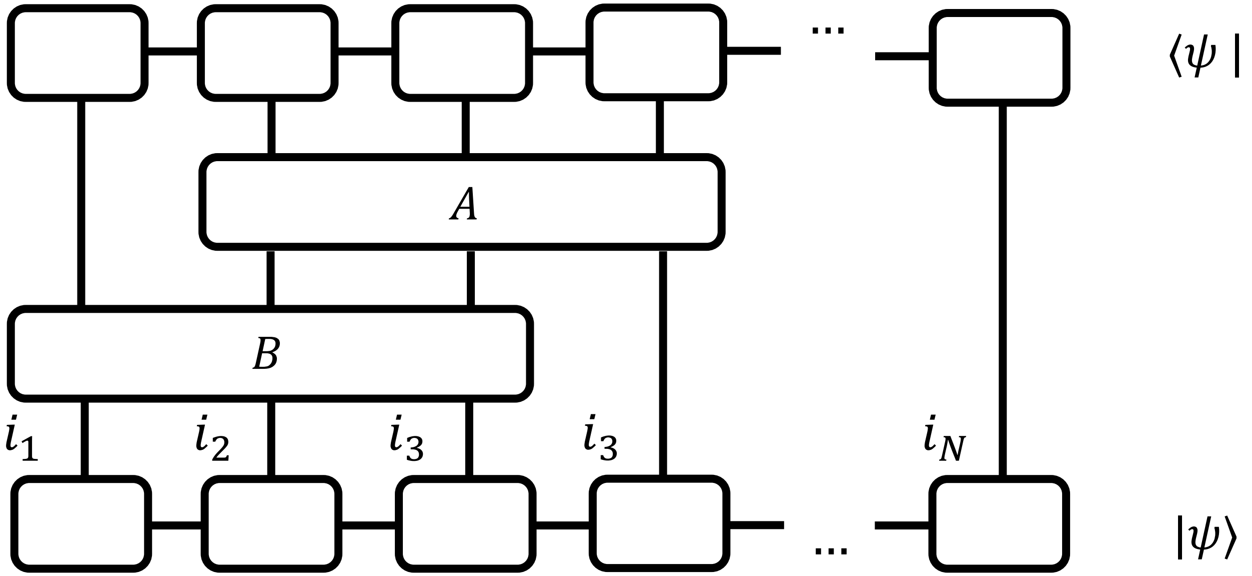
The density-matrix renormalization group
U. Schollwöck Rev. Mod. Phys. **77**, 259

Tensor Network representation

$\langle \psi | \psi \rangle =$

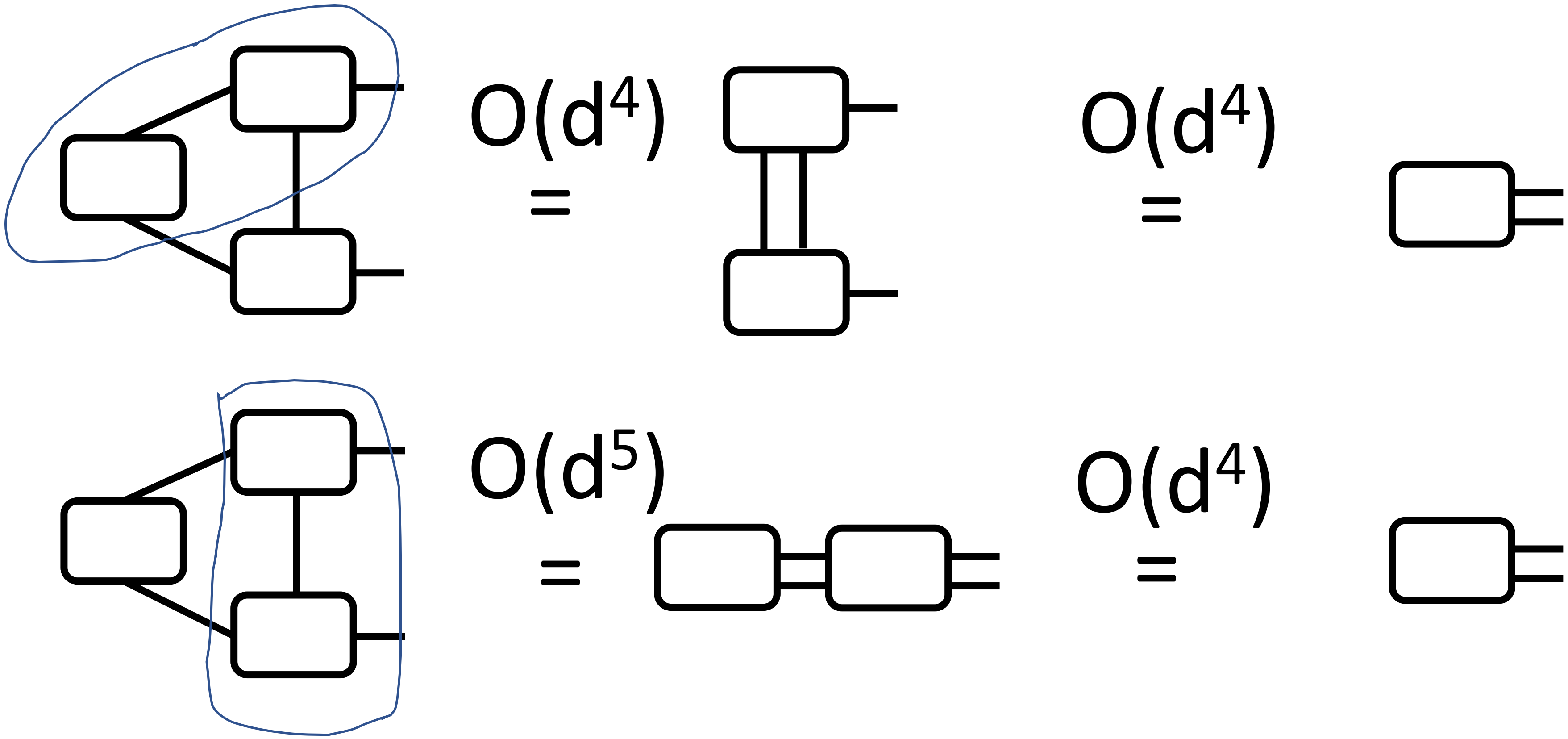


$\langle \psi | AB | \psi \rangle =$

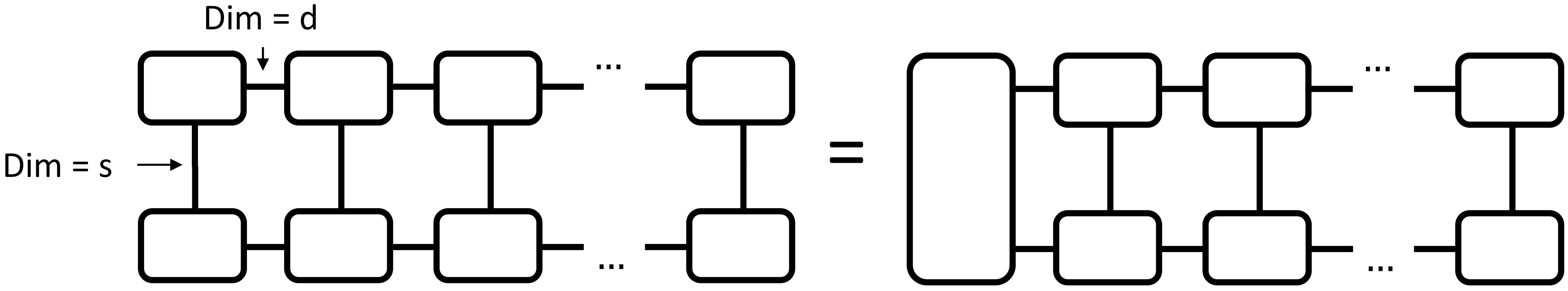


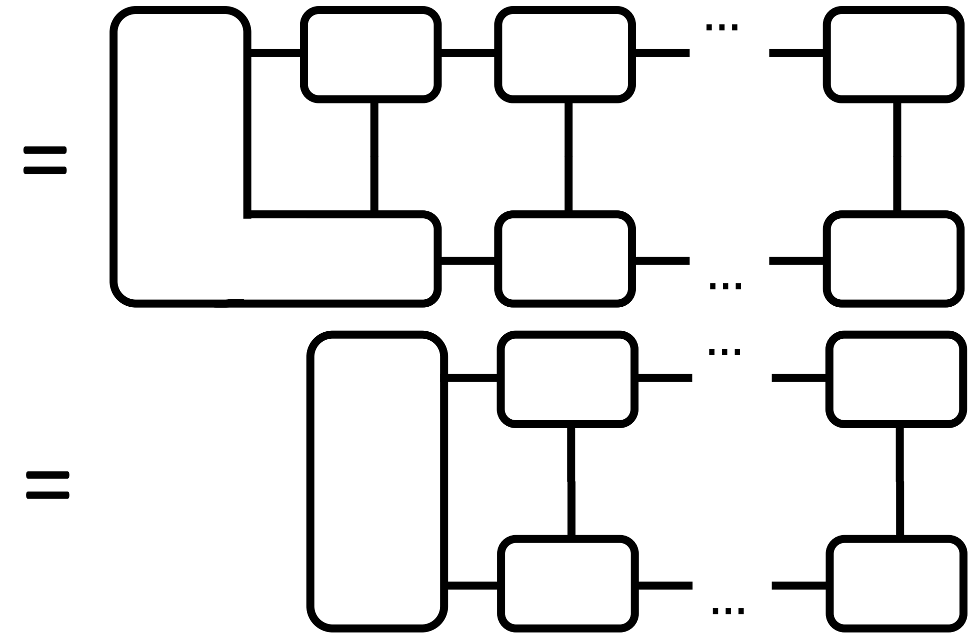
Contraction order

The algorithmic complexity of a tensor network contraction depends on the contraction order!

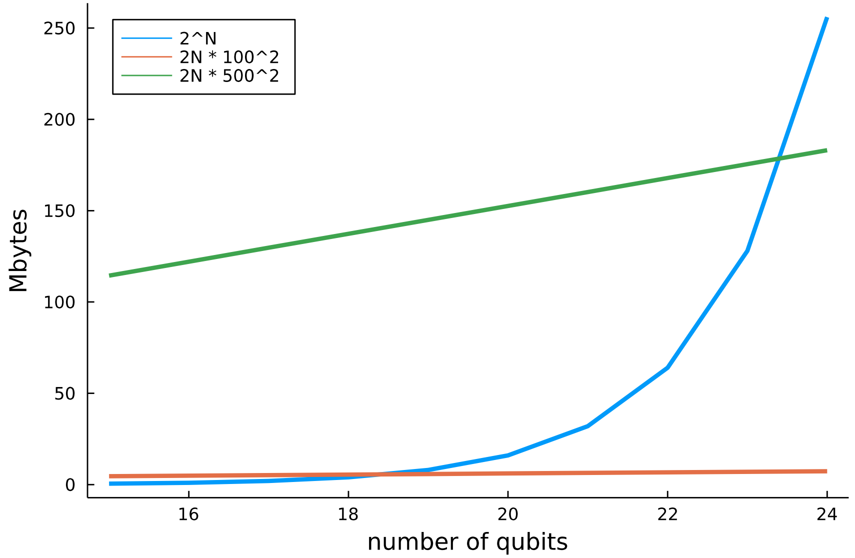


Contraction order





$O(sd^3)$

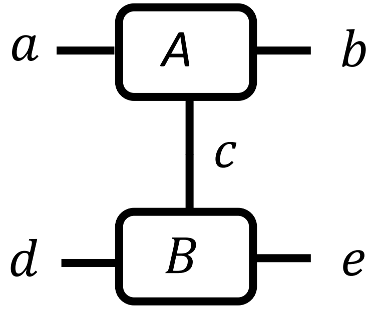


MPS networks scales linearly $\approx O(N)$
d depends on a quantum system

Example: Hardware efficient ansatz QVC

CNOT gate is an MPO of bond dimension 2

CNOT =

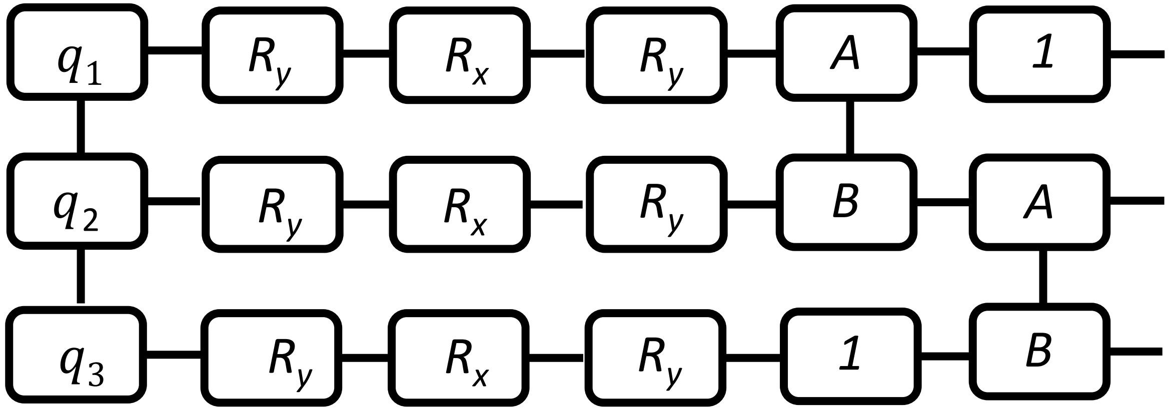


A and *B* both have dims (2,2,2)
The non-zero elements are, with indices in alphabetical order
$A_{000} = A_{111} = B_{000} = B_{011} = B_{101} = C_{110} = 1$

Single qubit gates are MPO's of bond dimension 1 in a trivial way

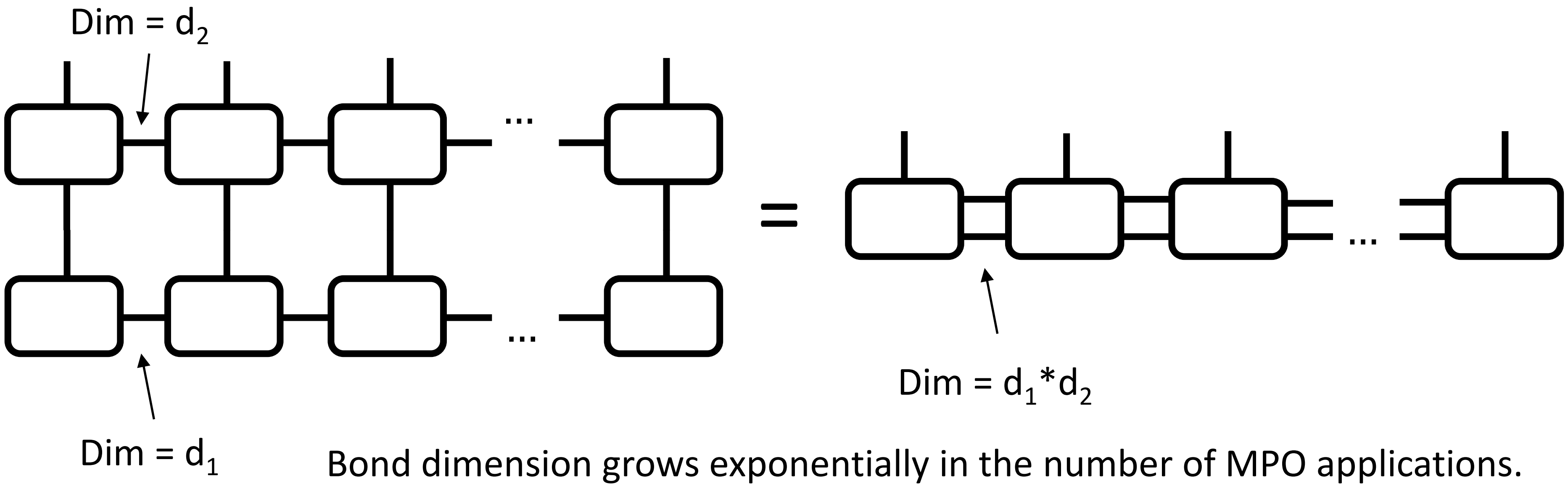
Example: Hardware efficient ansatz QVC

3 qubit hardware-efficient ansatz as a tensor network



PASQAL

Controlling bond dimension

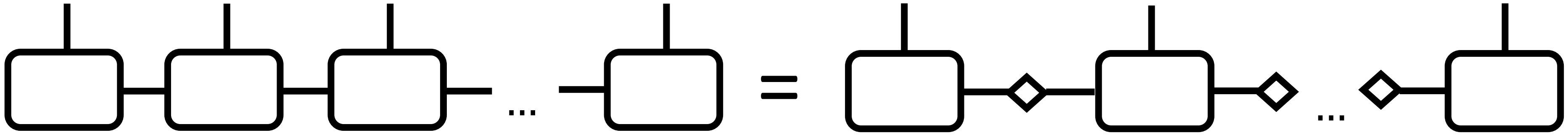


Bond dimension grows exponentially in the number of MPO applications. However, the rank of the matrices in the MPS might be much lower than the bond dimension.

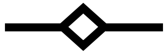
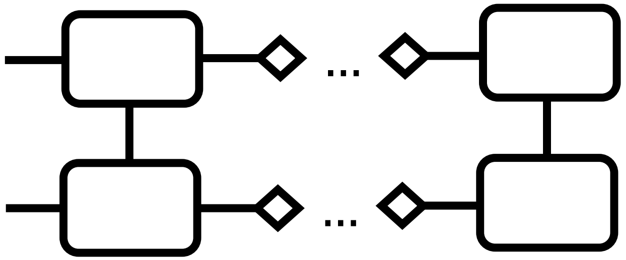
There are multiple algorithms to deal with this solution. A simple one uses the canonical form.

Controlling bond dimension

We can write:



Such that

 are diagonal  is the identity for all bonds, and similar for all the mirrored networks.

This is called the canonical form (A Practical Introduction to Tensor Networks: Matrix Product States and Projected Entangled Pair States, R. Orus, AOP 349: 117-158)

PASQAL

Controlling bond dimension

If the bond dimension of an MPS is larger than it needs to be, the ◇ will contain zeroes.
The bond dimension can be reduced by truncating those bonds.

This also forms the basis for an approximation scheme. In the canonical form, one can truncate a bond to drop all entries of ◇ smaller than some cutoff. In practice, this works really well.

PASQAL


PASQAL
FRESNEL
PASQAL