Technical University Munich
Department of Physics

Master's Thesis in Quantum Science & Technology

Using the Time Dependent Variational Principle for Tree Tensor Networks to Simulate Single-Qubit Quantum Error Correction

Anwenden des Time Dependent Variation Principle für Tree Tensor Networks zum Simulieren von Single-Qubit Quantum Error Correction

Julien Kollmann

November 15th, 2023

Supervised by Prof. Dr. Christian Mendl
Advised by Richard Milbradt

Preamble

Acknowledgements
I would like to thank my supervisor Prof. Dr. Christian Mendl and my advisor Richard Milbradt for their time, help and guidance. Richard's help during the work on this thesis was vital to its successful completion

Code
All code was written independently by me in Python using only standard modules as well as the module PyTreeNet. PyTreeNet is an unreleased in-development Python module for working with Tree Tensor Networks. Up to now, contributions to this module have been made by Richard Milbradt and me. In particular, the Time Dependent Variational Principle was implemented by me.

Figures
All figures were created independently by me using Python and the Python module Matplotlib.

Mathematical notation
An overview over the mathematical notation that is not specific for Tree Tensor Networks is provided in section A.

Abstract

Utilizing the recent generalization of the Time Dependent Variational Principle (TDVP) for Tree Tensor Networks (TTN) by [1], this thesis provides a TDVP implementation that is discussed along with TTN fundamentals extensively and then used to simulate quantum circuit operations for circuits that interact with an environment (or itself).
TTNs allow for well-performing computation of operations on the Hilbert space of a multi-qubit system. The TDVP algorithm for TTNs can be used to time-evolve a quantum state under the action of a system Hamiltonian, which is derived for a model of five qubits, where each qubit is connected to a single boson. The system dynamics are constituted of three parts: Nearest-neighbor qubit-qubit interaction, nearest-neighbor qubit-boson interaction and application of quantum circuit gates.
The goal is to apply this methodology to the 5-qubit quantum error correction (QEC) code by [2] and observe the impact a bosonic bath and qubit-qubit interaction have on the fidelity of the error correction. The model and error correction setup is similar to [3], where for the same QEC code a similar reduction of circuit fidelity is observed as in this thesis.
The presented algorithms and model are versatile and reliable. However, challenges arise when applying the algorithm to the presented model as the performance demand for accurate simulation is the main limitation to the number of presented configurations. A discussion of possible improvements to the implementations of the algorithms as well as to the model covers several aspects that could be included in future work.

Contents

1 Introduction

Despite experimental advances in quantum computation (QC) such as the demonstration of *quantum supremacy* by [4] or the advent of commercially available quantum compute time, e.g. IBM (see section 5.3), the error rates of modern quantum computers are still orders of magnitude too high for many practical applications of quantum algorithms ([5]). To mitigate errors such as bit or phase flips of the qubits, quantum error correction (QEC) codes are a useful tool. [3] employ the 5-qubit QEC code by [2] in their very recent publication[1] and observe increased resilience against noise introduced by a bosonic environment. To gain a better understanding of QEC and the challenges quantum computers face it is expedient to further analyze the 5-qubit code and the effects of noise models on the qubits.
Based on the unreleased, in-development Python module PyTreeNet to which Richard Milbradt and I have contributed, the relatively recent generalization of the Time Dependent Variational Principle (TDVP) for Tree Tensor Networks (TTN) by [6] enables a detailed exploration of the effects of n-boson bosonic baths on the qubits of the 5-qubit code. To that end, a suiting TTN has to be established along with specific operations that serve the representation and the time-evolution of quantum multi-particle states. The 5-qubit QEC code is discussed in-depth and the necessary operations are adapted to the model to allow simultaneous application of the 5-qubit en- and decoder as well as the dynamics induced by the n-boson bosonic baths.
TTNs allow performing meaningful operations on systems that could not be done in matrix-form as the matrix size scales exponentially. The capabilities of TTNs and manipulation of TTNs with the provided definitions and operations, such as the tensor-valued singular value decomposition (SVD), are relevant far beyond the scope of this specific problem.
QEC codes and in particular the 5-qubit code provide a pair of encoding and decoding operation that can protect a logical qubit from a single arbitrary error of one of its physical qubits – an error which could be catastrophic for the computation at hand – and is thus a powerful and important concept to QC entirely.
The presented model consists of 5 qubits which are each connected to their own bosonic bath that consists of one boson – a generalization of this model for multiple bosons per qubit is also provided. The Hamiltonian of the system is designed to allow for the qubits to interact with their respective boson as well as their nearest qubit neighbor(s). A modification of the Hamiltonian for the application of quantum gates is also presented. The methods used are therefore not unique to this problem and can also be transferred to other TTN problems.

[1] Only the preprint can be cited as the paper has been accepted only late October 2023.

2 Time-dependent Variational Principle for Tree Tensor Networks

Since the dimension D of the Hilbert space for quantum states grows exponentially with the dimensionality n of the quantum state (e.g. $D = 2^n$ for qubits), computing the time evolution of a quantum state represented by a vector becomes unfeasible for larger n. Physical interactions are local, so instead of representing the quantum state by a single vector, but a Matrix Product State (MPS) or – more generally – a Tree Tensor Network (TTN) with a structure well-suited for the interaction is used ([7]).
The underlying material for TTNs presented in this section is taken from [8], while the mathematical notation is adapted to the work done by [1] as the TDVP implementation in this thesis is based on it. Furthermore, to better align notation and Python code, tensor indices will be called "legs".
Some definitions that are commonly used in quantum physics and not specific to TTNs or TDVP are provided in section A.

2.1 TTN Basics

Tensors of rank N are N-dimensional arrays – just like matrices are 2-dimensional arrays and vectors are 1-dimensional arrays (or tensors of rank 2 or 1, respectively). A Tensor Network (TN) is composed of multiple tensors $T_i \coloneqq T_{Q_i}^{[s_i]}$, where s_i references the site of the tensor within the network and Q_i is the subset of legs q_k that are attached to the tensor. Every leg q_k is attached to exactly two tensors in such a way that the network remains loop-free, constituting a *Tree* Tensor Network (TTN) – those legs are also called bonds. There are exactly $N - 1$ unique q_k for every TTN. Additionally to the Q_i legs, each tensor T_i has attached either one or two so-called physical legs, as described in the upcoming section 2.1.2.

2.1.1 Contraction and decomposition

TNs are contracted along all legs q_k – which are those that connect exactly two tensors from the network. For two tensors $A_{a,b}$ and $B_{b,c}$ the contraction

$$C_{a,c} = \sum_b A_{a,b} B_{b,c} \tag{1}$$

is simply the matrix product of A and B. For tensors of higher rank, the legs can be grouped (either as legs of the resulting tensor or as contracted legs) and reinterpreted as single legs. Grouping legs and reinterpreting the shape of the tensor is the equivalent of reshaping an array in computational procedures.
Another important matrix function that can be generalized for tensors is the singular value decomposition which is used e.g. to perform low-rank approximations or in the TDVP algorithm for orthogonalizing the TTN. The matrix-SVD

$$C_{i,j} = \sum_q U_{i,q} S_{q,q} V^\dagger_{q,j} \tag{2}$$

can be rewritten for the tensor-SVD as follows:

$$C_{i_1,\dots,i_m;j_1,\dots,j_n} = \sum_q U_{i_1,\dots,i_m;q} S_{q,q} V^\dagger_{q;j_1,\dots,j_n} \,. \tag{3}$$

Oftentimes, e.g. when orthogonalizing a set of tensors, the singular values stored in S are not relevant and thus a QR-decomposition

$$C_{i_1,\dots,i_m;j_1,\dots,j_n} = \sum_q Q_{i_1,\dots,i_m;q} R_{q;j_1,\dots,j_n} \,, \tag{4}$$

which is easier to compute, suffices. Orthogonal tensors can simplify the computation of the contraction of the TN with their complex conjugate since the orthogonal matrix property $QQ^T = I$ translates to contracting a tensor along its open legs with its complex conjugate, i.e. orthogonal leaves and branches simplify to the

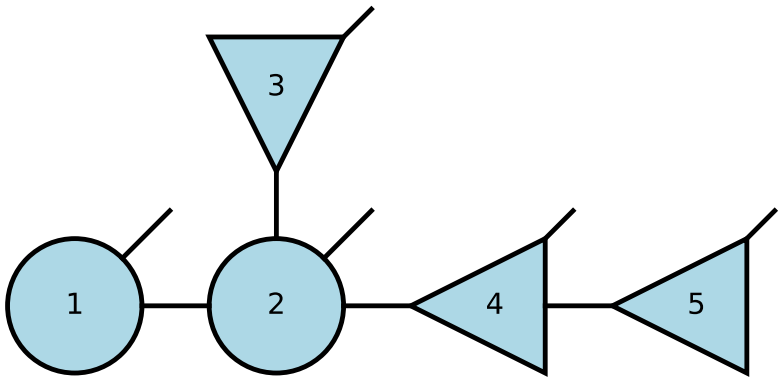

Figure 1: Tensors 1-5 form a loop-free TN, i.e. TTN. Each tensor has one physical leg, indicated by the upper-right diagonal lines. Tensors 3, 4 and 5 are orthogonal with respect to the tensor they point towards (2) as indicated by their triangular shape.

identity and can be ignored when contracting the TN with its adjoint. This is useful for calculating the norm of a TN or – in the context of quantum states – computing operator expectation values. For TNs, orthogonal parts are usually indicated by triangles pointing in the direction in which the contraction results in the identity. An example for a five-tensor TTN with partial orthogonalization is displayed in figure 1 (with one physical leg per tensor).
To touch on the Tree picture: The terminology is taken from the well-known standard graph theory. In the example of figure 1, T_2 could be chosen as the root. This would result in three branches: firstly T_1 and T_2 with leaf T_1, then T_2 and T_3 with leaf T_3 and lastly T_2, T_4 and T_5 with leaf T_5.

2.1.2 Representing quantum states and operators

As mentioned at the beginning of this chapter, TTNs are used for expressing quantum states in a more compact way. The N-dimensional quantum state

$$|\psi\rangle = \sum_{s_1,\dots s_N} c_{s_1,\dots,s_n} |s_1 \cdots s_N\rangle \in \mathcal{H} \tag{5}$$

written in the product basis $|s_1 \cdots s_N\rangle$ has $s_1 \cdots s_N$ coefficients $c_{s_1,\dots,s_N}$. When working with a TTN, those coefficients are split into a product of lower-rank tensors $\sum_{q_1,\dots,q_{N-1}} T_{Q_1}^{[s_1]} \cdots T_{Q_N}^{[s_N]} = c_{s_1,\dots,s_N}$. Every one of those $T_{Q_i}^{[s_i]}$ has exactly one physical leg s_i(having zero physical legs is equivalent to having one physical leg with dimension 1). While $c_{s_1,\dots,s_N}$ has $\dim(s_1)\cdots\dim(s_N)$ parameters, each tensor T_i with non-physical legs Q_i has $\dim(s_i)\cdot\prod_{q_k\in Q_i}\dim(q_k)$ parameters. For large N, i.e. systems with many sites, the latter expression can be substantially smaller. Requirement for that is that the dimensionality of the bonds between the tensors (bond dimension) is relatively small. The tensor network T is the set of tensors T_i and represents the quantum state $|\psi\rangle$, which is written as $|\psi[T]\rangle$.
The Hamiltonian (or any other operator $\in L(\mathcal{H})$) of the system has the same TTN structure as the state $|\psi[T]\rangle$, but with two physical legs at every site s_i (corresponding to matrix-valued operators for states in vector representation). An example is shown in figure 3, where for some quantum operator $\hat{O}$, the contraction corresponding to $\langle\psi|\hat{O}|\psi\rangle$ is shown for a 5-tensor TTN.
This network serves as a useful example representing a chain of 5 qubits, each connected to its neighbor(s). The tensors of the network can be chosen as

$$T \coloneqq \{T_{q_1}^{[s_1]}, T_{q_1,q_2}^{[s_2]}, T_{q_2,q_3}^{[s_3]}, T_{q_3,q_4}^{[s_4]}, T_{q_4}^{[s_5]}\}\,, \tag{6}$$

as shown in figure 2. The network T has physical legs $s_1, ..., s_5$, which can be grouped (vectorization) to match the single physical leg of $|s_1 \cdots s_5\rangle$. The tensors T_i also have legs $q_1, ..., q_4$. It is worth noting that

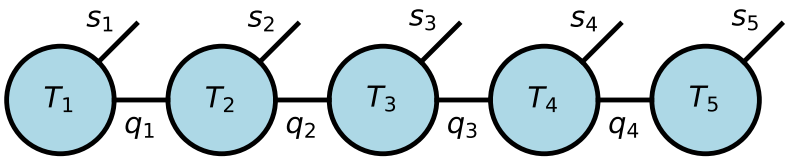

Figure 2: A 5-qubit chain is shown. Each qubit is represented by a tensor T_i with exactly one physical leg s_i of dimensionality 2. The tensors are connected to their respective neighbor(s) via bonds q_k. The bond dimensions between the tensors determine if the network is an exact representation of the quantum state (high enough bond dimension) or merely an approximation (truncated bond dimension).

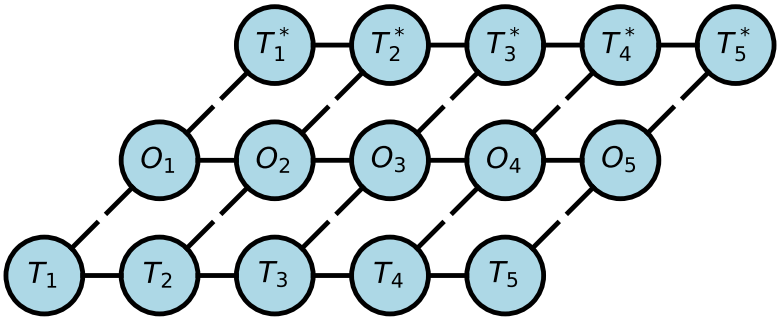

Figure 3: For the network from figure 2 the contraction $\langle\psi|\hat{O}|\psi\rangle$ for a TTN representation of an arbitrary quantum operator $\hat{O}$ acting on $|\psi\rangle$ is shown. The legs are not labeled to make the network structure easier to grasp. The physical legs are not connected in order to show that this is a contraction of three separate TTNs – interpreting the entire contraction as one single TTN is not possible (as there would be loops, e.g. T_1, T_2, O_2 and O_1).

while this representation classifies as a Matrix product state (MPS) and thus is only a special case of TTNs, it offers a great starting point for constructing TTNs with the intention of representing quantum circuits. The dimensionality of the physical legs is given by the nature of the system (i.e. 2 for qubits), but for the legs acting as the bond between the sites the dimensionality is not. If the bond dimension is chosen large enough, the quantum state is represented exactly. If on the other hand the bond dimension is truncated the TTN serves as a (good) approximation of the quantum state. The bond dimension necessary for an exact representation is lower for bonds closer to the leaves of the network and higher for bonds closer to the center of the network (i.e. far away from a leaf), see [7]. In the example T, q_2 and q_3 have higher dimensional demand than q_1 and q_4 when representing a highly entangled state. While the bond dimensions increase the amount of entanglement the network can represent exactly, they also increase the computational demand for contractions along the respective legs. It is thus useful to truncate the bond dimensions as far as possible without distorting the state too much by approximating too heavily. For operators corresponding to the network, e.g. the system Hamiltonian, the bond dimension is given by the mathematical formulation of the operator. There may be multiple exact representations with differing bond dimension though. In this case – since they perform equivalent operations – the representation with lower bond dimension is favorable due to lower computational cost.

Finding a suitable TTN structure is not always as trivial as in this example, but in many cases it is, e.g. Fork-Tree Product States (FTPS), as discussed in [6] and as used later in this thesis for the circuit model with one boson per qubit in section 4.

2.1.3 Orthogonality center

Two TTNs have the same value when their contractions result in the same tensor – even if the TTNs do not have the same parameters; There are degrees of freedom for TTNs. While [8] introduces also other degrees of freedom, the gauge degree of freedom is particularly relevant here. Constructing an invertible matrix $M \in \mathbb{C}^{\dim(q_k)}$ allows inserting $M^{-1}M = \mathbb{1}_{\dim(q_k)}$ along the leg q_k – obviously without changing the contraction result of the TTN or the quantum state the TTN represents. The physical state is invariant under the absorption of M (respectively M^{-1}) into the tensors connected by q_k. These so-called gauge transformations give rise to the question for canonical forms. It is quite useful (and for the derivation of the TDVP algorithm necessary) to orthogonalize a TTN. A TTN is orthogonalized if every tensor T_i of the network but one is orthogonalized to the remaining tensor T_c. A tensor A is orthogonalized towards a neighboring tensor B by first performing the QR-decomposition (eq. 4) and then replacing A with Q as well as absorbing R into B. A tensor A is orthogonalized towards another tensor B when all the tensors along the unique path that connects them are orthogonalized in that direction. To obtain an orthogonalized TTN, all tensors must be orthogonalized towards the desired orthogonality center, starting with the tensors that are the furthest away and then moving closer until the center is reached.
Looking only at orthogonalized parts of TTNs it is convenient to introduce the notation

$$|q_k^{[i]}\rangle\,, \tag{7}$$

where i is the site the orthogonalized sub-network is attached to via q_k. In this context, $q_k^{[i]}$ is a sub-network of the TTN and is named after the connecting leg. This notation allows to express a quantum state with orthogonality center $C_{Q_i}^{[s_i]}$ as

$$|\psi\rangle = \sum_{s_i,Q_i} C_{Q_i}^{[s_i]} |s_i\rangle \otimes_{q_k \in Q_i} |q_k^{[i]}\rangle \tag{8}$$

since connected at every leg in Q_i is an orthogonalized sub-network. Here, the sum is over all basis states of s_i as well as the $q_i \in Q_i$.
Orthogonalizing a TTN is important for the following section on the TDVP but is also useful for computing single-site expectation value of an operator $\hat{O}$. Complete contraction of the TTN can be avoided by orthogonalizing the TTN towards the site of interest, yielding the site-tensor T. After contracting T along its physical leg with $\hat{O}$ and its conjugate T^* (also along the physical leg), the expectation value can be obtained by tracing out all other legs, i.e. connecting each remaining leg of T with its T^* counterpart.

2.1.4 Reduced density matrix

For mixed (e.g. entangled) quantum states $|s_1 \cdots s_N\rangle$ the state of a single site/qubit s_i cannot be expressed as a vector $|s_i\rangle$. Instead, only the reduced density matrix ρ_{s_1} provides information about its state. TTNs offer a simple way to compute the reduced density matrix of a state $|\psi\rangle$ for a subsystem R:

$$\rho_R^{|\psi\rangle} = |\psi\rangle_R\langle\psi| = \mathrm{tr}_{(\neg R)}|\psi\rangle\langle\psi|\,, \tag{9}$$

which can be read as the outer product of $|\psi\rangle$ with its adjoint in combination with contracting the sites with their adjoints along all open legs except those of R, as shown in [8]. For the previous example T of the 5-qubit MPS this operation is shown in figure 4. Here, sites s_2, s_3 and s_4 have been chosen as the subsystem R. The shown contraction is thus $\rho_{s_2,s_3,s_4}^{|\psi\rangle}$.

2.2 TDVP

Quantum states $|\psi\rangle$ evolve under the action of a Hamiltonian $\hat{H}$ according to the well-known time-dependent Schrödinger equation (TDSE)

$$i\frac{d}{dt}|\psi\rangle = \hat{H}(t)|\psi\rangle \tag{10}$$

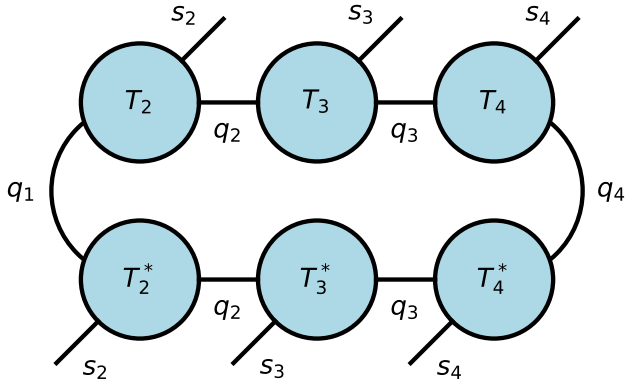

Figure 4: For the subsystem of sites 2, 3 and 4 of the network from figure 2 the partial trace is computed to obtain the reduced density matrix $\rho^{|\psi\rangle}_{s_2,s_3,s_4}$. The network graph shown here represents the outer product of the sites with their adjoints and the contraction along the remaining open legs with their adjoints.

with $\hbar = 1$. Utilizing the Time-Dependent Variational Principle (TDVP), this equation can be re-written as

$$i\frac{d}{dt}|\psi[T]\rangle = \hat{P}_{|\psi[T]\rangle}\hat{H}(t)|\psi[T]\rangle \tag{11}$$

for a TTN representation of the state $|\psi[T]\rangle$ and $\hat{P}$ the tangent space projector as derived in the following section.

2.2.1 Tangent space projector

The following derivation of the tangent space projector is based on work by [9][2] on TDVP for MPSs and extended by the generalization to TTNs by [1].
Let $\mathcal{M}_T \subset \mathcal{H}$ be the manifold (subspace) accessible by variation of the parameters of the TTN T that is used to represent a quantum state $|\psi\rangle \in \mathcal{H}$, i.e. $|\psi[T]\rangle \in \mathcal{M}_T \subset \mathcal{H}$. Considering the TDSE for $|\psi[T](t_0)\rangle$, the time-evolved state $|\psi(t_1)\rangle$ $(t_1 > t_0)$ will be in $\mathcal{H}$ but might be outside the manifold $\mathcal{M}_T$. To find a good approximation for $|\psi[T](t_1)\rangle$ (i.e. one that is $\in \mathcal{M}_T$) the TDVP utilizes a projection $\hat{P}_{|\psi[T]\rangle}$ onto the so-called tangent space – providing eq. 11 for solving the TDSE in the TTN subspace. The projection on the right hand side of the equation leads to vectors $i\frac{d}{dt}|\psi\rangle$, which are elements of the tangent space due to their nature as partial derivatives.
For an arbitrary vector $|\alpha\rangle \in \mathcal{H}$, the element $|\Theta_\alpha[B]\rangle = \hat{P}_{|\psi[T]\rangle}|\alpha\rangle$ of the tangent space of $|\psi[T]\rangle$ the optimization problem

$$\min_B \| \, |\Theta_\alpha[B]\rangle - |\alpha\rangle \, \|^2 \tag{12}$$

is equivalent to the projection problem. For TTNs specifically Bauernfeind et al. [1] derive the resulting projection operator

$$\begin{aligned}\hat{P}_{|\psi[T]\rangle} &= +\sum_{i=1}^{N} \mathbb{1}_{s_i} \otimes \sum_{Q_i} |q_1^{[i]} \cdots q_{r_i}^{[i]}\rangle\langle q_1^{[i]} \cdots q_{r_i}^{[i]}| - \sum_{q_k:(i,j)} |q_k^{[i]}\rangle\langle q_k^{[i]}| \otimes |q_k^{[j]}\rangle\langle q_k^{[j]}| \\ &=: +\sum_{i=1}^{N} \hat{P}_i^{\text{site}} - \sum_{q_k:(i,j)} \hat{P}_{i,j}^{\text{bond}}\end{aligned} \tag{13}$$

[2]Their supplementary material is extremely helpful. For an extensive derivation of the tangent space projector for MPSs, see the supplementary material of their later work [10].

where the sum over $q_k : (i, j)$ is a sum over all virtual legs q_k. The nearest-neighbor site-pair that is connected by q_k is site i and site j. Note the use of the orthogonalization notation from section 2.1.3. Applying the projector as demanded by the right hand side of eq. 11 results in the forward (+) time evolution of the N site tensors and the backward (−) time evolution of the $N-1$ bonds q_k. In this context the bonds q_k are also referred to as "links" – to emphasize that they are links between two tensor sites and not physical legs.

2.2.2 First-order integration

The integration scheme that solves this projected TDSE in the first-order solves the $2N-1$ terms individually (locally). Implementing a protocol to find a suitable integration sequence that completes one time step Δt can be done in various ways. The protocol described in [1] propagates from a start point to an end point through the network:

Algorithm 1: First-order TDVP integration sequence

```
start, end ⟵ Choose any two sites
i ⟵ start
while i ≠ end do
    q_k ⟵ Leg connecting i towards end
    if Any of Q_i \ q_k connects i towards pre-update site then
        i ⟵ Any leaf of the respective branch
    else
        SiteUpdate(i)
        LinkUpdate(q_k)
        i ⟵ Site connected to i via q_k
    end
end
SiteUpdate(end)
```

To update a site or a link, the time-discrete solution to the TDSE

$$A(t + \Delta t) = \mathrm{e}^{\pm i H_{eff} \Delta t} A(t) \tag{14}$$

for the vectorized site or link parameters A and the matricized effective site or link Hamiltonian H_{eff}. The sign in the exponent is − for the forward time evolution of the sites and + for the backward time evolution of the links[3]. The computation of A and H_{eff} will be discussed in the following subsection.
The sequence provided by algorithm 1 is not ideal and can require unnecessary many orthogonalizations between site updates. A good extension to the presented protocol is to choose start and end point of the sequence as two leaves, such that there is a maximum number of other sites along the path that connects these two. This way, the total distance being covered by jumping to the leaves of branches is reduced – which again reduces the number of orthogonalizations performed. When setting up a simulation for a specific tree structure it may be beneficial to design a specific sequence.

2.2.3 Site and link update

The site update as described in [1] uses

$$H_{(s_i Q_i),(s_i' Q_i')} = \langle s_i q_1^{[i]} \cdots q_{r_i}^{[i]} | H | s_i' q_1^{[i]\prime} \cdots q_{r_i}^{[i]\prime} \rangle \,, \tag{15}$$

where the grouped indices of H imply matricizing the tensor, as the effective Hamiltonian H_{eff}. Thus, the update of the site tensor T_i requires that the TTN is orthogonalized with T_i as its orthogonality center. Following the order of the previously introduced integration sequence yields many cases (all non-leaf sites) where the TTN is already in the correctly orthogonalized state – consequently this step can be often skipped

[3] Exactly contrary to the signs in the tangent space projector. This is because the TDSE needs to be multiplied with $-i$ before integrating.

during the site update:

Algorithm 2: Update site i

$|\psi[T]\rangle \longleftarrow T_i$-Orthogonalize T
$H_{eff} \longleftarrow H_{(s_iQ_i),(s_i'Q_i')}$
$T_i \longleftarrow timeEvolution(T_i, H_{eff}, forward)$

The link update includes a QR-decomposition orthogonalizing the TTN from site i towards the site j connected via q_k[4] as chosen in algorithm 1:

Algorithm 3: Update link q_k from site i towards site j

$C_{(s_iQ_i\setminus q_k),(q_k)} \longleftarrow$ Matricize T_i to isolate q_k
$Q, R \longleftarrow$ QR-Decompose $C_{(s_iQ_i\setminus q_k),(q_k)}$
$T_i \longleftarrow reshaped(Q)$
$H_{eff} \longleftarrow K_{(q_k^{[i]}q_k^{[j]}),(q_k^{[i]\prime}q_k^{[j]\prime})}$
$R \longleftarrow timeEvolution(R, H_{eff}, backward)$
$T_j \longleftarrow contract(T_j, R, q_k)$

with

$$K_{(q_k^{[i]}q_k^{[j]}),(q_k^{[i]\prime}q_k^{[j]\prime})} = \langle q_k^{[i]}q_k^{[j]}|H|q_k^{[i]\prime}q_k^{[j]\prime}\rangle \tag{16}$$

as the effective Hamiltonian H_{eff}.
To provide visualization of these updates, for both the site and the link update, figure 5 shows the effective Hamiltonian for a MPS with five sites. The effective Hamiltonian is shown in orange and is used to update the blue tensor respectively. For the site update that is the site tensor; For the link update the tensor to update is the R matrix from the QR decomposition of T_3.
A simple way to avoid unnecessary orthogonalizations is to perform the T_i-orthogonalization not as part of the site update, but rather immediately after jumping to a leaf in algorithm 1 as well as one time at the very start of the sequence.

2.2.4 Second-order integration

Instead of splitting the integration $e^{A+B+C} = e^Ae^Be^C$ (first-order approximation) it can also be split differently: $e^{A+B+C} = e^{A/2}e^{B/2}e^Ce^{B/2}e^{A/2}$ – providing a second-order approximation. Leveraging the algorithms developed for a first-order TDVP, the implementation of a second-order TDVP becomes extremely simple: Instead of updating the sites along the sequence with time step Δt, updates are performed along the sequence with time step $\frac{\Delta t}{2}$. After the last site has been updated, the sequence is applied again – in reverse order. When performing the updates in reverse order it needs to be considered that the link q_k previously updated from site i to site j now effectively switches direction and is updated after site j and before site i. Applying second-order integration reduces the numerical error more than it increases computation time as shown in section 2.3.
Another (small) benefit of second-order updating is that the orthogonalization for the starting point of the sequence only must be performed for the first time step. Since the starting point is also the last update in the second-order variant the network already has the starting point as its orthogonality center.

2.2.5 TDVP variants and dynamic bond dimension adjustments

The presented TDVP algorithm is known as *one-site* TDVP as another popular TDVP variant is *the two-site* TDVP (see [6]) where instead of the single-site and link updates a sequence of two site updates is performed. The two-site update contains the contraction of the two sites – which after the update is split up again using the tensor-valued SVD. In this step, the bond dimension q can be adjusted by decreasing the number of singular values from q to q' (truncation) or increased by filling the q singular values up with zeroes to q'

[4]Note that site j is not necessarily the site that gets updated next as one of $Q_j \setminus q_k$ might connect site j to sites that are yet to be updated.

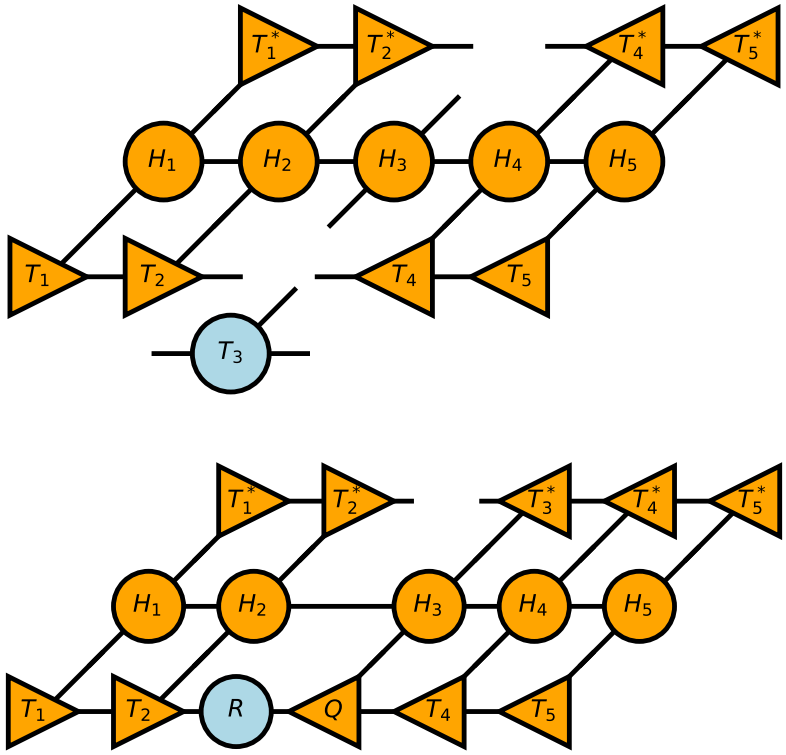

Figure 5: For the site (upper) and the link (lower) update the effective Hamiltonian is shown in orange for an MPS with five sites. The tensor that is subject to the update is shown in blue. For the link update, R originates in the QR decomposition of T_3 as part of updating the link from T_3 towards T_2. This figure is thus a visualization of the terms in equation 15 and 16 respectively.

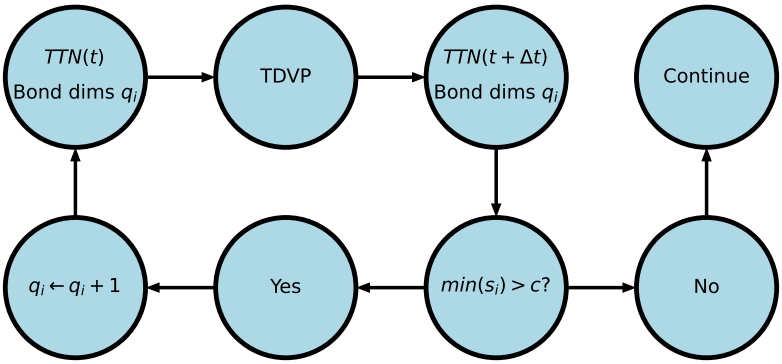

Figure 6: The procedure for increasing the TTN bond dimensions by demand. If the smallest singular value $min(s_i)$ obtained by an SVD on the two-site tensor of the sites connected by q_i is greater than a cutoff value c, the time-evolved state is computed again with slightly enlarged bond dimension(s). This increases the accuracy of the TTN representation of the state and is motivated by the two-site TDVP variant as well as the TDVP variant proposed by [11]. Note that here s_i refers to the singular values of the two-site tensor along q_i and not to a site of the TTN.

values. The advantage of this variant is that every time-step the bond dimensions can be increased to satisfy the demand created by entanglement. The drawback of this variant however is that there is a truncation error from the SVD decomposition which there is not in the one-site TDVP.
Increasing the bond dimension is beneficial for the accuracy of the TDVP algorithm as the state is in most cases only approximated by the TTN representation. This is also achievable without using the two-site TDVP, e.g. by using the algorithm proposed in [11]. Contracting two sites and splitting them up with an SVD can also be done independently of the time evolution. Consider a scheme where every n time-steps the singular values are calculated for every pair of nearest-neighbor sites. If the smallest singular value for the current bond dimension is larger than a cutoff value c, the time-evolved state is rejected and the last n time-steps are repeated with enlarged bond dimensions for the respective site-pairs. This procedure is motivated by arguments made about the distribution of singular values in [11] and will be used in this thesis (where indicated). A clear drawback is having to repeat certain parts of the time evolution but this method is beneficial to the accuracy of the simulation. In the context of simulating quantum circuits (see section 4.2) the singular values are computed after every gate application: The TDVP proposes a time-evolved state for which all bond dimensions are analyzed. The bond dimensions with singular values that violate the cutoff are then increased by 1 and the time-evolved state is rejected. The TDVP is then run again to propose a new time-evolved state with higher bond dimensions. Consequently, for some gates the TDVP will be run multiple times – which can partly be mitigated by setting the initial bond dimensions using an educated guess. A visualization of this procedure is shown in figure 6.
A reduction of bond dimensions (e.g. if there are many very small singular values) is not considered.

2.2.6 Implementation

The implementation of TDVP for this thesis is done in Python 3.10.
The time evolution in equation 14 is a matrix exponential. The computation of matrix exponentials can be costly (see [12]) and can become a bottleneck for performance and should thus be discussed. The implementation of the matrix exponential on a vector provided by [13] based on an algorithm by [14] turns out to be sufficient.
As the TTN is contracted with its conjugate and the Hamiltonian repeatedly for calculating the effective Hamiltonian, it is advisable to store those contractions in a cache, most of those contractions would otherwise be repeated without the parameters even changing.
To adequately judge the validity of the simulation and retrieve insights beyond the evolved network state, an expectation value measurement is performed at every (n-th) time step. One good choice is to define one

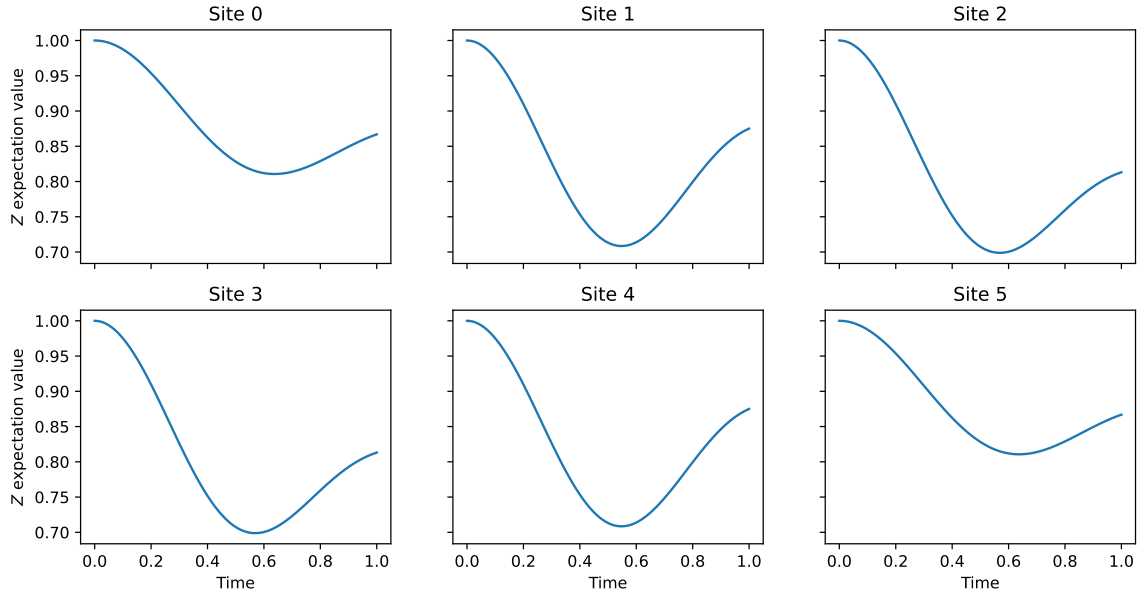

Figure 7: For the six-site MPS evolution under the Hamiltonian in eq. 17 the $\hat{Z}$-expectation value is computed every $\Delta t = 0.01$ and shown for every site individually.

operator for every site (e.g. $\hat{Z}$) and measure its expectation value by inserting it into the scalar product of the TTN with itself at the respective site as explained in section 2.1.3.

2.3 Benchmarking and accuracy

To validate and evaluate the TDVP implementation it is necessary to compare results and performance to a known example. To do se, the performances of first and second-order TDVP are compared to the exact result for a six-site MPS. The evolution of a six-site qubit chain, i.e. MPS, under the transverse-field Ising model (see e.g. [15]) Hamiltonian

$$H = \sum_{i=0}^{4} J\hat{X}_i\hat{X}_{i+1} + \sum_{i=0}^{5} g\hat{Z}_i \tag{17}$$

with $g = -1.3$ and $J = 0.8$ can be calculated exactly using matrix exponentiation since the Hilbert space is low-dimensional ($2^6 = 64$). This is done for $t \in [0, 1]$ with a time step size of $\Delta t = 0.01$ and every time-step the $\hat{Z}$-expectation value is saved for every qubit. The resulting dynamics are shown in figure 7. As expected the results show symmetry with respect to the site number and have a reasonable time-resolution.

The same evolution is performed using first and second-order TDVP – with varying virtual bond dimensions $1, 3, 5, 7, 9$. To show the advantage of using second-order TDVP over first-order TDVP, the second-order variant uses twice the time-step size, $\Delta t = 0.02$. Figure 8 shows that gain in accuracy from using second-order integration despite doubling the time-step size. For the virtual bond dimensions $3, 5, 7, 9$ the absolute error for the $\hat{Z}$-expectation values (averaged across the six sites) is shown for both TDVP modes. With increasing virtual bond dimension, the error decreases drastically for both variants, but more so for the second-order integration.

With decreasing the number of time-steps for the second-order TDVP, the amount of computational overhead is also decreased. Thus, for a low virtual bond dimension-regime, the second-order TDVP simulation requires less compute time than the first-order variant. For higher virtual bond dimensions the compute time is almost only network contraction and the time-savings are much smaller. Figure 9 shows the computation time needed for the TDVP simulations of the above example. The values are normalized to t_{base} –

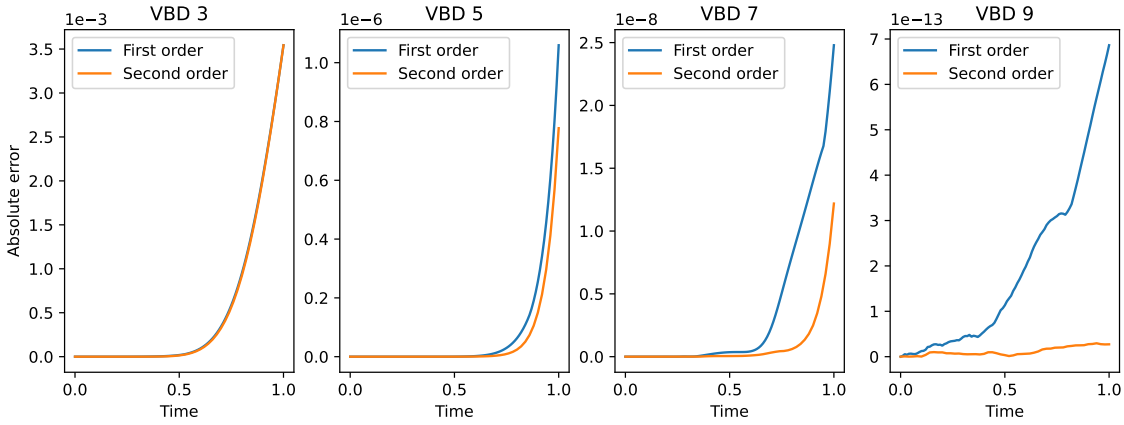

Figure 8: For the six-site MPS evolution under the Hamiltonian in eq. 17 the absolute error in $\hat{Z}$-expectation values (averaged across all six sites) compared to the exact solution is shown for first and second-order TDVP. The error dynamics are shown for virtual bond dimensions (VBD) $3, 5, 7, 9$ – demonstrating that with increasing bond dimension the MPS representation of the state becomes more precise. Also, despite having twice as large time-steps, the second-order TDVP simulation has smaller error than the first-order variant. The drastic increase in error over time could be mitigated by choosing smaller time-step sizes for both variants.

the time of the second-order TDVP simulation for virtual bond dimension 1[5]. The data show exponential increase of computation time with increasing bond dimension as well as clear performance advantage for the second-order TDVP.

In conclusion, the second-order TDVP is more accurate and faster to compute than the first-order TDVP. Also, the dilemma of gain in accuracy with loss in performance for increasing virtual bond dimensions persists – for both methods alike.

3 Quantum Error Correction

One way of working around high qubit error rates are Quantum Error Correction (QEC) codes, where multiple physical qubits are used to encode a state of interest – called logical qubit. Generally speaking, the encoded, multi-qubit state can be restored after a small portion of its physical qubits have been affected by an error and is thus more robust against decoherence than a single physical qubit.

QEC has become a fundamental concept of QC and is featured extensively in [16], which the following section will use as its base. Further helpful references are [17] and [18].

3.1 QEC Codes

The standard textbook example for QEC is the Shor code ([19]). One qubit of information $|\psi\rangle$ is encoded in a superposition of codewords

$$\begin{aligned} |\psi\rangle = \alpha|0\rangle + \beta|1\rangle \longrightarrow \quad & \alpha|0_L\rangle + \beta|1_L\rangle \coloneqq \\ = \quad & \alpha\frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{\sqrt{2^3}} \\ & +\beta\frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{\sqrt{2^3}}\,. \end{aligned} \tag{18}$$

[5] $t_{base} \approx 889$ms on an Intel Core i7-8550U.

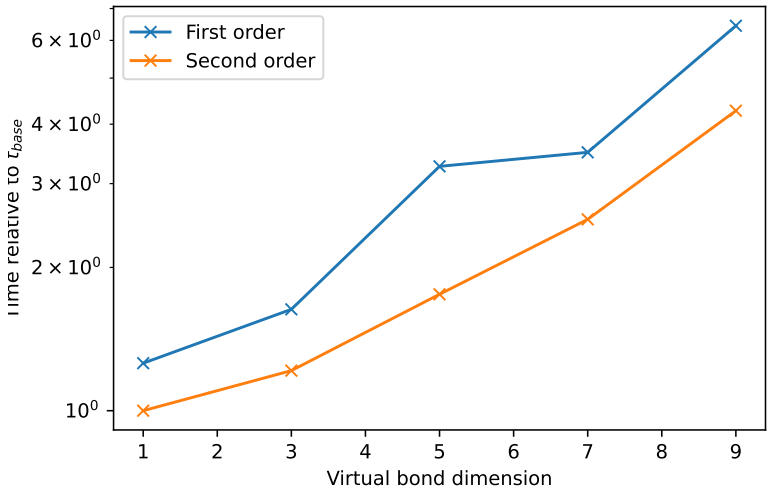

Figure 9: TDVP computation time for six-site MPS. For the one-site TDVP ($\Delta t = 0.01$) and the two-site TDVP ($\Delta t = 0.02$) the computation time is shown relative to t_base ($\approx 889ms$ on an Intel Core i7-8550U). The computational demand increases exponentially with the bond dimension of the TTN for both algorithm variants. While for lower bond dimensions the overhead is proportionally larger, even for greater bond dimensions the first-order TDVP always requires more time to compute than the second-order variant.

This encoding consists of two smaller encodings. The encoding

$$\begin{aligned} |0\rangle &\longrightarrow \quad |+++\rangle \\ |1\rangle &\longrightarrow \quad |---\rangle\,, \end{aligned} \tag{19}$$

followed by

$$\begin{aligned} |+\rangle &\longrightarrow \quad \frac{|000\rangle + |111\rangle}{\sqrt{2}} \\ |-\rangle &\longrightarrow \quad \frac{|000\rangle + |111\rangle}{\sqrt{2}} \end{aligned} \tag{20}$$

produces the 9-qubit codewords $|0_L\rangle$ and $|1_L\rangle$. The two-part concatenation of the encoding for the Shor code is a hint towards what errors can be corrected with this code: Equation 19 is the three-qubit phase flip code and equation 20 corresponds to the the three-qubit bit flip code. Together, any arbitrary single-qubit error can be corrected as any single-qubit error is a (linear) combination of bit- and phase flipping.
The Shor code requires 9 physical qubits to encode one qubit of information. Other codes have been developed that require less physical bits, while still meeting the requirements for correction of a single arbitrary error. There is a 7-qubit error by Steane ([20]) and even a 5-qubit code by Laflamme et al. ([2]. For the 5-qubit code it has been shown that this is the minimal number of physical qubits required to perform error correction of one arbitrary single-qubit error.

3.2 5-qubit QEC Code

For the 5-qubit error code, a qubit of information $|\psi\rangle$ stored in a single physical qubit $|q_3\rangle$ is encoded with the help of 4 other qubits into a superposition of $|0_L\rangle$ and $|1_L\rangle$. The quantum circuit for the encoding is provided by [2][6], but since section 5 compares results to those of [3], the notation will be adapted to the

[6]The notation used by [2] is partially outdated. What seems to be a CSWAP gate in the circuit notations is actually a sequence of two CNOT gates with the same control qubit. See section B.1 and/or figure 10.

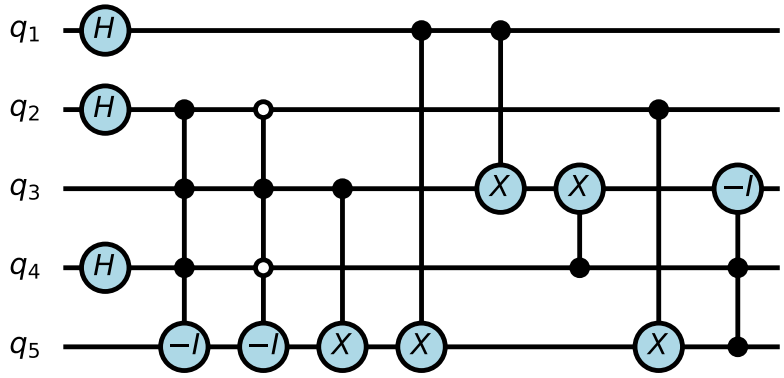

Figure 10: The 5-qubit encoding circuit used for the QEC code: The qubit of information in q_3 is encoded into a 5-qubit state via the shown circuit. The circuit consists of Hadamard gates, $CNOT$-gates as well as multi-control phase flip gates $-I$. The controlled gates act if the control qubits are in state $|1\rangle$ (for the empty circle in $|0\rangle$). The decoding circuit is U^{-1} and is the same circuit but has the gates in reverse order (right to left).

latter.
The encoding circuit U as shown in figure 10 and as further discussed in section B.1 uses multi-controlled $-I$-gates as well as $CNOT$-gates and Hadamard gates to encode

$$\begin{aligned}|q_1q_2q_3q_4q_5\rangle = |00\rangle_{12} \otimes (\alpha|0\rangle+\beta|1\rangle)_3 \otimes |00\rangle_{45} \longrightarrow \\ \alpha(|0\rangle + |6\rangle + |9\rangle - |15\rangle - |19\rangle + |21\rangle + |26\rangle + |28\rangle)/\sqrt{2^3} \\ +\beta(|3\rangle + |5\rangle + |10\rangle - |12\rangle - |16\rangle + |22\rangle + |25\rangle + |31\rangle)/\sqrt{2^3}\,, \end{aligned} \tag{21}$$

where the possible $q_1q_2q_3q_4q_5$ (with $q_j \in \{0,1\}$) have been interpreted as binary numbers and written out as decimal numbers for a briefer notation.

To recover from possible single-qubit errors induced by some erroneous quantum channel $\mathcal{E}$, the 5-qubit code uses U^{-1} to decode the 5-qubit state

$$|\psi\rangle \longrightarrow |q_1'q_2'\rangle \otimes (\alpha'|0\rangle + \beta'|1\rangle) \otimes |q_4'q_5'\rangle\,. \tag{22}$$

The original qubit of information $|\psi\rangle$ (i.e. α and β) can be restored from α' and β' using $|q_1'q_2'q_4'q_5'\rangle$ and without using any ancillary qubits: For each unique measurement result of $|q_1'q_2'q_4'q_5'\rangle$, α' and β' can be determined exactly – and all possible outcomes are listed in table 1. Completeness is obvious, given the two rightmost columns. To obtain measurement result as well as $|q_3'\rangle$ for a given error consider e.g. a bit flip error on qubit q_4 (as shown in a detailed example calculation in section B.1).

After determining α' and β', the original qubit can be restored by either performing simple single-qubit rotations or by re-interpreting $|0\rangle$ and $|1\rangle$ (change of basis). The error syndrome measurement and the qubit-recovery can also be combined into a projective measurement

$$\mathcal{R}(\rho) = \sum_{i=0}^{15} \hat{R}_k \rho \hat{R}_k^\dagger\,, \tag{23}$$

where ρ is the density matrix of the 5-qubit state *after* decoding via U^{-1} and R_i are Kraus operators as given in table B.3. Note that while table 1 and table B.3 can be derived one from another, their focus differs: The table given above focuses on the error at hand and is sorted accordingly, while the Kraus operators listed in table B.3 are structured with a focus on the result of the error syndrome measurement.

$\|q_1'q_2'q_4'q_5'\rangle$ measurement	α'	β'	Error type	Affected qubit
0110	$-\beta$	$-\alpha$	Bit flip	q_1
1000	$-\alpha$	$-\beta$	Phase flip	q_1
1110	$-\beta$	$-\alpha$	Bit & phase flip	q_1
0001	α	$-\beta$	Bit flip	q_2
0100	$-\alpha$	$-\beta$	Phase flip	q_2
0101	α	$-\beta$	Bit & phase flip	q_2
0111	$-\beta$	$-\alpha$	Bit flip	q_3
1010	α	$-\beta$	Phase flip	q_3
1101	β	$-\alpha$	Bit & phase flip	q_3
1011	$-\beta$	$-\alpha$	Bit flip	q_4
0010	$-\alpha$	$-\beta$	Phase flip	q_4
1001	$-\beta$	$-\alpha$	Bit & phase flip	q_4
0011	$-\alpha$	$-\beta$	Bit flip	q_5
1100	α	$-\beta$	Phase flip	q_5
1111	$-\alpha$	β	Bit & phase flip	q_5
0000	α	β	None	-

Table 1: Lookup table for restoring $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ based on the measurement results from the other 4 qubits q_1, q_2, q_3 and q_4 after the decoding operation U^{-1} as provided by [2].

Note that $\mathcal{R}(\rho)$ is not unitary. The corrected state is thus for arbitrary single-qubit errors (i.e. for single-qubit errors that are not either pure X or pure Z errors) not normalized and off by a global phase. Thus, instead of $\alpha|0\rangle + \beta|1\rangle$ the corrected state is $ce^{i\varphi}(\alpha|0\rangle + \beta|1\rangle)$ with $c, \varphi \in \mathbb{R}$. Numerically this can easily be resolved by dividing the resulting state (vector or TTN) by its norm, i.e. renormalizing the state. The phase can be ignored for most purposes, especially all cases that observe only the qubits used in the QEC code.

4 QEC within bosonic environment

The fidelity of quantum computations of course suffers not only from the errors discussed in the previous section, i.e. single-qubit bit and sign flips. Systems of qubits also undergo errors in the form of unwanted interactions with the environment as well as between themselves. To study the effect of QEC on the fidelity of a quantum circuit, a five-qubit system is coupled to a bosonic bath – simulating the environment of a quantum device. The chosen system Hamiltonian has nearest-neighbor qubit-qubit interaction (cross-talk) as well as qubit-boson (environment) interaction and thus offers a good challenge for the 5-qubit system.
Single-qubit errors as well as unwanted interactions only affect a qubit's ability to maintain its state. Quantum computation is also prone to errors in regards to preparing, manipulating and measuring states; See chapter 7 of [16]. In the scope of this work, state preparation and measurement as well as (multi-qubit) gate application are assumed ideal.

4.1 5-qubit system with bosonic bath

To simulate the system dynamics using TDVP, the model needs to be presented in a TTN structure – along with the corresponding Hamiltonian in the same structure.

4.1.1 TTN representation

Each of the five qubits is connected to exactly n_b bosons, all of which are attached to no other qubit, i.e. each qubit has its own, local environment. The model thus consists of five qubit tensors $q_1 ... q_5$ - connected via the legs $l_1 ... l_4$ - and $5n_b$ boson tensors $b_{1,i} ... b_{5,i}$ ($i = 1 ... n_b$) with their first index matching their adjoint qubits index. The bosons are connected to their respective qubit via links $l_{1,i} ... l_{5,i}$ (link index matches boson

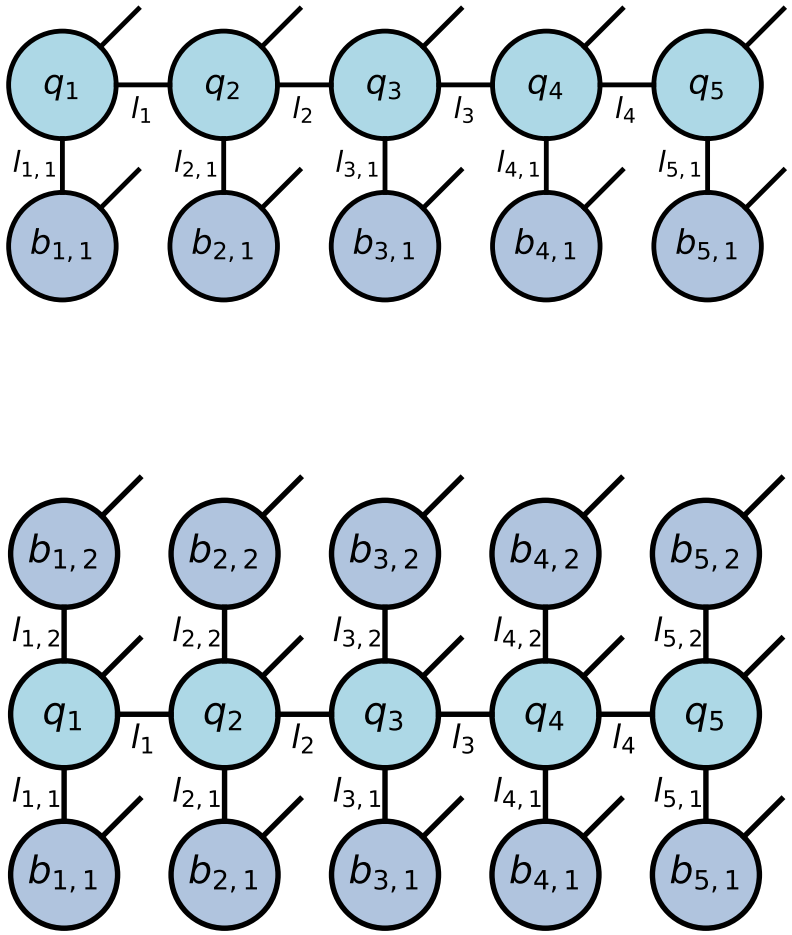

Figure 11: The 5-qubit system with its bosonic bath consists of five qubit tensors q_i with each qubit having one boson (upper) or two bosons (lower) $b_{i,j}$ attached via the link $l_{i,j}$. Each qubit q_i is connected to its neighbor q_{i+1} along the link l_i. The physical legs are indicated by the upper-diagonal strokes but not labeled in order to reduce clutter.

tensor index). The complete model is shown for $n_b = 1$ and $n_b = 2$ in figure 11 (with all ten /fifteen physical legs drawn, but for better readability not labeled).

4.1.2 System Hamiltonian

The system Hamiltonian is based on an XX-interaction Heisenberg model. While the XX-interaction between qubits constitutes the cross-talk between qubits, the model is expanded by qubit-boson as used in [21] (Chapter 10) and similar to [3]. The system Hamiltonian is thus given by

$$\hat{H} = \hat{H}_Q + \hat{H}_B + \hat{H}_I \,, \tag{24}$$

where

$$\hat{H}_Q = \sum_{q=1}^{4} J_q \cdot \hat{X}_q \hat{X}_{q+1} \tag{25}$$

is the XX nearest-neighbor Hamiltonian. The bosonic terms

$$\hat{H}_B = \sum_{q=1,b=1}^{5,n_b} \frac{m_{q,b}\omega_{q,b}}{2}\hat{\mathcal{N}}_{q,b} \tag{26}$$

and

$$\hat{H}_I = \sum_{q=1,b=1}^{5,n_b} g_{q,b}\hat{Z}_q(\hat{a}^\dagger_{q,b} + \hat{a}_{q,b}) \tag{27}$$

let the bosons interact with the qubits by coupling bosonic creation $\hat{a}^\dagger$ and annihilation $\hat{a}$ to phase flipping qubits via $\hat{Z}$. $\hat{\mathcal{N}} = \hat{a}^\dagger\hat{a}$ is the bosonic number operator. In general, operators with a single index q correspond to the qubit sites with the respective index, while operators with index (q, b) act on the bosonic sites with respective index.
To construct a TTN-valued operator for this Hamiltonian the structure is taken from the TTN representation of the state introduced above. The starting point is again the XX model (see [22]), with qubit-qubit bond dimension three:

$$\hat{H}_q^{TTO} = \begin{pmatrix} \hat{I} & 0 & 0 \\ \hat{X} & 0 & 0 \\ 0 & J_q \cdot \hat{X} & \hat{I} \end{pmatrix}, \tag{28}$$

where for the first site q_1 the Hamiltonian is given by only the third row and analogously for the last site q_5 by the first column. In this context 0 represents an all-zero 2×2 matrix. As this partial Hamiltonian leaves all bosons unchanged the corresponding bosonic operator is simply

$$\hat{H}_{q,b}^{TTO} = \hat{I}\,, \tag{29}$$

with a bond between the qubit sites and the boson sites of dimension one. To account for the bosonic energy term $\hat{H}_B$, the bond dimension is increased to two – and then to three to also account for the interaction term $\hat{H}_I$, yielding

$$\hat{H}_q^{TTO} = \begin{bmatrix} \begin{pmatrix} \hat{I} & 0 & 0 \\ \hat{X} & 0 & 0 \\ 0 & J_q \cdot \hat{X} & \hat{I} \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hat{I} & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hat{Z}_q & 0 & 0 \end{pmatrix} \end{bmatrix}, \tag{30}$$

$$\hat{H}_{q,b}^{TTO} = \begin{bmatrix} \hat{I} \\ \frac{1}{2} \cdot m_{q,b}\omega_{q,b} \cdot \hat{\mathcal{N}}_{q,b} \\ g_{q,b}(\hat{a}^\dagger_{q,b} + \hat{a}_{q,b}) \end{bmatrix} \tag{31}$$

for each boson. For n_b bosons attached to each qubit, this representation grows from a vector of length 3 to a $3^{\times n_b}$-tensor where the given vector is found along each main axis. Within the scope of this model all other terms, i.e. mixed terms (e.g. for two bosons the entry in the second row and second column), are zero-valued.

4.2 Quantum circuit simulation

To perform targeted manipulations on the 5-qubit system, circuit operations must be adapted to the TTN structure. Instead of writing multi-gates as operators on the $q_1 \otimes ... \otimes q_5$ space, a TTO for the TTN at hand has to be constructed. There are two different possibilities for doing so: Writing the gate action as a TTO – or constructing a Hamiltonian under which the system evolution for a specific time is equivalent to the gate action.

4.2.1 TTO representation

Choosing a TTO representation over a evolution under a Hamiltonian has one very clear advantage: The gate is applied by performing one simple contraction instead of performing a time evolution with many time steps. This leads to extremely shorter computational demand.
For single-qubit gates U_i it is sufficient to absorb the gate into qubit q_i, i.e. applying a TTO with bond dimension one that has the identity for every site except qubit q_i – where the tensor value is U.
For multi-qubit gates, i.e. controlled gates $C_{[j_1,...]}\overline{C}_{[k_1,...]}U_i$, that apply U onto qubit q_i if all the qubits $q_{j_1}, ...$ are 1 and all the qubits $q_{k_1}, ...$ are 0 can be obtained in a similar way: Again, the base of the TTO is the identity. The tensor-tuple $\begin{pmatrix}\hat{I} & |1\rangle\langle 1|\end{pmatrix}$ is then assigned to every control qubit $q_{j_1}, ...$, the tuple $\begin{pmatrix}\hat{I} & |0\rangle\langle 0|\end{pmatrix}$ is assigned to every inverse control qubit $q_{k_1}, ...$ and $\begin{pmatrix}\hat{I} & \hat{U} - \hat{I}\end{pmatrix}$ is assigned to the target qubit q_i. To obtain a viable TTO, every tensor that is between any of the gate-qubits is of shape $\begin{pmatrix}\hat{a} & 0\\ 0 & \hat{b}\end{pmatrix}$, where $\hat{a}$ and $\hat{b}$ are the two tensors from the tuple assigned to the site (twice the identity for uninvolved qubits). For the two out-most gate-qubits the shape is simply $\begin{pmatrix}\hat{a} & \hat{b}\end{pmatrix}$. As an example the $U = \overline{C}C\overline{C}\Pi_{2345}$ gate from the 5-qubit error code thus has the TTO representation

$$
\begin{aligned}
U_2 &= \begin{pmatrix}\hat{I} & |0\rangle\langle 0|\end{pmatrix}\\
U_3 &= \begin{pmatrix}\hat{I} & 0\\ 0 & |1\rangle\langle 1|\end{pmatrix}\\
U_4 &= \begin{pmatrix}\hat{I} & 0\\ 0 & |0\rangle\langle 0|\end{pmatrix}\\
U_5 &= \begin{pmatrix}\hat{I} & \hat{\Pi} - \hat{I}\end{pmatrix}\ .
\end{aligned}
$$

Helpful resources for constructing tensor representations of quantum gates are [22] and [23], however the construction can be understood fairly intuitively: The identities are applied in any case. If all control projection operators result in non-zero values, the target gate $\hat{U}$ is applied to the state. In this case the identity must be subtracted again as it has already been applied anyway.
Explicitly contracting the identity on all other sites has no benefit. In the context of 5-qubit QEC this explanation covers all necessary gates. Writing the whole 5-qubit QEC circuit as a single TTO (as opposed to using a TTO for every single gate) is not necessary as the required bond dimension is large enough to render the performance advantage over applying many TTOs meaningless.
For applying a TTO to a TTN there is an additional consideration to be made. After contracting all legs that connect the TTO with the TTN (i.e. the physical legs of the TTN) the network has the same structure as the TTN but all nodes that were involved in the contraction are now connected by two legs instead of one. Interpreting the two parallel legs as one single leg gives a new leg with dimension $\dim(l_T)\dim(l_O)$ where l_T is the leg that connects the respective nodes of the state and l_O the corresponding leg for the operator. Since $\dim(l_O)$ is not generally 1 (e.g. 2 for the operator in the above TTO representation) the dimension of the new leg is increased by this factor. There are two choices: Either the new leg can be accepted, which inflates the bond dimensions of the TTN especially for multiple TTO applications, or it can be truncated to the dimension it had before applying the TTO (via a tensor-valued truncated SVD). For feasibility and performance reason the second option seems to be the obvious choice – though this method introduces an unwanted truncation error to the network. This truncation error can in part be recovered from by re-normalizing the TTN after the truncation, but in general the error will be non-zero. Avoiding truncation errors is a further reason to apply gates via a circuit Hamiltonian.

4.2.2 Circuit Hamiltonian

Using a TTO representation of every gate used in the 5-qubit QEC is fast but has a major drawback: Gate application is instant, from a time evolution perspective. This means that the time evolution the system undergoes from the cross-talk in combination with interaction with the bosonic bath is completely seperate from the gate application. Since gate application is not instant, it is better to model the gate application by expanding the system Hamiltonian by a gate Hamiltonian for the duration of the gate. For extremely short gate times this solution approaches the instant TTO application.
Since a quantum system $|\psi\rangle$ evolves under a Hamiltonian H according to

$$|\psi(t)\rangle = \mathrm{e}^{-iHt}|\psi(0)\rangle\,, \tag{32}$$

the Hamiltonian H_U corresponding to a gate U according to

$$|\psi(t)\rangle = U|\psi(0)\rangle = \mathrm{e}^{-iH_U t}|\psi(0)\rangle \tag{33}$$

is given by

$$H_U t = iV \cdot \ln\Lambda \cdot V^{-1}\,, \tag{34}$$

where $U = V\Lambda V^{-1}$ is the eigendecomposition of U with the usual convention[7].
For single-qubit gates a gate Hamiltonian is thus given in the TTO format by choosing $H_U t$ at the respective site (and the identity everywhere else) and has bond dimension one.
For multi-qubit controlled gates, the Hamiltonian also has bond dimension one and is obtained (similar to the gate TTO) by placing $|1\rangle\langle 1|$ at every qubit site that is used for control (or $|0\rangle\langle 0|$ for inverted control) and $H_U t$ (the single-qubit Hamiltonian corresponding to U) at the target site.
To combine the gate Hamiltonian with the system Hamiltonian, the qubit-qubit bond dimension is increased by one and the third dimension of the Hamiltonian can be used for the gate matrices $\hat{U}_1, ..., \hat{U}_5$ as follows:

$$\hat{H}_1^{TTO} = \begin{bmatrix} \begin{pmatrix} 0 & J_1\cdot\hat{X} & \hat{U}_1 & \hat{I}\end{pmatrix} \\ \begin{pmatrix}\hat{I} & 0 & 0 & 0\end{pmatrix} \\ \begin{pmatrix}\hat{Z}_1 & 0 & 0 & 0\end{pmatrix} \end{bmatrix}, \hat{H}_{q\in\{2,3,4\}}^{TTO} = \begin{bmatrix} \begin{pmatrix} \hat{I} & 0 & 0 & 0 \\ \hat{X} & 0 & 0 & 0 \\ 0 & 0 & \hat{U}_q & 0 \\ 0 & J_q\cdot\hat{X} & 0 & \hat{I}\end{pmatrix} \\ \begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\\hat{I}&0&0&0\end{pmatrix} \\ \begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\\hat{Z}_q&0&0&0\end{pmatrix} \end{bmatrix}, \hat{H}_5^{TTO} = \begin{bmatrix} \begin{pmatrix}\hat{I}\\\hat{X}\\\hat{U}_5\\0\end{pmatrix} \\ \begin{pmatrix}0\\0\\0\\\hat{I}\end{pmatrix} \\ \begin{pmatrix}0\\0\\0\\\hat{Z}_5\end{pmatrix} \end{bmatrix}. \tag{35}$$

With this final version of the Hamiltonian, any single-qubit and any multi-controlled gate can be applied to the 5-qubit system while also evolving the system with its bosonic bath.
For cases where zero error is assumed (e.g. instantaneous gate application or reference computations), the Hamiltonian can be reduced to

$$\hat{H}_q^{TTO} = \hat{U}_q\,, \quad \hat{H}_{q,b}^{TTO} = \hat{I} \tag{36}$$

and always has virtual bond dimensions 1.

4.2.3 Quantum circuit fidelity

To meaningfully express how well a quantum circuit performs, a suiting decision for defining the fidelity in relation to the pre-encoding state $|\psi\rangle$ has to be made. Since the boson states (not relevant for quantum

[7] U is a gate and is thus unitary, i.e. the ln is well-defined for its diagonal matrix of eigenvalues.

computation) as well as the states of qubits q_1, q_2, q_4 and q_5 (set to $|0\rangle$ during error measurement) are not of interest for the correction process, the only significant information about the post-correction state $|\psi'\rangle$ is within the reduced density matrix $\rho_{q_3}^{|\psi'\rangle}$ of qubit q_3. Similar to [3], $F\left(|\psi\rangle, |\psi'\rangle\right) = \mathrm{tr}\left(\rho_{q_3}^{|\psi'\rangle} \cdot \rho_{q_3}^{|\psi\rangle}\right)$ can be considered. Since $|\psi\rangle$ is a multi-particle state, obtaining pure states for reduced single-site density matrices is unlikely – and it is thus useful to define the fidelity

$$F\left(|\psi\rangle, |\psi'\rangle\right) = \frac{\mathrm{tr}\left(\rho_{q_3}^{|\psi'\rangle} \cdot \rho_{q_3}^{|\psi\rangle}\right)}{\mathrm{tr}\left(\left(\rho_{q_3}^{|\psi\rangle}\right)^2\right)}, \tag{37}$$

normalized by the purity of the pre-encoding state. This definition of the fidelity yields $F = 1$ for perfect qubit recovery and can thus be understood as a direct measure of how well the 5-qubit QEC code performs.

4.3 Simulation setup

Before using the 5-qubit system with bosonic bath for simulating the 5-qubit QEC code, the network and system parameters need to be discussed. Apart from setting the values $J_q, g_{q,b}, ...$ for the system Hamiltonian, TDVP parameters like bond dimensions and time step size need to be considered. Most importantly, the number of bosons per qubit n_b is set to 1 (as visualized in figure 11, upper graph) in all following sections and experiments to reduce computation time.

4.3.1 System variables

For the qubit-qubit cross-talk strength J_q as well as for the bosonic properties $m_{q,b}, \omega_{q,b}$ and the interaction $g_{q,b}$ it is possible to set a different value for every qubit/boson. Since neither qubit-qubit cross-talk, nor the bosonic interaction are the main point of interest here – but rather the 5-qubit QEC code – all the values J, m, ω, g will be set to constant values, i.e. $J_q =: J$, $m_{q,b}\omega_{q,b} =: m$ and $g_{q,b} =: g$ for all q, b. The exact values used in each experiment are explicitly noted in the respective paragraphs.
The Hamiltonian for circuit simulation derived before defines the gates only up to a factor of $t =: t_{gate}$, i.e. the time interval within which the gate is applied. Since the circuit for the 5-qubit QEC encoder is composed of three single-qubit gates and eight multi-qubit gates, each multi-qubit gate time is set to $t_{gate} = \frac{1}{9}t_{enc.}$ (with $t_{enc.}$ the total encoding time), while the three single-qubit gates combined take up another time interval of $t = \frac{1}{9}t_{enc.}$ – split evenly across the three operations to give $t_{gate} = \frac{1}{27}t_{enc.}$. The 5-qubit encoder thus encodes the qubit within $\Delta t = t_{enc.}$ and single-qubit gates are three times as quick as multi-qubit gates.
To allow for short gate times (smaller t equals larger H_U) and thus a steeper gradient, each gate operation is performed with the same number of time steps, independent of t_{gate}. Since the action of the gates is much larger than the error action for relevant parameters (otherwise the circuit would simply be too noisy), the bottleneck are the gate applications and not the error dynamics.

4.3.2 TDVP parameters

The algorithm of choice is second-order TDVP since it can meet accuracy demands with higher time-step sizes than first-order TDVP as demonstrated in section 2.3.
For the TDVP simulation the most important parameters are the bond dimensions between the TTN sites as well as the time step size. With increasing bond dimension and decreasing time step size the simulation becomes more precise – but also much more time-consuming. In the case of applying the 5-qubit QEC code in particular the bond dimensions have to be large enough to allow accurate representation of the highly-entangled encoded state. Using the dynamic bond dimension increasing scheme proposed in section 2.2.5 the fidelity of the zero-error circuit (i.e. $J = m = g = 0$) is measured for different singular value cutoff values $c \in \{10^0, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$. The resulting maximum bond dimensions of the TTN are then $\{8, 10, 17, 22, 36\}$[8]. The fidelity-cutoff pairs are shown in figure 12 for time-steps (per circuit operation)

[8]The bond dimensions at the start of the simulation were chosen as 4 for qubit-boson bonds and as 8 for qubit-qubit bonds.

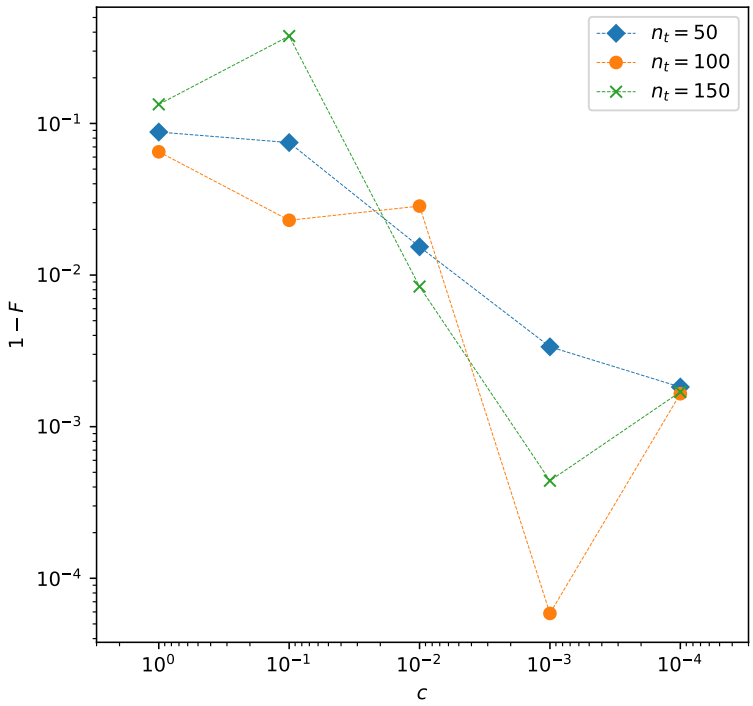

Figure 12: For the zero-error ($J = m = g = 0$) circuit that first encodes and then decodes the target state the relationship between the fidelity F and the singular value cutoff c is shown. For lower c the bond dimensions increase as more singular values are included – resulting in better accuracy of the circuit and approaching $F = 1$. The findings are shown for three numbers of time-steps n_t per circuit operation: $50, 100, 150$. There is no necessity for increasing n_t beyond 50 as the results vary very little for different n_t and the simulation inaccuracy is thus in a bond-dimension regime rather than a time-step regime.

$n_t \in \{50, 100, 150\}$. The ideal result for this simulation would of course be $1 - F = 0$ as in theory the applied operation is $U^{-1}U = I$. It is clearly visible that the accuracy of the simulation improves as c gets lower. Also, the time-step size is not the reason for $F < 1$ as for the three different n_t the results do not indicate any trend in that regard – which means that choosing $n_t = 50$ is sufficient for the values of c shown here. The necessary cutoff value for good accuracy seems to be even lower than 10^{-4}, however the resulting increase is linked to a computational expense that is too high producing meaningful, quantitative results within the scope of this thesis. To balance acceptable accuracy ($1 - F \approx 10^{-2}$ and acceptable computation time (up to $30s$ per time-step), $c = 10^{-2}$ is chosen. It is important to note that due to the complexity of the numerical methods, for certain parameters J, m, g the simulation may produce more accurate results than for others and $1 - F = 10^{-2}$ is not a limit on the accuracy for $c = 10^{-2}$.

For the observables, $\hat{Z}$ is chosen for the qubits and $\hat{\mathcal{N}}$ for the bosons. Using the bond dimension cutoff parameter $c = 10^{-2}$ ([9]) for parameters $J = 0.01$, $m = 0.1$ and $g = 0.01$ the recorded observables of a selection of sites are given in figures 15 (qubits) and 16 (boson) for an evolution of time $t = 1$ after error-free encoding of $|0\rangle$ within $t_{enc.} = 0.1$ and before error-free decoding with $t_{dec.} = t_{enc.} = 0.1$. The graphs demonstrate the effect of error activity for $t \in (0.1, 1.1)$ by showing rising occupation numbers for all bosonic sites as time progresses. For qubits q_3 and q_5 (who are not in superposition after the encoding) the expectation value

[9] In terms of the bond dimension increasing procedure the time evolution of the system without circuit operations is considered as a sequence of I-gates, i.e. the system can still seen as a quantum circuit, but idling.

changes during that interval can be spotted easily[10]
This concludes the setup of variables for the final simulations. To summarize, every circuit operation/gate is applied using $n_t = 50$ second-order TDVP time-steps. At every time-step there is a measurement for the single-site operator expectation values $\hat{Z}$ (qubits) and $\hat{\mathcal{N}}$ (bosons). To improve accuracy, whenever the smallest singular value obtained from an SVD on the two-site tensor along a bond l is greater than $c = 10^{-2}$, the dimension of link l is increased by one and the time-step computation is repeated.

5 Results

While the previous sections derive a model and methods that can be used to a more general extent, some model dynamics can be observed within the scope of this thesis. The following results are derived using the parameters introduced in the previous section, in particular $n_t = 50$ and $c = 10^{-2}$. Starting point is always the 5-qubit state $|00000\rangle$, i.e. the qubit to encode is $|0\rangle$.

5.1 State preservation in environment

For error-free encoding and decoding of the target state before and after an interaction time $t_{err.}$ with the environment, i.e. circuit idling, the circuit fidelity can be observed for varying J, m, g. Error-free de- and encoding can be interpreted as instantaneous gate application under the action of the environment Hamiltonian, i.e. the obtained fidelities are independent of the en-/decoding time. For $m = 0.1$ the parameters J and g are swept across a range of interest: $J \in (0.0001, 0.3)$ and $g \in (0.03, 0.3)$. The resulting fidelities are shown in figure 13. The data show improved circuit fidelity for lower values of J while the circuit fidelity does not exceed $\approx 1 - 10^{-3}$, which is likely due to the relatively high singular value cutoff value $c = 10^{-2}$. While the data does not fundamentally differ for changing values of g, the predominant effect in this graph is the qubit-qubit interaction governed by J – to accurately display the action on the circuit fidelity induced by m and g the range across which the parameters are swept needs to be increased (especially for m). However for the cross-talk effect the figure shows that for large enough J the error correction quality is so low it can be considered catastrophic in the context of error correction.

Similar work is done by [3]: Each qubit of the 5-qubit state is coupled with its own bosonic bath and the circuit fidelity is measured for different coupling strengths. Their numerical methods allow for more sophisticated bosonic baths and their model is thus an open quantum system, rather than a closed system of qubit-boson pairs. Their implementation however is not capable of applying the system dynamics to the state during the encoding and decoding as those are done by applying the respective operations directly to the state rather than incorporating them in the system Hamiltonian, i.e. the en-/decoding happens instantaneous and perfectly. While an exact comparison is not possible due to the model differences (and different research objectives), the results obtained by [3] show a similar trend to 13 – namely a decreased circuit fidelity F for stronger coupling J, m, g.

5.2 Non-zero encoding times

Considering the interaction between the bath and the qubits also during en- and decoding, i.e. not regarding the en- and decoding as instantaneous, the fidelity of the quantum circuit might change. For one set of parameters the simulation is successfully performed within a bit less than 10 hours. Due to the long compute times of the simulation, no further meaningful simulation results can be provided in this thesis. For $J = 0.1, m = 0.1, g = 0.01$ and a ratio of $t_{enc.}$ to t_{idle} of 0.1 the circuit fidelity is computed to be 0.927. For the same set of parameters, the fidelity is 0.940 for the case where the interaction is off during the encoding and decoding.
This single data point is not at all conclusive but highlights why running further simulations might yield

[10]The other qubits experience of course a similar effect – this is merely not as clearly visible in the Z-basis. Additionally measuring the $\hat{X}$-expectation value could show the dynamics clearer but since the exact dynamics are not as interesting as the circuit fidelity, the expectation value measurement is restricted to one operator for performance.

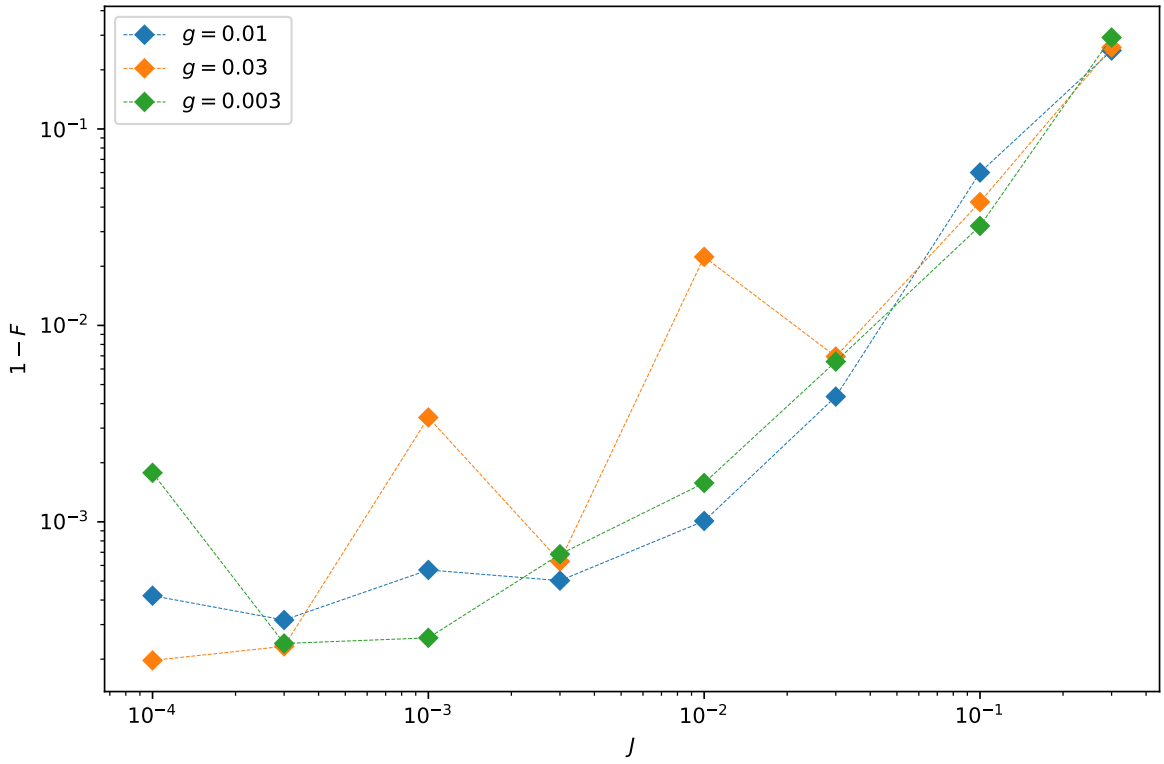

Figure 13: For varying parameters J, m the fidelity of the circuit is shown for $m = 0.1$. The encoding and decoding processes are considered instantaneous, i.e. error-free. The idle time between those two processes is $t = 1$. The results show that the fidelity of the circuit decreases as error parameters – namely J – increase. Fidelities above $\approx 1 - 10^{-3}$ cannot be reliably predicted as the accuracy of the simulation is limited by the singular value cutoff $c = 10^{-2}$.

results that improve the understanding of the impact the qubit-qubit interactions as well as the qubit-boson interactions have on the 5-qubit QEC code. This is further motivated by findings by [3]: When not considering the interactions during the en- and decoding it is found that the circuit fidelity can be increased by applying the QEC code repeatedly. The reduced fidelity of the simulation presented above gives rise to the question of whether or not (and if yes, for which parameter sub-spaces) this still holds true when *not* considering the encoding as well as the decoding as instantaneous processes. The results by [3] should be obtained (approximately) with the method presented in this thesis for $t_{idle} \gg 2Nt_{enc.}$ where N is the number of times the QEC protocol is applied – for other configurations and model parameters the fidelity might even decrease for repeated QEC application.

5.3 Implementation on commercially available quantum hardware

The importance of accounting for the environment interaction during the encoding and decoding processes is highlighted by looking at results from implementing the en- and decoder on commercially available hardware: IBM's quantum 27-qubit processor Falcon[11]. The encoder is implemented using Qiskit ([24]) by writing the controlled $-I$-gate as a *UGate* with parameters $2\pi, 0, 0$ (implementing the rest of the circuit is straightforward). The individual qubits are measured after the decoding for $n = 1024$ shots and the relative frequencies of the 5-qubit states are shown in figure 14. Since the decoded state was $|0\rangle$ and the state after decoding resembles the 5-qubit equal-superposition state, no information can be reliably recovered in this example, i.e. the fidelity is minimal even though the idle time of the circuit is zero.
Of course on real hardware many more errors can occur than those covered by the bosonic-bath model (as mentioned in section 4), but this result is still indicative of the current state of commercially available quantum computers.

6 Conclusion

This work successfully provides an introduction to the topics of TTNs, TDVP and QEC as well as quantum circuits and their integration in the system Hamiltonian – with the goal of simulating the interaction of a quantum system with its environment while performing circuit operations such as QEC codes. To discuss the limitations and opportunities of this work, aspects of implementation and modeling are discussed separately.

6.1 Sophisticated implementation

At the computational core of this thesis is the TDVP implementation that is successfully implemented for first and second order integration. A procedure to change the bond dimension as needed is proposed and applied to the simulations – yielding the desired changes to the TTN and the linked increase in accuracy. The bottleneck for the number of experiments for this thesis as well as their extent however is mainly the TDVP algorithm. The contraction of the effective Hamiltonians as well as the computation of matrix exponentials is very time intensive and limits the possibilities of applying the algorithm within the scope of this thesis - especially for those cases where the gate and the error dynamics are applied simultaneously due to the large Hamiltonian bond dimension.
To improve performance, major computational results are stored in a cache to minimize the number of computations necessary, but performance remains poor. For future simulations, these issues could be resolved e.g. by optimizing the underlying data structure for the TTNs or by executing time-sensitive parts of the code (e.g. the network contractions) in another programming language.
For large-scale experiments it is also possible to execute the relevant code on more sophisticated hardware, where less-optimized performance is not as restricting.

[11] Falcon r5.11, 27 Qubits, via the algiers backend. Run on October 21, 2023. IBM Cloud access required.
See https://quantum-computing.ibm.com/ and https://quantum-computing.ibm.com/services/resources?tab=systems

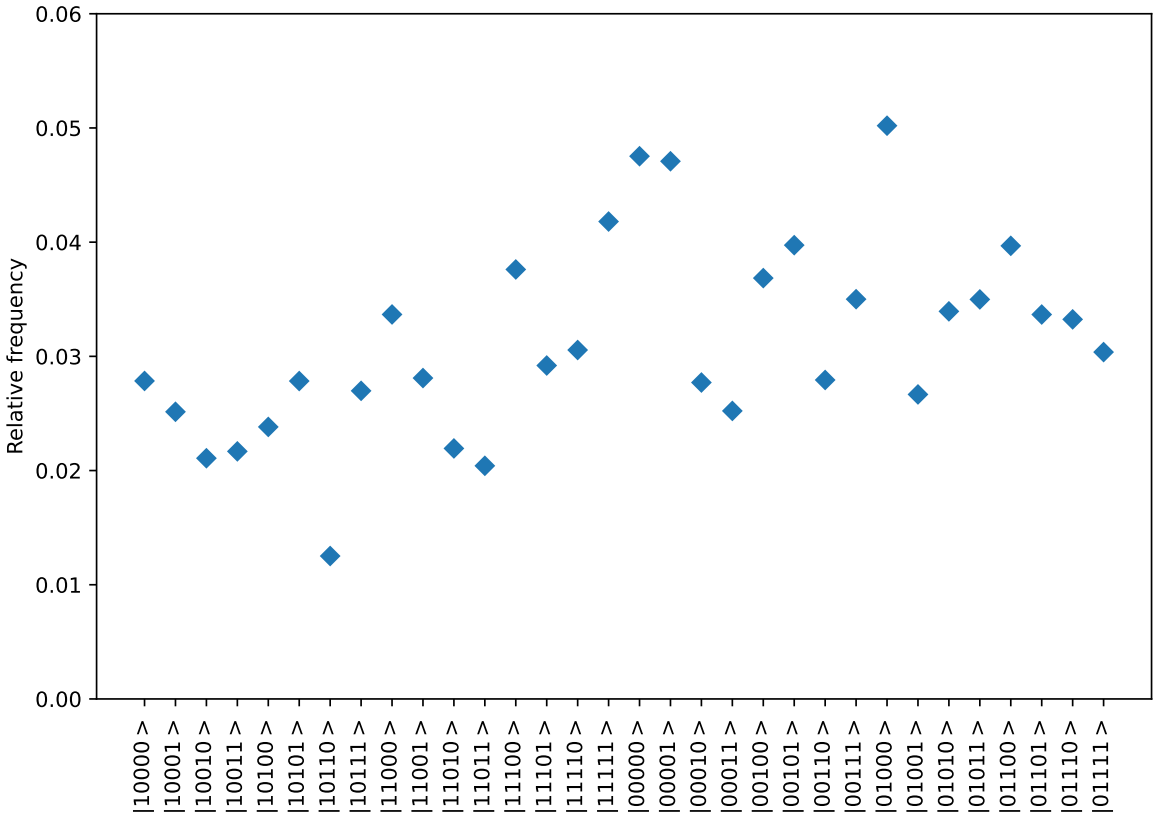

Figure 14: A circuit composed of the 5-qubit encode followed by the decoder with no idle time between, the relative frequency of the measured states is shown for $n = 1024$ shots. The result is obtained using a Falcon quantum processor by IBM. The resulting state has very low overlap with the computational result $|00000\rangle$.

6.2 Improvements to the model

The quality of the experiments is strongly correlated to the extent of the model, which is implemented to accommodate for a variable number of bosons per qubit and the option to have different parameters J, m, g for every site. However, several adaptations could be made to better align the system dynamics with reality:

- Gate errors: Gate imperfections occur for real quantum gates but are not considered in this thesis since this would introduce more model parameters and require more simulations, which is a problem due to the performance issues mentioned above. Implementing gate errors may yield more realistic results. To achieve this, no fundamental change to the implementation is necessary.
- Encoder: The presented QEC code is not the only QEC code. Considering alternative encoding and error recovery schemes may increase the diversity of the results. Additionally, in this work the error correction is only done for a single qubit. Quantum computations require multiple qubits for which other codes might be a better choice. The implementation of quantum circuits for this thesis allows specific gates and circuits to be constructed quickly.
- Hardware: The model presented here is independent of the intended quantum hardware. Depending on the architecture of the quantum computer, certain operations might be favorable while others might not possible to do at all. Matching the quantum operations of the model to the capabilities of the quantum hardware at hand might produce more accurate results. Furthermore, the qubits of the circuit are arranged in a straight line. For certain hardware this might not be the case and it may be better to find a TTN structure for the qubits that fits the hardware layout better. The implemented TTN and TDVP algorithms are viable for any tree structure and can thus still be used after potential adjustments to the system layout.
- Cross-talk range: To keep the bond dimension of the system Hamiltonian low, cross-talk is only considered for nearest neighbors. Adding longer-range cross-talk to the Hamiltonian might represent computational noise more realistically and can easily be done by manipulating the existing Hamiltonian by adding additional bond dimensions.

To summarize, the presented model is versatile and easy to adapt in order to expand functionality. The model can be used to simulate circuits, not only in the context of QEC, as well as their interaction with the environment reliably and easily.

6.3 Summary

The presented concepts for TTNs (namely the TDVP algorithm) and QEC are applied to the constructed model successfully. It is observed that for larger coupling between qubits and qubits as well as between qubits and bosons, the circuit fidelity of the QEC code decreases, which aligns with current literature.
Algorithms and methods such as the TDVP algorithm or representing quantum circuits as part of the system Hamiltonian can be applied beyond the scope of this thesis.
The findings of this study are limited by the high performance demand of the implemented TDVP algorithm. However, observing that the TDVP algorithm can be used on this model to simulate QEC on a quantum circuit that has interaction with the environment as well as within itself, is part of the result of this work; So is identifying the weak spots of the implementation as well. Improving the performance of the implementation and accessing high-performing hardware may allow to run the simulation for more configurations – e.g. in order to explore how to improve QEC codes.

References

1. Bauernfeind, D. & Aichhorn, M. Time dependent variational principle for tree Tensor Networks. *SciPost Physics* **8** (Feb. 2020).
2. Laflamme, R., Miquel, C., Paz, J. P. & Zurek, W. H. Perfect Quantum Error Correcting Code. *Phys. Rev. Lett.* **77,** 198–201. https://link.aps.org/doi/10.1103/PhysRevLett.77.198 (1 July 1996).
3. Babu, A. P. *et al.* Quantum error correction under numerically exact open-quantum-system dynamics. arXiv: 2212.07718 [quant-ph] (Dec. 2022).
4. Arute, F. *et al.* Quantum supremacy using a programmable superconducting processor. *Nature* **574,** 505–510 (Oct. 2019).
5. And Zijun Chen *et al.* Exponential suppression of bit or phase errors with cyclic error correction. *Nature* **595,** 383–387 (July 2021).
6. Bauernfeind, D., Zingl, M., Triebl, R., Aichhorn, M. & Evertz, H. G. Fork Tensor-Product States: Efficient Multiorbital Real-Time DMFT Solver. *Physical Review X* **7,** 031013 (July 2017).
7. Perez-Garcia, D., Verstraete, F., Wolf, M. M. & Cirac, J. I. Matrix Product State Representations. *Quantum Inf. Comput. 7, 401 (2007).* arXiv: quant-ph/0608197 [quant-ph] (Aug. 2006).
8. Bridgeman, J. C. & Chubb, C. T. Hand-waving and Interpretive Dance: An Introductory Course on Tensor Networks. *J. Phys. A: Math. Theor. 50 223001 (2017)* **50,** 223001. arXiv: 1603.03039 [quant-ph] (Mar. 9, 2016).
9. Haegeman, J. *et al.* Time-dependent variational principle for quantum lattices. *Phys. Rev. Lett. 107, 070601 (2011)* **107,** 070601. arXiv: 1103.0936 [cond-mat.str-el] (Mar. 4, 2011).
10. Haegeman, J., Lubich, C., Oseledets, I., Vandereycken, B. & Verstraete, F. Unifying time evolution and optimization with matrix product states. *Phys. Rev. B 94, 165116 (2016)* **94,** 165116. arXiv: 1408.5056 [quant-ph] (Aug. 21, 2014).
11. Xu, Y., Xie, Z., Xie, X., Schollwöck, U. & Ma, H. Stochastic Adaptive Single-Site Time-Dependent Variational Principle. *JACS Au 2, 335 (2022)* **2,** 335–340. arXiv: 2110.12703 [cond-mat.str-el] (Oct. 25, 2021).
12. Moler, C. & Loan, C. V. Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later. **45,** 3–49. ISSN: 0036-1445 (2003).
13. Virtanen, P. *et al.* SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* **17,** 261–272 (2020).
14. Al-Mohy, A. H. & Higham, N. J. Computing the Action of the Matrix Exponential, with an Application to Exponential Integrators. *SIAM Journal on Scientific Computing* **33,** 488–511. eprint: https://doi.org/10.1137/100788860. https://doi.org/10.1137/100788860 (2011).
15. Stinchcombe, R. B. Ising model in a transverse field. I. Basic theory. **6,** 2459–2483. ISSN: 0022-3719 (1973).
16. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information: 10th Anniversary Edition* 10th. ISBN: 1107002176 (Cambridge University Press, USA, 2011).
17. Devitt, S. J., Nemoto, K. & Munro, W. J. Quantum Error Correction for Beginners. *Rep. Prog. Phys. 76 (2013) 076001* **76,** 076001. arXiv: 0905.2794 [quant-ph] (May 18, 2009).

18. Gottesman, D. Stabilizer Codes and Quantum Error Correction. arXiv: quant-ph/9705052 [quant-ph] (May 1997).

19. Shor, P. W. Scheme for reducing decoherence in quantum computer memory. *Physical Review A* **52,** R2493–R2496 (Oct. 1995).

20. Steane, A. Multiple Particle Interference and Quantum Error Correction. *Proc.Roy.Soc.Lond. A452 (1996) 2551* **452,** 2551–2577. arXiv: quant-ph/9601029 [quant-ph] (Jan. 29, 1996).

21. Breuer, H.-P. & Petruccione, F. *The Theory of Open Quantum Systems* ISBN: 9780199213900. https://doi.org/10.1093/acprof:oso/9780199213900.001.0001 (Oxford University Press, Jan. 2007).

22. Fröwis, F., Nebendahl, V. & Dür, W. Tensor operators: Constructions and applications for long-range interaction systems. *Phys. Rev. A* **81,** 062337. https://link.aps.org/doi/10.1103/PhysRevA.81.062337 (6 June 2010).

23. Gelß, P., Klus, S., Knebel, S., Shakibaei, Z. & Pokutta, S. Low-Rank Tensor Decompositions of Quantum Circuits. *Journal of Computational Physics.* under review (2022).

24. Qiskit contributors. *Qiskit: An Open-source Framework for Quantum Computing* 2023.

A Definitions

- $\hat{X}$, $\hat{Y}$, and $\hat{Z}$ are the usual Pauli matrixes

$$\hat{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \hat{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \hat{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{38}$$

- $\hat{H}$ is the normalized two-dimensional Hadamard matrix $\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$.
- All matrices and tensors$O$ are written as $\hat{O}$ if they are used as operators in the given context.
- Tensor networks (or tensors or vectors) ψ that represent quantum states are written as $|\psi\rangle$ and their complex conjugate as $\langle\psi|$.
- $|\psi_1\psi_2\rangle$ is short for the Kronecker product $|\psi_1\rangle \otimes |\psi_2\rangle$.

B Calculations

B.1 En- and decoding scheme for 5-qubit QEC code

The unitary encoding operation U for the 5-qubit code by [2] is given by

$$U = CC\Pi_{453} \cdot CX_{25} \cdot CX_{43} \cdot CX_{13} \cdot CX_{15} \cdot CX_{35} \cdot \overline{C}C\overline{C}\Pi_{2345} \cdot CCC\Pi_{2345} \cdot H_1 \cdot H_2 \cdot H_4\,,$$

where the indices denote the qubit number with qubit 3 holding the initial state of interest. CX_{ab} is the standard controlled X gate with control qubit a and target qubit b. H_a is the Hadamard gate acting on qubit a. The other gates are controlled $\Pi \coloneqq -I$ operations with a variable number of control qubits, which is given by the amount of C's. $\overline{C}$ indicates inversion of the control (i.e. $\overline{C}\Pi_{ab} = X_a \cdot CZ\Pi_{ab} \cdot X_a$) and the last index is always the target qubit.
It is worth noting, that the circuit in [2] might be hard to understand. The gate that is by modern notation a $CSWAP$ gate is actually a composition of two consecutive $CNOT/CX$ gates that shre their control qubit. The R gates are Hadamard gates (H) and the Π operations are sign flips, i.e. $\Pi = -I$, as mentioned above. To show that this interpretation is correct a detailed example calculation is shown in section B.1.
Since every factor, i.e. gate, of U is unitary, the decoding operation is given by U^{-1} which can be obtained by applying the gates in reverse order.
To shorten notation for calculations, split $U =: U_1 \cdot U_2$ with

$$\begin{aligned} U_1 &= CC\Pi_{453} \cdot CX_{25} \cdot CX_{43} \cdot CX_{13} \cdot CX_{15} \cdot CX_{35} \cdot \overline{C}C\overline{C}\Pi_{2345} \cdot CCC\Pi_{2345} \\ U_2 &= H_1 \cdot H_2 \cdot H_4\,, \end{aligned}$$

and $U^{-1} = U_2^{-1} \cdot U_1^{-1}$. Note that this decomposes the decoding into a sequence of U_1^{-1} followed by U_2^{-1}.

B.2 Example calculation for 5-qubit error syndrome

$$
\begin{aligned}
|\psi\rangle_{inf.} &= |00\rangle_{12} \otimes (\alpha|0\rangle + \beta|1\rangle)_3 \otimes |00\rangle_{45} \\
\text{apply } U \longrightarrow \sqrt{8}|\psi\rangle_{enc.} &= \alpha|00000\rangle - \beta|00011\rangle - \beta|00101\rangle + \alpha|00110\rangle \\
&\quad + \alpha|01001\rangle - \beta|01010\rangle + \beta|01100\rangle + \alpha|01111\rangle \\
&\quad - \beta|10000\rangle - \alpha|10011\rangle + \alpha|10101\rangle + \beta|10110\rangle \\
&\quad + \beta|11001\rangle + \alpha|11010\rangle + \alpha|11100\rangle + \beta|11111\rangle \\
\text{Bit flip } q_4 \longrightarrow \sqrt{8}|\psi\rangle_{err.} &= \alpha|00010\rangle - \beta|00001\rangle - \beta|00111\rangle + \alpha|00100\rangle \\
&\quad + \alpha|01011\rangle - \beta|01000\rangle + \beta|01110\rangle + \alpha|01101\rangle \\
&\quad - \beta|10010\rangle - \alpha|10001\rangle + \alpha|10111\rangle + \beta|10100\rangle \\
&\quad + \beta|11011\rangle + \alpha|11000\rangle + \alpha|11110\rangle + \beta|11101\rangle \\
\text{apply } U_1^{-1} \longrightarrow \sqrt{8}|\psi\rangle_{int.} &= \alpha|00111\rangle - \beta|00001\rangle + \beta|00011\rangle - \alpha|00101\rangle \\
&\quad + \alpha|01111\rangle - \beta|01001\rangle + \beta|01011\rangle - \alpha|01101\rangle \\
&\quad - \beta|10011\rangle + \alpha|10101\rangle - \alpha|10111\rangle + \beta|10001\rangle \\
&\quad - \beta|11011\rangle + \alpha|11101\rangle - \alpha|11111\rangle + \beta|11001\rangle
\end{aligned}
$$

After the last step it is already obvious, that $|0\rangle_3$ is to be associated with β and $|1\rangle_3$ with α. Writing $|\psi\rangle_{int.} = |\psi\rangle_{1245} \otimes (\alpha|1\rangle + \beta|0\rangle)_3$ simplifies the expression significantly:

$$
\begin{aligned}
|\psi\rangle_{1245} &= \frac{1}{\sqrt{8}}(|0011\rangle - |0001\rangle + |0111\rangle - |0101\rangle + |1001\rangle - |1011\rangle + |1101\rangle - |1111\rangle) \\
&= -|-+-1\rangle
\end{aligned}
$$

After applying U_2^{-1} the decoded state

$$
|\psi\rangle_{dec.} = |10\rangle_{12} \otimes (-\beta|0\rangle - \alpha|1\rangle)_3 \otimes |11\rangle_{45} \tag{39}
$$

can be looked up in the error table 1: 1011 corresponds to a bit flip on the fourth qubit and the state $(-\beta|0\rangle - \alpha|1\rangle)_3$ is correctly predicted. The corresponding operator of the map in table B.3 is $\hat{R}_1 1$, applying $\hat{X}_3$ to the state. The resulting qubit then reads $-\alpha|0\rangle - \beta|1\rangle$ – which is equal to the initially encoded single-qubit state up to a global (i.e. irrelevant) phase -1.

B.3 Kraus operators for 5-qubit code recovery step

C Additional figures

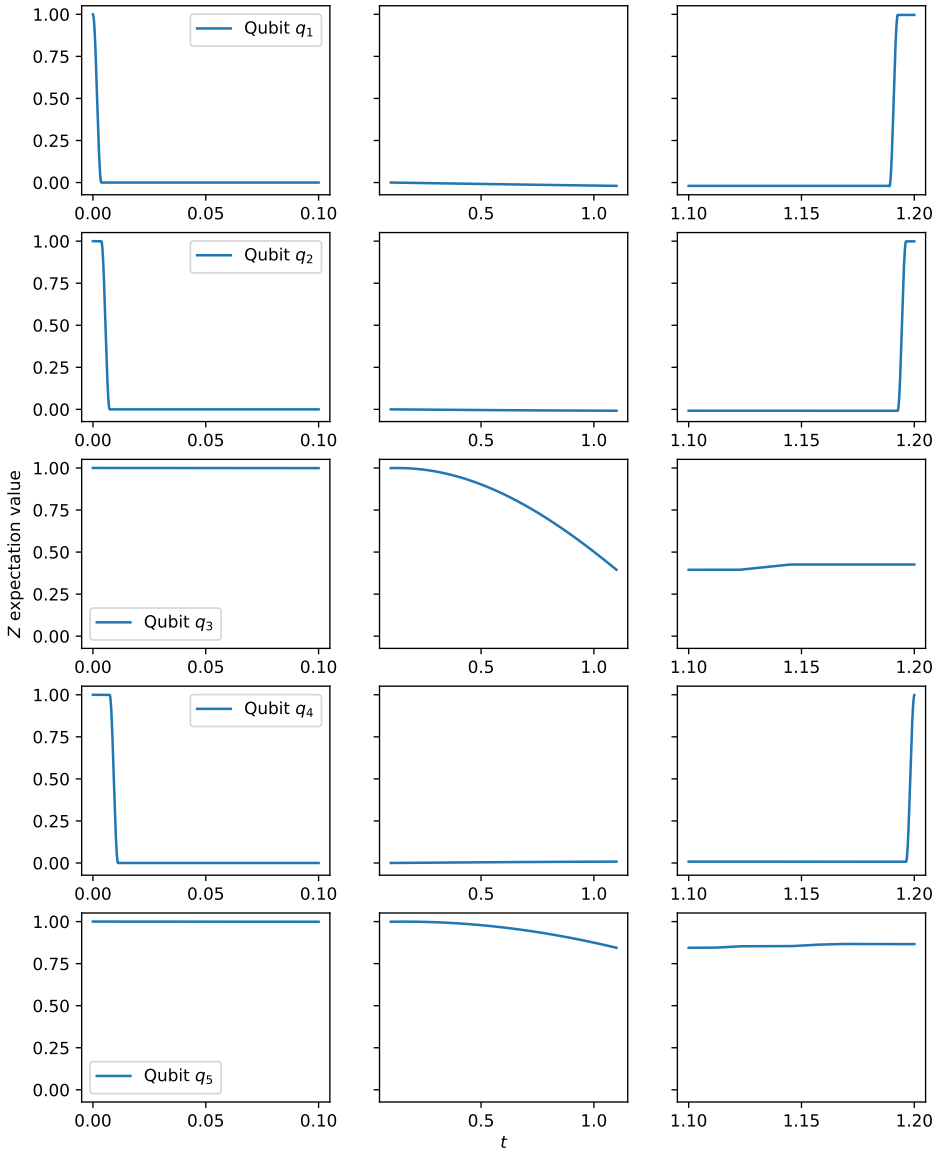

Figure 15: For the example with error-free encoding of $t_{enc.} = t_{dec.} = 0.1$ and an idle time of $t = 1$ with $J = 0.1, m = 0.1, g = 0.01$ the $\hat{Z}$-expectation values are shown for the qubit sites. The corresponding bosons are shown in figure 16. The simulation was done for $n_t = 50$ time-steps per circuit operation and $c = 10^{-2}$ as the singular value cutoff.

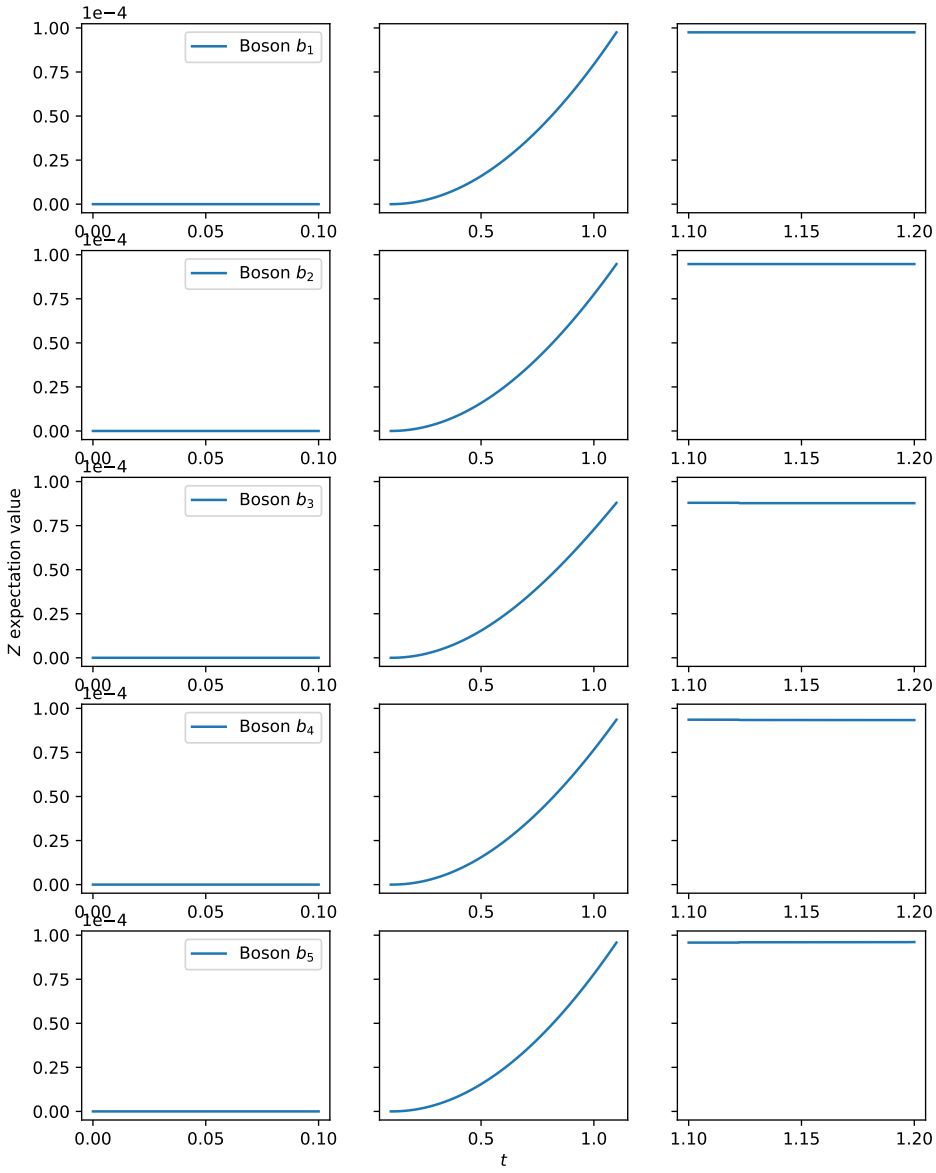

Figure 16: For the example with error-free encoding of $t_{enc.} = t_{dec.} = 0.1$ and an idle time of $t = 1$ with $J = 0.1, m = 0.1, g = 0.01$ the $\hat{Z}$-expectation values are shown for the boson sites. The corresponding qubits are shown in figure 15. The simulation was done for $n_t = 50$ time-steps per circuit operation and $c = 10^{-2}$ as the singular value cutoff.

$$
\begin{aligned}
\hat{R}_0 &= |00\rangle_{12}\langle 00| \otimes \hat{I}_3 \otimes |00\rangle_{45}\langle 00| \\
\hat{R}_1 &= |00\rangle_{12}\langle 00| \otimes \hat{Z}_3 \otimes |00\rangle_{45}\langle 01| \\
\hat{R}_2 &= |00\rangle_{12}\langle 00| \otimes \hat{I}_3 \otimes |00\rangle_{45}\langle 10| \\
\hat{R}_3 &= |00\rangle_{12}\langle 00| \otimes \hat{I}_3 \otimes |00\rangle_{45}\langle 11| \\
\hat{R}_4 &= |00\rangle_{12}\langle 01| \otimes \hat{I}_3 \otimes |00\rangle_{45}\langle 00| \\
\hat{R}_5 &= |00\rangle_{12}\langle 01| \otimes \hat{Z}_3 \otimes |00\rangle_{45}\langle 01| \\
\hat{R}_6 &= |00\rangle_{12}\langle 01| \otimes \hat{X}_3 \otimes |00\rangle_{45}\langle 10| \\
\hat{R}_7 &= |00\rangle_{12}\langle 01| \otimes \hat{X}_3 \otimes |00\rangle_{45}\langle 11| \\
\hat{R}_8 &= |00\rangle_{12}\langle 10| \otimes \hat{I}_3 \otimes |00\rangle_{45}\langle 00| \\
\hat{R}_9 &= |00\rangle_{12}\langle 10| \otimes \hat{X}_3 \otimes |00\rangle_{45}\langle 01| \\
\hat{R}_{10} &= |00\rangle_{12}\langle 10| \otimes \hat{Z}_3 \otimes |00\rangle_{45}\langle 10| \\
\hat{R}_{11} &= |00\rangle_{12}\langle 10| \otimes \hat{X}_3 \otimes |00\rangle_{45}\langle 11| \\
\hat{R}_{12} &= |00\rangle_{12}\langle 11| \otimes \hat{Z}_3 \otimes |00\rangle_{45}\langle 00| \\
\hat{R}_{13} &= |00\rangle_{12}\langle 11| \otimes \hat{X}_3\hat{Z}_3 \otimes |00\rangle_{45}\langle 01| \\
\hat{R}_{14} &= |00\rangle_{12}\langle 11| \otimes \hat{X}_3 \otimes |00\rangle_{45}\langle 10| \\
\hat{R}_{15} &= |00\rangle_{12}\langle 11| \otimes \hat{Z}_3 \otimes |00\rangle_{45}\langle 11|
\end{aligned}
$$

Table 2: This table shows the Kraus operators for the error recovery of the 5-qubit QEC code as used in [3]. $\hat{I}_3$, $\hat{Z}_3$ and $\hat{X}_3$ are the identity, X and Z operators applied to qubit q_3 (the one holding the information). While table 1 is sorted by which error occurred, this list is sorted by the result of the error syndrome measurement encoded in the projective measurements for qubits q_1, q_2, q_4 and q_5. Choosing $|0\rangle$ as the projection for these four qubits effectively resets them to the $|0\rangle$ state – enabling the re-encoding of the information qubit.