Open source Matrix Product States: Exact diagonalization and other entanglement-accurate methods revisited in quantum systems

Daniel Jaschke\textsuperscript{1} and Lincoln D. Carr\textsuperscript{1}

\textsuperscript{1}Department of Physics, Colorado School of Mines, Golden, Colorado 80401, USA

Tensor network methods as presented in our open source Matrix Product States software have opened up the possibility to study many-body quantum physics in one and quasi-one-dimensional systems in an easily accessible package similar to density functional theory codes but for strongly correlated dynamics. Here, we address methods which allow one to capture the full entanglement without truncation of the Hilbert space. Such methods are suitable for validation of and comparisons to tensor network algorithms, but especially useful in the case of new kinds of quantum states with high entanglement violating the truncation in tensor networks. Quantum cellular automata are one example for such a system, characterized by tunable complexity, entanglement, and a large spread over the Hilbert space. Beyond the evolution of pure states as a closed system, we adapt the techniques for open quantum systems simulated via the Lindblad master equation. We present three algorithms for solving closed-system many-body time evolution without truncation of the Hilbert space. Exact diagonalization methods have the advantage that they not only keep the full entanglement but also require no approximations to the propagator. Seeking the limits of a maximal number of qubits on a single core, we use Trotter decompositions or Krylov approximation to the exponential of the Hamiltonian. All three methods are also implemented for open systems represented via the Lindblad master equation built from local channels. We show their convergence parameters and focus on efficient schemes for their implementations including Abelian symmetries, e.g., $\mathbb{U}(1)$ symmetry used for number conservation in the Bose-Hubbard model or discrete $\mathbb{Z}_2$ symmetries in the quantum Ising model. We present the thermalization timescale in the long-range quantum Ising model as a key example of how exact diagonalization contributes to novel physics.
I. INTRODUCTION

Recently, exact diagonalization (ED) methods have seemed less important due to many available simulation methods for many-body quantum systems including the Density Matrix Renormalization Group (DMRG) [1], Quantum Monte Carlo methods [2, 3], dynamical mean field theory [4], and the truncated Wigner approach [5, 6]. The tensor network methods themselves contain a variety of tailored approaches such as Matrix Product States (MPS) [7, 8], Multi-scale Entanglement Renormalization Ansatz (MERA) [9], Tree Tensor Networks (TTN) [10], and Projected Entangled Pair States (PEPS) [11]. But with growing computing resources today, quantum states of larger and larger systems can be simulated without entanglement truncation. The first limitation of the simulation is the calculation of the propagator in the full Hilbert space. This propagator can be approximated. The second limitation is the initial state, which cannot be chosen as the ground state when systems are too large for exact diagonalization methods. Quantum computing and quantum information theory provide us with well-characterized states beyond the class of product or Fock states. With these techniques, system sizes can be pushed to 27 qubits with easily available computational resources, i.e., a single core on a computer. The boundaries of such simulations have recently been pushed to 45 qubits in the context of quantum supremacy [12]. Simulations, using also quantum circuits, have achieved simulations for 49 and 56 qubits [13] shortly after, then 64 qubits [14]. On the other hand, such methods become a valuable comparison for
benchmarking tensor network methods when exploring highly entangled states as recently studied with Quantum Elementary Cellular Automata (QECA) [15] based on the original proposition in [16], the Quantum Game of Life [17, 18], and for open quantum systems [19] among other physically important contexts. We foresee fruitful applications to the developing field of quantum simulators. These systems exist on a variety of platforms and form one significant part in the development of quantum technologies as proposed in the quantum manifesto in Europe [20, 21]. With exact diagonalization methods, quantum simulators with large entanglement have the possibility at hand to simulate systems up to a modest many-body system equivalent to 27 qubits. Furthermore, the area law for entanglement [22] can be violated for long-range interactions [23], and tensor network methods are likely to lose accuracy when the area law does not bound entanglement. Other possible applications are the emerging field of synthetic quantum matter.

The inclusion of open system methods allows us to approach thermalization of few-body quantum systems from different aspects, e.g., exploring whether or not a subsystem follows a thermal state after taking the partial trace of a pure state. A recent study considers the case of six $^{87}$Rb atoms [24] and is well within the scope of exact diagonalization. Research on quantum computing and quantum information can profit from such simulations provided through our library, too. For example, one can study mutual information matrices as done in a similar study with the MPS library [25]. Moreover, exact diagonalization methods are more accessible to new researchers in the field as they require less knowledge of the methods and how to tune them with the set of convergence parameters, e.g., associated with our MPS methods [26, 27]. Returning to exact diagonalization methods with the background of the many-body simulations in tensor networks advances them with regards to various aspects. For example, the usage of predefined rules to represent the Hamiltonian not only simplifies the problem but allows one to multiply the Hamiltonian with a state vector when the matrix of the Hamiltonian itself cannot be represented due to memory limitations. The Krylov method employs precisely this feature and overcomes limitations which are present in the Trotter decomposition with the primary purpose of evolving models with nearest-neighbor interactions. Starting with a certain number of predefined rules for the Hamiltonian, sophisticated models can be represented as well as with hand-built ED codes, but at a much lower cost, avoiding development overhead. A preliminary version of this library was used for the momentum-space study of the Bose-Hubbard model in [28]. We envision that open systems will profit from this approach enormously in the future to provide a variety of different Lindblad channels.

With these motivations in mind, we present three kinds of algorithms retaining the complete Hilbert space and their efficient implementations for closed systems. The first approach calculates the exponential for the propagator exactly. Second, we choose the Trotter decomposition for nearest-neighbor Hamiltonians allowing for larger systems due to the local propagators. In the third algorithm, we approximate the propagated state in the Krylov subspace allowing for the inclusion of long-range terms in the Hamiltonian. The interfaces work seamlessly with our package Open Source Matrix Product States (OSMPS) [27, 29] and therefore support both $U(1)$ and $\mathbb{Z}_2$ symmetries in the system. The measurements are adapted from tensor network methods and have a many-body focus: For example, we include the bond entropy, a value typically calculated for tensor networks obtained from the Schmidt decomposition for two bipartitions of the system. Local observables and two-site correlators are measured via the reduced density matrices, which are a more efficient approach in comparison to representing the observables on the complete Hilbert space. Beyond closed systems, we include methods for the Lindblad master equation [19, 30], which describes the dynamics of an open quantum system weakly coupled to its environment. Methods such as the matrix exponential, Trotter decomposition, or Krylov approximation can be adapted to density matrices. Quantum trajectories (QT) [31, 32] are another approach to simulate the Lindblad master equation. Quantum trajectories sample over a variety of pure state systems and therefore all of our three time evolution methods can be used for this approach.

In this Article, we discuss our versatile implementation of methods without entanglement truncation and show their convergence. The scaling of computational resources is compared for all three time evolution methods, where each method has its advantage: matrix exponentials avoid errors in the propagator; Trotter decompositions are ideal for nearest-neighbor interactions; and Krylov overcomes the limitation of the matrix exponential on the complete space without being limited to nearest-neighbor Hamiltonians. These approaches are applicable to various systems. Exact diagonalization can simulate a significant fraction of the number of ions of recent cold ions quantum simulator experiments, i.e., around one-tenth of [33]. This fraction is much closer than, e.g., in systems with ultracold atoms in optical lattices, where thousands to millions of atoms [34] are used in quantum simulators; the gap between ED methods and the experimental number of particles is then much larger. Looking at individual control of trapped ions, the state-of-the-art experiments are simulating 53 ions [35] and ED methods are even more relevant. Moreover, the development of new platforms for a universal quantum computer is likely to be tested on small scales first, and the corresponding numerical methods can contribute to a better understanding of those platforms. Thus, we envision multiple applications of our exact diagonalization codes beyond pure quantum computation and quantum information research. For example, exact diagonalization can simulate systems according to the full-spectrum Lindblad equation, where we consider a tensor network implementation of the same technique as prohibitive. We show that the thermalization timescale can both increase or decrease with the strength of the long-range interactions in the long-range quantum Ising model.

We introduce the construction of the system Hamiltonian in Sec. II. In Sec. III, we discuss the three different time evolution schemes before turning to the implementation of measurements in Sec. IV. We give an overview of the convergence of these methods in Sec. V and discuss the necessary modification with regards to open quantum system according to the Lindblad master equation in Sec. VI. We present the thermalization timescale with the full spectrum Lindblad master equation for the long-range...
quantum Ising model as key application in Sec. VII. After a benchmarking study in Sec. VIII of our package versus another standard package, i.e., QuTip, we conclude in Sec. IX.

II. CONSTRUCTION OF HAMILTONIANS AND STATICS

We first introduce the definition of systems in OSMPS and explain the exact diagonalization statics. The focus in this section is on closed quantum systems governed by the Schrödinger equation

$$\frac{\partial}{\partial t} |\psi(t)\rangle = -\frac{i}{\hbar} H |\psi(t)\rangle ,$$

where $|\psi\rangle$ is the wave function depending on the time $t$. $H$ represents the Hamiltonian of the system. We consider in the following a generic Hamiltonian with different terms defined as

$$H = \sum_{\zeta=1}^{n_S} \sum_{k=1}^{L} c_k O_{\zeta}^{[k]} + \sum_{\zeta=1}^{n_S} \sum_{k=1}^{L-1} c_k^{'} O_{\zeta}^{[B,\zeta]} O_{\zeta+1}^{[B,\zeta]} + \sum_{\zeta=1}^{n_S} \sum_{k=1}^{L-1} \sum_{k'=k+1}^{L} c_{k',k} O_{\zeta}^{[I,\zeta]} O_{\zeta}^{[I,\zeta]} ,$$

where $L$ is the size of the one-dimensional system. We take a look at the different terms one-by-one. The first term represents a local Hamiltonian term acting on a single particle or site in a system. The first sum over $\zeta$ allows us to have $n_S$ different local terms. The second sum runs over the different sites in the system, and each local term might have a spatially dependent coupling with the constants $c_k$. $O_{\zeta}^{[k]}$ is the corresponding operator acting on site $k$; actually, identities are padded on the sites $k' \neq k$ via the outer product, or Kronecker product, to represent the operator acting on the full Hilbert space. This term is necessary for many different systems. It represents the interaction with an external field in the quantum Ising model, and encodes the on-site interaction and chemical potential in the Bose-Hubbard model, just to name two examples. In detail, we take a look at the Bose-Hubbard model with the Hamiltonian

$$H_{BH} = -J \sum_{k=1}^{L-1} (b_k b_{k+1}^\dagger + h.c.) + U \sum_{k=1}^{L} n_k (n_k - 1) - \mu \sum_{k=1}^{L} n_k ,$$

where $b$ and $b^\dagger$ are the bosonic annihilation and creation operators and the number operator $n$ is defined as $n = b^\dagger b$. $J$ is the tunneling energy, and $U > 0$ is the repulsive interaction energy on each site. Thus, the sum over the local terms runs over $\zeta \in \{1, 2\}$. The first operators $O_{\zeta=1}^{[k]} = n_k (n_k - \mathbb{1})$ is a compound operator combining the number operators and the identity, all acting on site $k$. The chemical potential is the second local term, i.e., $O_{\zeta=2}^{[k]} = \mathbf{1}_k$. The next term is nearest-neighbor interactions, where the operator $O_{\zeta}^{[B,\zeta]}$ acting on site $k$ and $O_{\zeta+1}^{[B,\zeta]}$ on site $k+1$ can be different operators; hermiticity has to be fulfilled for all terms together. Looking at the example of the Bose-Hubbard in Eq. (3), the tunneling term is a nearest-neighbor term. Specifically, we have $\zeta \in \{1, 2\}$ for the bond term with $O_{\zeta=1}^{[B,\zeta=1]} = b_k$, $O_{\zeta=1}^{[B,\zeta=1]} = b_{k+1}^\dagger$, $O_{\zeta=2}^{[B,\zeta=2]} = b_k$ and $O_{\zeta+1}^{[B,\zeta=2]} = b_{k+1}$. The last term is a two-site interaction at arbitrary distance governed by a coupling $c_{k',k} = c(k' - k)$ depending only on the distance between the sites. This term extends many models where the nearest-neighbor interaction is an approximation, and a more accurate Hamiltonian includes further terms. The extended Bose-Hubbard model with long-range tunneling is one example. In the following, we refer to the three terms as site rules, bond rules, and infinite function rules. Appendix A contains a complete description of more possible terms in a Hamiltonian. We need to build the Hamiltonian on the complete Hilbert space to determine the statics. The same applies to the dynamics if using the complete matrix for taking the exponential or for the Krylov method. For systems without symmetry, this construction can be done purely using the Kronecker product. Using Python for our implementation, this Kronecker product is either done with *nump.kron* when building a dense Hamiltonian or with *scipy.sparse.kron* aiming for a sparse Hamiltonian. Dense methods allow one to calculate the complete set of eigenvalues and eigenvectors while sparse methods calculate only the eigenvalues and eigenvectors close to the ground state for the static case. Considering the entries in the final matrix in case of a site rule presented in Eq. (2), the computational scaling $F$ is then

$$F_{\text{site,dense}} = \mathcal{O}(d^{2L}) , \quad F_{\text{site,sparse}} = \mathcal{O}(d^{L+1}) ,$$

where $F$ is the number of floating point operations. We introduced $d$ as the local dimension of the Hilbert space of a single site, e.g., for the Ising model $d = 2$. The scaling in Eq. (4) includes the number of multiplications, but not any overhead keeping track of the sparse matrices. Similar calculations hold for the remaining rules defined in Eq. (2).

The picture gets more complicated when introducing symmetries. We support $U(1)$ and $\mathbb{Z}_2$ symmetries defined by their diagonal generator defined on the local Hilbert space. For example, the Bose-Hubbard model conserves the number of particles
Figure 1. Building the Hamiltonian with Symmetries. The process of obtaining the Hamiltonian $H$ with symmetry for a site rule can be split into three steps. (a) To build the Hamiltonian, we loop over the rows of $H$. (b) For each row, we find the corresponding row in the local matrix $O_{\text{local}}$ defining the term in the rule set. We mark the non-zero entries in this row. (c) Starting from the non-zero entries marked in step (b), we find the corresponding columns in $H$ and update the matrix element.

- a $U(1)$ symmetry — and the simulation can be executed in the symmetry sector with $N$ bosons. The eigenstates of the basis in the symmetry sector can either be passed as an argument, or the algorithm iterates once over all $D = d^L$ states checking if the state obeys the symmetry, which is possibly slow. Either way, we have a reduced basis of $D_S$ basis states $|i\rangle = |i_1, i_2, \ldots, i_L\rangle, \quad i \in \{1, \ldots, D_s\}$, where $i_k$ are the corresponding indices in the local Hilbert space. The complete wave function is written as $|\psi\rangle = \sum_i C_i |i\rangle$ with coefficients $C_i$ fulfilling the normalization constraint of the wave function. Neither can the local basis $|i_k\rangle$ be deduced easily from $|i\rangle$ nor $|i\rangle$ from a set of $|i_k\rangle$, because the complete Hilbert space is not a tensor product of the local Hilbert spaces. Therefore, a Python class `SymmetrySector` provides two key functions:

- A dictionary returns the index $|i\rangle$ providing the tuple of all local dimensions as key. Since dictionaries in Python have an underlying hash table, the lookup is fast. Henceforward we use the class method `__getitem__` for getting dictionary entries in Python, e.g., `SymmSec[(0, 0, 1)]`.

- `SymmetrySector` provides the local indices for the state $|i\rangle$. This implementation is achieved with a simple matrix $B$. The $i^{th}$ row contains the local indices for state $|i\rangle$. The vector of indices is obtained via the `__call__` attribute as `SymmSec(i)`.

Knowing the basic structure for the symmetries, we continue with the construction of the Hamiltonian in the example of a site rule. To set the matrix elements for the local operator $O$ on site $k$, we iterate over the $D_S$ basis states in the Hamiltonian. We choose to iterate over the rows $|i\rangle$ and obtain the corresponding local basis states $|i_1, i_2, \ldots, i_L\rangle$ via the dictionary. We need the index for the columns identified via $|i'\rangle$ and the local indices $|i_1', i_2', \ldots, i_L'\rangle$. Considering the structure $I \otimes O_k \otimes I$, we know that all indices except $k$ stay equal: $i_k' = i_k$, $\forall k' \neq k$. The index $i_k'$ can be determined by finding the columns with non-zero entries of row $i_k$ of $O_k$. Therefore, we loop next over the local dimension $i_k'$ of site $k$ where $|i_1, i_2, \ldots, i_k', \ldots, i_L\rangle$ corresponds to the entry in the column. We depict this procedure in Fig. 1. The computational time for building a site rule in the complete Hamiltonian $H$ then scales as

$$F_{\text{Build } H, \text{site}} = \mathcal{O}(LD_Sd).$$

This scaling can be further improved when iterating only over non-zero elements in the columns of the local operator $O_k$ instead of all the local dimension $d$. The fact of building a sparse or a dense Hamiltonian does not influence the algorithm apart from setting the matrix elements in a sparse or dense matrix. The interface for addressing a matrix entry in Python is equivalent for both approaches.

The ground and low-lying excited states can now be calculated via the corresponding numpy and scipy functions. For dense matrices, we obtain the full spectrum via `numpy.linalg.eigh` containing all eigenvalues, while in the sparse case only the
The ground state serves as the default initial state for the dynamics if no other state is specified. At present, the ground state is calculated from the complete Hamiltonian and therefore large systems are limited through the memory when representing the corresponding matrix.

III. METHODS FOR TIME EVOLUTION OF CLOSED SYSTEMS

Having explained the general setup of the system and how to obtain the ground state, we consider the more important question of time evolution. The first approach for the time evolution of a quantum state is (1) to calculate the propagator as a matrix exponential of the Hamiltonian on the complete Hilbert space. This method requires few modifications with regards to the ground state where we already have functions to construct the Hamiltonian. For time-dependent Hamiltonians, the procedure is especially expensive since the Hamiltonian and its propagator have to be calculated for every time step. Also, the Hilbert space grows exponentially for increasing system size $L$. Therefore, we consider two additional approximations: (2) the Suzuki-Trotter decomposition applies local propagators to the quantum state, and (3) the Krylov subspace approximates the state directly after applying the propagator. After a brief overview of two test models and their computational scaling in Sec. III A, we proceed to describe the three time evolution methods in Sec. III B to III D.

A. Computational Scaling in the Quantum Ising and Bose-Hubbard Model

We introduce the first two models which we use to study the scaling of CPU time $T_{CPU}$ for the different time evolution methods. First, we consider the 1D quantum Ising model [36, 37] defined as

$$H_{QI} = -\sum_{k=1}^{L-1} \sigma_k^x \sigma_{k+1}^x - h \sum_{k=1}^{L} \sigma_k^z,$$

(7)

which obeys $\mathbb{Z}_2$ symmetry. The operators $\sigma_k^x$ and $\sigma_k^z$ are the Pauli matrices acting on site $k$. The interaction energy is normalized to 1, and the value of the external field is $h$. $L$ is the number of sites in the system. Second, we study the Bose-Hubbard Hamiltonian introduced in Eq. (3), where the chemical potential $\mu$ can be disregarded when we use the $U(1)$ symmetry leading to a fixed number of bosons $N = \sum_{k=1}^{L} n_k$.

We present the scaling of the CPU time with increasing system size $L$ for the quantum Ising model in Fig. 2 (a). We see that the matrix exponential (ME) is limited first, where we simulated up to twelve qubits. Within the different algorithms using the matrix exponential on the complete Hilbert space, the choice of using dense matrices and $\mathbb{Z}_2$ symmetry is optimal. Sparse methods are labeled with $sp$. The symmetry reduces the size of the matrix to be exponentiated in the case of the quantum Ising model by a factor of 2 and therefore reduces the time significantly. The different memory requirements themselves are not the restricting limit. Other simulations have higher memory needs. For the Suzuki-Trotter or Krylov subspace method the use of symmetries is slower in case of the Ising model due to matrix-vector operations, which are needed to decode the basis. In the Krylov method with $\mathbb{Z}_2$ symmetry, we observe the step when switching to a memory-optimized but slower Krylov method for the 10th qubit. This memory-optimized method is necessary for larger systems as the set of Krylov vectors has to be stored, where each Krylov vector is of the same size as the state vector. The slow down for the memory-optimized version originates in reading and writing vectors to and from the hard disk. The Bose-Hubbard model with a local dimension of $d = 5$, corresponding to maximally four bosons per site, is shown in Fig. 2 (b). Simulations with $U(1)$ are at unit filling. Due to the large local dimension simulations are limited to smaller system sizes. The benefits of using the symmetry become relevant for the Krylov and Trotter method when going to large system sizes. The details on the implementation leading to those results for each method are described in the following three subsections.

B. Time Evolution Method 1: The Propagator as a Matrix Exponential

The approach for the time evolution without making any approximations apart from the time-dependence of the Hamiltonian $H(t)$ and the base numerical resolution inherent in any floating point computational scheme is to calculate the propagator as

$$U(t \rightarrow t + dt) = \exp (-iH(t + dt/2)dt).$$

(8)

We take the corresponding functions scipy.sparse.linalg.expm in the sparse case and scipy.linalg.expm in the dense case. The construction of $H$ is identical to the static case except that time-dependent coupling may be introduced via the OSMPS interfaces. We point out that the coupling constants are estimated for all methods at mid-time step $t + \frac{dt}{2}$, which
corresponds to a time-ordering of $O(dt^2)$. Finally, the state is propagated with a matrix-vector multiplication. The main problem is that taking the matrix exponential on the complete Hilbert space is expensive in memory and time. The scaling in memory is quadratic in $D$; the calculation time of the matrix exponential scales cubically with $D$. Therefore, computation time limits the system size before memory problems. We recall the dimension of the full Hilbert space $D$ scales exponentially with the number of sites $L$. In the following two subsections we explain how to simplify this exhaustive computational problem with approximate methods.

C. Time Evolution Method 2: Trotter Decomposition for Nearest-Neighbor Models

The main bottleneck so far is taking the exponential of a matrix due to CPU time, which grows exponentially with system size. In its original version, the Trotter decomposition allows us to approximate the exponential of the Hamiltonian on the complete Hilbert space with local exponentials on two sites. We present in the following an efficient scheme for applying these two-site propagators to the state, starting without symmetry. We point out that the method in this form is only applicable to Hamiltonians of local exponentials on two sites. We leave the upper limit for the summation intentionally open since it depends on open versus closed boundary conditions and even vs. odd system sizes $L$. This decomposition in Eq. (9) neglects the second order terms of $O(dt^2)$ according to the Baker-Campbell-Hausdorff equation [38]. We use higher-order Trotter approximations, see [8], in the implementations defined as

$$
\exp(-iHdt) \approx \exp\left(-idt \sum_k H_{2k-1,2k}\right) \exp\left(-idt \sum_k H_{2k,2k+1}\right)
= \exp(-idtH_o) \exp(-idtH_e). \tag{9}
$$

$H_o$ contains terms acting on odd sites and their right nearest neighbors, and likewise in $H_e$, except for even sites, with $H = H_o + H_e$. We leave the upper limit for the summation intentionally open since it depends on open versus closed boundary conditions and even vs. odd system sizes $L$. This decomposition in Eq. (9) neglects the second order terms of $O(dt^2)$ according to the Baker-Campbell-Hausdorff equation [38]. We use higher-order Trotter approximations, see [8], in the implementations defined as

$$
\exp(-iHdt) \approx \exp\left(-\frac{i}{2}dtH_e\right) \exp(-idtH_o) \exp\left(-\frac{i}{2}dtH_e\right), \tag{10}
$$
Figure 3. Tensor Operations on a State Vector without Symmetries. (a) The Hilbert space is a tensor product of local Hilbert spaces, and therefore we can transform our state vector $|\psi\rangle$ into a rank-$L$ tensor. (b) The application of a two-site propagator is reduced to a contraction with the two corresponding indices at a cost of $F = O(d^{L+2})$. (c) The implementation of the `tensordot` in numpy permutes the indices which we have to take into account in the algorithm. (d) One step of the second order Trotter decomposition for four sites contains the contraction of the propagators, depicted as rectangles.

Figure 4. Multiplying the Wave Function by a Local Matrix. The multiplication $|\psi'\rangle = O_{\text{global}} |\psi\rangle$ is carried out in multiple steps, where $O_{\text{global}}$ is defined on local Hilbert spaces as operator $O_{\text{local}}$ and padded with identities on all other sites. (a) We loop over the rows of the matrix $O_{\text{global}}$ and final state $|\psi'\rangle$ and select the corresponding row marking the selected row in blue. (b) In the next step, we find the corresponding row in the matrix $O_{\text{local}}$ defined on the local Hilbert space and mark the non-zero elements (red, green) in this row. (c) In the final, third step we find the corresponding column index for each non-zero element in $O_{\text{global}}$, which is identical to the row-index of the wave function $|\psi\rangle$ to be multiplied. Therefore, we can update the blue entry in the new vector $|\psi'\rangle$.

for the second order, and

$$
\exp(-idtH) \approx \exp(-ir_3dtH_e) \exp(-ir_4dtH_o) \exp(-ir_4dtH_e) \exp(-ir_4dtH_o) \\
\times \exp(-ir_2dtH_e) \exp(-ir_1dtH_o) \exp(-ir_2dtH_e) \\
\times \exp(-ir_4dtH_o) \exp(-ir_4dtH_e) \exp(-ir_4dtH_o) \exp(-ir_4dtH_e) \\
= \exp(-i\tau_4dtH_e) \exp(-i\tau_3dtH_o) \exp(-i\tau_4dtH_e) \\
\times \exp(-i\tau_3dtH_o) \exp(-i\tau_4dtH_e) \exp(-i\tau_4dtH_o) \\
\times \exp(-i\tau_4dtH_e) \exp(-i\tau_3dtH_o) \exp(-i\tau_4dtH_e) \\
\times \exp(-i\tau_3dtH_o) \exp(-i\tau_4dtH_e) \exp(-i\tau_3dtH_o) \exp(-i\tau_4dtH_e)
$$

for the fourth order, respectively. Other formulations of the fourth order approximation exist. We remark that the two-site terms either in $H_e$ or $H_o$ act on different sites and therefore commute. Their exponentials can thus be taken separately. Building the local exponentials, we then distribute the site rules into $H_e$ and $H_o$, where we choose to weight them with $1/2$ in each layer. This weighting applies to periodic boundary conditions, while in systems with open boundary conditions the first and last site receive a weight of 1.

Now we search for an efficient application scheme of these local propagators for systems without symmetries. A transformation of the state vector into a rank-$L$ tensor allows one to contract the rank-4 tensor of the local propagator with the corresponding
The approach via the Trotter decomposition in Sec. III C allows one to avoid the matrix exponential of the complete Hilbert space as long as there are no long-range interactions. As mentioned before in the introduction in Sec. I, this is not the case for the exponential rule, infinite function rule, and the MBString rule. An alternative is to approximate the exponential of the complete state at the end of one layer. The sketch for the second order decomposition can be found in Fig 3 (d), where the different steps described previously are shown. Considering the second order decomposition, the number of terms can be approximated by \( 3L/2 \) leading to an overall scaling of the method for one time step as

\[
F_{\text{Trotter-2 step}} = \frac{3L}{2} \mathcal{O}(d^{L+2}), \quad F_{\text{Trotter-4 step}} = \frac{11L}{2} \mathcal{O}(d^{L+2}).
\]

A similar result counting the number of layers is shown for the 4th order. Equation (11) shows that we have eleven layers in comparison to three layers in the second order represented in Eq. (10). In comparison, systems with symmetries scale as

\[
F_{\text{contract 2-site operator}} = \mathcal{O}(d^{L+2}).
\]

The trick of rewriting the wave function as a rank-\( L \) tensor does not work with symmetries since the basis is not a tensor product of local Hilbert spaces. Here, we take the same approach as when building the Hamiltonian and show a visualization of the algorithm in Fig. 4. First, we obtain the two-site Hamiltonian and build the subblocks, which conserve the symmetry. This reduction to subblocks is a minor improvement on the performance, taking exponentials on a smaller space and searching for non-zero elements only for the subblocks forming the propagator. In the matrix-vector multiplication, we loop first over the rows of the matrix consisting of the \( D_k \) basis states. The local propagator \( U_{k,k+1} \) is padded with identity matrices to the left and right, which do not modify the local basis in \( |v'\rangle = |i'_1, i'_2, \ldots, i'_L\rangle \) on any site \( k' \notin \{k, k+1\} \). Looping over the non-zero entries in the columns of the propagator \( U_{k,k+1} \) yields the corresponding local indices for \( i_k, i_{k+1} \). Then the basis state \( |i\rangle \) corresponding to the column in the full propagator can be easily obtained. The value is multiplied by the coefficient of \( C_i \) of \( |\psi(t)\rangle \) and added to \( C_i' \) of the new wave function \( |\psi'\rangle = |\psi(t+dt)\rangle \). Therefore, we have two copies of a wave function \( |\psi'\rangle \) in memory at the same time.

D. Time Evolution Method 3: Krylov Approximation

The approach via the Trotter decomposition in Sec. III C allows one to avoid the matrix exponential of the complete Hilbert space as long as there are no long-range interactions. As mentioned before in the introduction in Sec. I, this is not the case for the exponential rule, infinite function rule, and the MBString rule. An alternative is to approximate the exponential of the complete Hilbert space on a subspace, where the Krylov subspace approach [39] seems to be convenient for this purpose. In detail, it does not approximate the exponential, but instead directly the propagated state

\[
|\psi(t+dt)\rangle \approx \exp(-iHdt)|\psi(t)\rangle.
\]

The new state is written in terms of the basis vectors \( v_i \) of the Krylov subspace

\[
|\psi(t+dt)\rangle \approx \sum_{i=1}^{m} \xi_i v_i,
\]

where \( v_i \) is \( H^{i-1}v_i \) orthogonalized against all previous eigenvectors \( v_j, j < i \), and \( \xi_i \) is the matrix entry \( U_{i,1}^{[\text{Krylov}]} \) in the exponential transformed to the Krylov basis. This approach only saves computational effort when the number of Krylov vectors \( m \) is much smaller than the dimension of the Hilbert space. The algorithm can be controlled through convergence parameters. We provide four different modes: (0) the built-in scipy method based on the complete Hamiltonian is used for edilbmode=0. (1-3) These modes use the Krylov method of OSMPS, which can be tuned by the following convergence parameters. On the one hand, we can set a maximal number of Krylov vectors \( m \) building the basis. On the other hand, the algorithm can check a tolerance lanczos_tol, which stops the iterations before reaching \( m \). The modes are distinguished as follows:

(0): Built-in scipy method based on the complete Hamiltonian. The complete Hamiltonian is built from the rule sets.

(1): The Hamiltonian is built as a matrix and multiplied with the state vector to construct the Krylov method. Being similar to the scipy method, it cannot achieve the same performance as the scipy implementation in (0).

(2): The Krylov vectors are constructed from a multiplication of the rule set with the state vector and therefore does not need to allocate the memory to represent the matrix \( H \).
(3): The Krylov vectors are constructed from a multiplication of the rule set with the state vector. Instead of keeping all Krylov vectors in memory, this mode saves them temporarily on the hard disk to save memory and reads them on demand.

Figure 5 shows the CPU times for the different modes for five time-steps, where we introduce the double-well problem based on the Bose-Hubbard Hamiltonian introduced in Eq. (3). The potential at the center site splits the system into the two wells,

\[ H_{\text{DW}} = H_{\text{BH}} + V \cdot n_{(L-1)/2}, \]

which is always considered at a filling with \( N = (L-1)/2 \) particles, \( L \) odd. The fraction of Hilbert space needed in comparison to unit filling is smaller and allows us to simulate more sites than with unit filling. Mode 2 is preferable over 0, 1 and 3 for small systems from the viewpoint of CPU times in the case of the quantum Ising Hamiltonian \( H_{\text{QI}} \) from Eq. (7) shown in Fig. 5(a). For larger systems, the single matrix-vector multiplication in mode 0 may have a favorable scaling over the multiple contractions involved for the rule set in mode 2. For the Bose-Hubbard double-well problem shown in (b), mode 0 is always preferable. Mode 3 is intended for large systems and can be scaled up to 27 qubits without encountering memory limitations, derived below in Eq. (19).

Apart from the implementation of the Krylov algorithm, we need an efficient multiplication \( H | \psi \rangle \). Seeking limits beyond twenty qubits, building the Hamiltonian in the Hilbert space is not appropriate. Therefore, we implement directly the matrix-vector product of a rule set with a vector. In the case without symmetries, we transform \( | \psi \rangle \) into a rank-\( L \) tensor and then use np.tensordot in addition to a permutation to obtain the results similar to the procedure in the Trotter decomposition, which requires nested loops over every rule and site. The same applies to the case with symmetries: instead of multiplying the propagator with the state, we apply the multiplication with the matrices of the Hamiltonian rule sets looping over all rules sets and sites. Considering the case with \( n_s \) site rules and \( n_b \) bond rules without using symmetries, the scaling for \( H | \psi \rangle \) is equal to

\[ \mathcal{F}_{\text{Rule set} \times | \psi \rangle} = Ln_s \mathcal{O}(d^{L+1}) + 2(L-1)n_b \mathcal{O}(d^{L+1}) \approx L(2n_b + n_s) \mathcal{O}(d^{L+1}). \]  

(19)

For one time step, we have to build \( m \) Krylov vectors. The leading term in the computational scaling should be the matrix-vector multiplication; therefore, we have

\[ \mathcal{F}_{\text{Krylov step}} = mL(2n_b + n_s) \mathcal{O}(d^{L+1}). \]  

(20)

We introduce the number of infinite function rules as \( n_i \). Replacing the \((L-1)\) interactions of a bond rule with the \( L(L-1)/2 \) interactions of an infinite function rule \((n_i = n_b)\), the scaling for one time steps turns into

\[ \mathcal{F}'_{\text{Krylov step}} = mL \left[ n_s + (L-1)n_i \right] \mathcal{O}(d^{L+1}). \]  

(21)
assuming the phase term is the identity and not contracted as done in the implementation. For fermionic systems with phase operators, the number of floating point operations would further increase.

If we compare the Krylov method to the 4th order Trotter decomposition for the nearest neighbor case of the quantum Ising model, we estimate that the Krylov takes longer than the Trotter decomposition by a factor of

$$\frac{\mathcal{F}_{\text{Krylov}}}{\mathcal{F}_{\text{Trotter}}} = \frac{2d}{11} m \frac{2n_b + n_a}{O(mn_b)}.$$  

Plugging in the actual numbers with $n_b = n_a = 1$ and $d = 2$, we obtain finally a factor of $3m/11$. We can take the calculation times of Fig. 2 without $Z_2$ symmetry and obtain a factor of approximately 30 between the two methods for large systems. That would yield $m \approx 110$, which is above the maximal number specified in the convergence parameters as 100.

In conclusion, these scaling considerations can only serve as estimates. But we point out why long-range simulations are so expensive, according to Eq. (22). If we assume that we replace the nearest-neighbor interactions in the quantum Ising model with long-range interactions, i.e., we replace the bond rule by one infinite function rule without phase operators, the resulting long-range model has the following factor in comparison to the nearest-neighbor Trotter decomposition:

$$\frac{\mathcal{F}_{\text{Krylov}}^{\text{exp}}}{\mathcal{F}_{\text{Trotter}}^{\text{NN}-4}} = \frac{2m}{11d} [n_a + (L-1)n_i] \approx \frac{2m}{11d} (L-1)n_i,$$

where the approximation in the last steps assumes that the site rule does not contribute the major part to the total computation. The quantum Ising model has one non-local rule $n_i = 1$ and a local dimension $d = 2$, which leads to $m(L-1)/11$. Taking, for example, $L = 18$ and $m = 50$ for the long-range Ising model, one could calculate in the same compute time the nearest-neighbor Ising model with a Trotter decomposition for $L = 24$, since $m(L-1)/11 \approx 77.3 \approx 2^6$.

IV. EFFICIENT MEASUREMENTS OF PURE STATES

As our focus is on many-body systems, we draw our attention to the corresponding measurement procedures. In addition to local observables and correlation measurements, we are especially interested in the entanglement between two subsystems known in MPS methods as bond entropy. For the local measurement, we use the reduced density matrix $\rho_k$ for site $k$, tracing out all other sites. This operation scales as $O(d^{L+1})$. The actual measurement $\text{Tr}(O_k \rho_k)$ is negligible with $O(d^3)$. The first alternative method is to use the approach of the Trotter decomposition, i.e., contracting a local operator with the state vector leading to $|\psi'\rangle = (O_k |\psi\rangle)$. Then, we calculate the measurement outcome $\langle \psi | \psi' \rangle$ at a cost of $O(d^{L+1}) + O(d^L)$. Using the observable $O_k$ padded with identities in the complete Hilbert space has a leading scaling of $O(d^{2L})$ for the measurement $\langle \psi | O_k | \psi \rangle$, let alone the tensor products to obtain $O_k$. These considerations are valid for non-symmetry conserving systems.

To summarize, local and two-site density matrices are an efficient approach to measurements. In addition, we need the bond entropies. The scaling with system size $L$ for the complete sets of single-site, two-site density matrices, the energy, and bond entropies are shown in Fig. 6. The CPU times show that the symmetry-adapted algorithms are slower for all four different measurements in the case of the quantum Ising model and its $Z_2$ symmetry. The Bose-Hubbard model shows a more diverse picture. In the following, we explain the details of the algorithms.

The simplest way to build the single-site density matrix is to use the function $\text{np.tensordot}$ to contract $|\psi\rangle$ and $\langle \psi |$ over all indices not appearing in the reduced density matrix after transforming the vectors into rank-$L$ tensors. This function is helpful for systems without symmetries, but we lay out another path to lead to the calculation of reduced density matrices for systems with conserved quantities. Assuming we calculate the density matrix of the first site in Python starting from a state vector $|\psi\rangle$, we have the following order in the row-major memory\footnote{The memory in a computer is linear, but matrices have two dimensions. Row-major memory stores matrices row-by-row. Column-major memory stores matrices column-by-column. Each memory order is generalized to higher rank tensors. Python uses row-major memory order; therefore, we use this order for our example.}

$$|\psi\rangle = |0\rangle_1 \otimes \ldots |0\rangle_{2,...,L} |1\rangle_1 \otimes \ldots |1\rangle_{2,...,L} = [A][B],$$

where we focus on qubits in this example and refer to the blocks as $[A][B]$. The reduced density matrix of the first site is then

$$\rho_1 = \left( \sum A A^* \sum B A^* \sum B B^* \right).$$

The $^*$ takes the complex conjugate of the elements. Each of the entries in $\rho_1$ can be calculated efficiently using the $\text{numpy.sum}$ and the element-wise multiplication. In fact, it is sufficient to calculate the upper (lower) triangular part of $\rho$ and use the property $\rho = \rho^T$ to fill the lower (upper) triangular part of $\rho$. Reusing the previous method for $k$ with $k \neq 1$ we choose the following path to avoid writing specific summations for different sites $k$:
We show CPU times for the construction of local and two-site density matrices \( \rho_k \) and \( \rho_{k,j} \), the bond entropy \( S \), and the energy measurement \( E \). All times are determined on a 2x Intel Xeon E5-2680 Dodeca-core) 24 Cores 2.50GHz. (a) We observe from the scaling for the quantum Ising model with and without \( Z_2 \) symmetry that the measurements with symmetry are slowed down. The jagged behavior for every second data point in the bond entropy originates in the fact that every even system size and the next bigger odd system size have their eigenvalue decomposition on a matrix of the same dimension, although the odd system size has two of them. But the dimension increases again for the next even system size. (b) For the Bose-Hubbard model, we show the scaling without symmetry and with \( U(1) \) symmetry at unit filling. Due to the local dimension of \( d = 5 \), system sizes are especially limited if the \( U(1) \) symmetry is not used.

1. Reshape the vector \( \ket{\psi} \) into a rank-\( L \) tensor with the local dimension (numpy.reshape).
2. Permute the index or indices, which are not traced out, to the front (numpy.transpose).
3. The reduced density matrix can now be calculated as explained for the first site. We point out that BLAS subroutines scipy.blas.dsyrk and scipy.blas.zherk only build the upper or lower triangular matrix; they are not used right now, but may enhance the computation time for systems without symmetry.

The previous method does not work directly for the reduced density matrices of states in a specific symmetry sector. The first possibility is to map the state vector from the Hilbert space with symmetries back to the complete Hilbert space without symmetries. This method has the disadvantage that the memory needs might increase drastically in addition to the fact of multiplying a lot of zeros. For systems where only a small fraction of the complete Hilbert space is needed, this increase in memory imposes a problem. For example, the Bose-Hubbard model at unit filling for a local dimension of \( d = 5 \) and fourteen sites corresponds to approximately 23.67 qubits. The complete Hilbert space has a dimension of \( 5^{14} \), which corresponds to approximately 32.51 qubits. We can overcome this problem in two ways. We may use booleans to index the complete Hilbert space into indices being part of the symmetric Hilbert space (True) and not belonging to the symmetry sector specified (False).

Thus, we use less memory: a factor of 16 (8) in comparison to complex-valued (real-valued) states knowing that numpy stores each boolean in a byte, not a bit. Assuming this approach is not always enough gain, we use a sparse structure. There are no sparse structures for tensors or vectors in scipy, so we save the state vector in a sparse matrix. Therefore, the permutation has to be accounted for while writing the sparse structure. We propose, again in the simple example of qubits and a single-site density matrix for site \( k \), the following steps focusing solely on the method with the sparse matrix:

1. For each basis state \( \ket{i} \) we need the index in the Hilbert space only considering the traced out or retained sites. For the new Hilbert space of the retained sites, we create the vector

\[
v_{\text{retain}} = (0, \ldots, 2^0, \ldots, 0),
\]

where the entry \( 2^0 \) is at the position \( k \). Retaining more than one index, the entries for the \( l \) retained sites would be \( 2^{l-1}, \ldots, 2^0 \). In contrast, the vector for the traced out part of the Hilbert space is

\[
v_{\text{trace}} = (2^{L-2}, 2^{L-3}, \ldots, 2^{L-k}, 0, 2^{L-k-1}, \ldots, 2^0),
\]
where the entry 0 is at the position \( k \). The indices in the corresponding complete subspace disregarding the symmetry can be obtained with the matrix \( \mathcal{B} \), which contains the basis, via a matrix-vector multiplication:

\[
i_1 = \mathcal{B} \cdot \mathbf{v}_{\text{retain}}, \quad i_2 = \mathcal{B} \cdot \mathbf{v}_{\text{trace}}.
\] (28)

\( i_1 \) and \( i_2 \) are vectors. Building the vectors is cheap and scales linearly with the number of sites. The expensive part is the two matrix-vector multiplications at \( \mathcal{O}(D_S L) \).

2. Looping over all basis states \( i \) in the reduced density matrix, we create for each state \( i \) a sparse vector. The indices in the present (symmetry-adapted) Hilbert space \( i_{\text{match}} \) are identified via the comparison \( i_1 = i \). The sparse vector contains the entries of \( \psi_{i_{\text{match}}} \) at the corresponding indices in the traced out Hilbert space \( i_2(i_{\text{match}}) \). If we retain \( l \) sites, they approximately have a scaling of \( \mathcal{O}(d^l D_S) \), where \( d^l \) is due to the loop over the basis states and \( D_S \) corresponds to the comparisons \( i_1 = k \).

3. Looping over all upper triangular entries \( \rho_{i,i'} \) of the density matrix, each entry is calculated via the dot product of the sparse vectors with indices \( i \) and \( i' \), where we have to take the complex conjugated vector for \( i' \). Although we loop only over the upper triangular part, the leading scaling for the loops is still \( \mathcal{O}(d^l) \). For the vector-vector multiplication, we can only give an upper limit of \( \mathcal{O}(D_S) \) valid for the limit \( l = 0 \) leading overall to \( \mathcal{O}(d^2 D_S) \).

So the overall scaling is \( \mathcal{O}(D_S L) + \mathcal{O}(d^{2l} D_S) \). The sparse structure in scipy is limited to a dimension of \( 2^{31} - 1 \), which becomes an issue if the complete Hilbert space of the sites traced over exceeds this value. The example above with fourteen sites with a local dimension of \( d = 5 \) exceeds this limit. Instead of using the indices of the complete Hilbert space, we build a symmetry-adapted basis for the reduced density matrix and use its indices, which can never exceed the number of basis states of the original quantum state. A similar algorithm can be formulated if the resulting density matrix should keep the symmetry-adapted space to the extent possible.

For the calculation of the bond entropy we employ the following algorithm; we do not use the characteristic \( \rho = \rho^\dagger \) so far. Starting in the middle of the system we create the density matrix for the left bipartition \( 1, \ldots, l \) as matrix-matrix multiplication

\[
\rho_l = MM^\dagger,
\] (29)

where \( M \) is the vector \( |\psi\rangle \) reshaped into the dimension \( d^l, d^{L-l} \). In the case of symmetries present, we create \( M \) as a sparse matrix. An eigenvalue decomposition yields the bond entropy. The next reduced density matrix is obtained via a partial trace

\[
\text{Tr}_{l+1} \rho_{l+1} = \rho_l.
\] (30)

This approach should have a favorable scaling with \( d^{2l+1} \) additions in addition to a permutation to form blocks of the memory along \( \text{np.transpose}((0, 2, 1, 3)) \) in regard to calculating the reduced density matrix for every site with a matrix multiplication scaling with \( d^{L+1} \). The summation for each of the \( d^l \) entries runs only over the \( d \) diagonal entries of the memory block. In the case with symmetries, the scaling for the matrix multiplication does not hold. The second step takes care of the right half of the system building an initial, reduced density matrix for \( l + 2, \ldots, L \), that is,

\[
\rho_l' = M' MM^\dagger.
\] (31)

In the following we trace out the first site in the reduced density matrix instead of the last. The permutation with \( \text{np.transpose}((1, 3, 0, 2)) \) delivers the entries to be summed over in blocks of the memory, although we recall that only the diagonal elements are summed over. Overall the bond entropy can be obtained equally well with singular value decomposition of the matrix \( M \) build for each splitting. In the tensor network section of OSMPS, eigendecompositions are preferred over SVD, so we use them in this case as well.

V. CONVERGENCE OF TIME EVOLUTION METHODS

Finally, we are interested in the convergence of those methods, especially in the error of the Trotter decomposition and Krylov evolution in comparison to the matrix exponential on the complete Hilbert space (ME). For the convergence studies we consider the following four kinds of errors taking into account local measurement, correlation measurements between operators on two sites, the energy \( E \) and the entanglement specified through the bond entropy or Schmidt entropy \( S \). These error measurements are defined as:

\[
\epsilon_{\text{local}} = \max_k \mathcal{D}(\rho_k, \rho_k^{\text{ME}}), k = 1, \ldots L
\] (32)

\[
\epsilon_{\text{corr}} = \max_{(k,j)} \mathcal{D}(\rho_{k,j}, \rho_{k,j}^{\text{ME}}), k, j = 1, \ldots L, k \neq j
\] (33)

\[
\epsilon_E = |E_E - E_{\text{ME}}|
\] (34)

\[
\epsilon_S = |S(L/2) - S_{\text{ME}}(L/2)|,
\] (35)
where the superscript ME specifies that the matrix exponential was used as reference method. The trace distance $D$ and the bond entropy $S$ are defined as

$$D = \frac{1}{2}|\rho - \rho'|, \quad |A| = \sqrt{A^\dagger A}$$

$$S(l) = -\sum_j \Lambda_j(l) \log(\Lambda_j(l)).$$

$A$ is a matrix used to define the norm and is replaced in this case with the difference of the two density matrices $\rho - \rho'$. $\Lambda_j(l)$ are the eigenvalues of the reduced density matrix of the subsystem reaching from site 1 to $l$. An alternative distance measure would be the infidelity $I = 1 - \text{Tr}\sqrt{\rho A \rho B \sqrt{\rho A}}$, but seems to introduce an additional instability taking the square root of a matrix twice. For the convergence rates, we use

$$r = \frac{1}{\ln(\alpha)} \ln \left( \frac{\epsilon(\alpha dt)}{\epsilon(dt)} \right),$$

where $\alpha$ is the proportion between the size of the time steps for the two evolutions carried out with time steps $dt$ and $\alpha \cdot dt$. $\epsilon$ is the error, e.g., one of the measures introduced in Eqs. (32) to (35). We consider the error with regards to the simulations with the matrix exponential and the smallest time step. We consider two different models and two scenarios for the dynamics. We recall that we introduced the definitions of the Ising Hamiltonian $H_{\text{Is}}$ in Eq. (7) and the Bose-Hubbard model with $H_{\text{BH}}$ in Eq. (3). The two scenarios considered are as follows.

1. A linear quench in the paramagnetic (Mott insulator) phase for the Ising (Bose-Hubbard) model starting at $h = 5.0$ ($U = 10$) and reaching after 0.5 time units $h = 4.5$ ($U = 8$). The time is in units of the interaction $\sigma^z \sigma^z$ (tunneling strength $J$).

2. A sudden quench in the paramagnetic (Mott insulator) phase evolving the ground state of the Ising (Bose-Hubbard) model $h = 5.0$ ($U = 10$) with a value of $h = 4.5$ ($U = 8$) for 0.5 time units.

The results for the scenario (1) in case of the Bose-Hubbard model can be found in Fig. 7. We point out that the Krylov method has a similar error to taking the complete matrix exponential. Greater errors in the Trotter decomposition are due to non-commutating terms in the decomposition originating from Eq. (10). For the example in Fig. 7, we obtain a convergence rate of $dt^2$ for the second order Trotter decomposition (2.15, 2.0), the fourth order Trotter decomposition (4.21, 2.05), and the Krylov methods (2.73, 2.0), where we consider the pairs $(dt = 0.01, dt' = 0.1)$ and $(dt = 0.001, dt' = 0.01)$ to evaluate the rate of convergence according the distance of the single site density matrices. The limitation to the second order convergence rate for all methods can be explained since the time-ordering is correct up to second order for the time-dependent Hamiltonian. The Krylov method is then exact independent of the time step, and the fourth-order Trotter convergence rate is 4.59 for $(dt = 0.01, dt' = 0.1)$ before reaching machine precision. Additional plots for the Ising model and other scenarios can be found in App. E.

VI. OPEN SYSTEMS IN UNTRUNCATED SPACE

The methods so far described are applicable for closed systems described by pure states. These methods ignore the fact that any quantum system is coupled to some extent to its environment. The Lindblad master equation is one way to describe an open quantum system through the evolution of its density matrix $\rho$. The assumptions are that the system is never entangled with its environment, the correlation time of the environment is small in comparison to the evolution time, and the evolution time itself is small in comparison to the equilibration time of the system. These approximations apply if the environment is large and has no memory. They are referred to as Born-Markov approximation and secular approximation. In addition to Hamiltonian dynamics described via the von Neumann equation, the evolution is described via $\eta$ Lindblad channels acting on different sites $k$:

$$\dot{\rho} = -\frac{i}{\hbar} [H, \rho] + \sum_{\eta, k} L_{\eta, k} \rho L_{\eta, k}^\dagger - \frac{1}{2} \left\{ L_{\eta, k}^\dagger L_{\eta, k}, \rho \right\} = -\frac{i}{\hbar} [H, \rho] + \sum_\nu L_\nu \rho L_\nu^\dagger - \frac{1}{2} \left\{ L_\nu^\dagger L_\nu, \rho \right\},$$

where the indices $\eta$ and $k$ can be combined in another index $\nu = (\eta, k)$. One of our additional rule sets for the open system is a local Lindblad operator

$$\text{Lind}1 \text{ rule} : \sum_{k=1}^L w \cdot c_k \cdot L_i,$$
which would correspond to one channel $\eta$. We can always choose zero Lindblad channels and have an evolution of a possibly initially mixed system under the von Neumann equation

$$\dot{\rho} = -\frac{i}{\hbar} [H, \rho].$$  \hfill (41)

In the following, we describe two approaches to simulate the Lindblad master equation. On the one hand, we can simulate the whole density matrix $\rho$ as represented in Eq. (39). Therefore, $\rho$ is represented as superket vector $|\rho\rangle$ in the Liouville space obtaining a Schrödinger-like equation. The superket vector is the density matrix written as a vector. A $D \times D$ density matrix turns into a vector with $D^2$ entries. But first, we lay out the quantum trajectories, which average over many simulations. Each simulation describes a single realization of what possibly could happen in an experiment.
A. Quantum Trajectories

The method of quantum trajectories has been proposed to avoid simulation of the complete density matrix at the cost of sampling over many realizations of a simulation with pure states. The algorithm used can be described by the following steps [31, 32, 40]:

1. Throw a random number \( r \) and evolve under the effective Hamiltonian \( H_{\text{eff}} \) until the norm drops below \( r \),

\[
H_{\text{eff}} = H - \frac{1}{2} \sum_{\nu} L^\dagger_\nu L_\nu .
\]

2. Throw another random number \( r_\kappa \) and calculate the probabilities \( p_\nu \) to apply any possible Lindblad operator and the cumulative probability \( P_\nu \),

\[
p_\nu = \langle \psi | L^\dagger_\nu L_\nu | \psi \rangle , \quad P_\nu = \frac{\sum_{j=1}^{\nu} p_j}{\sum_{j=1}^{\nu_{\text{max}}} p_j} .
\]

3. Apply Lindblad operator \( \kappa \) with \( P_\nu < r_\kappa \leq P_\nu \) and renormalize \( |\psi'\rangle = L_\kappa |\psi\rangle \). Restart with step (1).

This approach can be used with the matrix exponential on the complete Hilbert space, the Trotter and Krylov evolution.

B. Liouville Space

The Liouville space is a mapping from the Hilbert space defined over

\[
O' \rho O \rightarrow O' \otimes O^T |\rho\rangle \rangle ,
\]

where the original density matrix \( \rho \) is now represented as a vector, the super-ket \( |\rho\rangle \rangle \). The superscript \( T \) is the transpose of the matrix. A quick derivation of the transformation is given in App. F. The corresponding equation governing the evolution of the density matrix is Schrödinger-like,

\[
\frac{\partial}{\partial t} |\rho\rangle \rangle = \left[ -\frac{i}{\hbar} \left( H \otimes I - I \otimes H^T \right) + \sum L_k \otimes L^*_k - \frac{1}{2} \left( L_k L^\dagger_k \otimes I + I \otimes L^\dagger_k L_k^T \right) \right] |\rho\rangle \rangle .
\]

The short-hand notation is \( \partial/\partial t |\rho\rangle \rangle = L |\rho\rangle \rangle \), where \( L \) represents Hamiltonian part and dissipative part of the time evolution in Liouville space. Thanks to the structure we can reuse two of the previous methods with small modifications. In the first method presented, we obtain the propagator via the matrix exponential. Calling appropriate methods for non-hermitian matrices — the matrix in Eq. (45) is not necessarily hermitian due to the term \( L_k \otimes L^*_k \) — we can reuse the method. For the Trotter decomposition, the same strategy is necessary for the local two-site propagators. The matrix exponential for the two site propagator is now of dimension \( d^4 \times d^4 \) instead of \( d^2 \times d^2 \) for state vectors. Figure 8 shows how to apply the two-site operators acting on the density matrix formatted as a rank-(2\(L\)) tensor. Based on the contraction over four indices for two sites, the application scheme for the Trotter decomposition can be adapted. In contrast to the pure system, we permute the tensor after every contraction back to its original order. The Krylov method used previously has to be modified for non-hermitian matrices since the exponential of the matrix in the Krylov subspace is no longer tridiagonal, but of upper Hessenberg form.

C. Error Analysis in Case Study 1: Coupled Cavities

To check the convergence of the code, we analyze two coupled cavities, or photon Josephson junctions (PJJs), described through the Jaynes-Cummings model with loss [41]. The Hamiltonian of the system without a loss is defined as

\[
H_{\text{PJJ}} = (\omega_q n_1 + \omega_c n_2 + \omega_c n_3 + \omega_q n_4) + g \left( \sigma^+_1 a_2 + \sigma^-_1 a_2^\dagger + a_3^\dagger \sigma^-_4 + a_3 \sigma^+_4 \right) - J \left( a_1^\dagger a_3 + a_2 a_3^\dagger \right) ,
\]

where \( \omega_q = 1.0 \) (\( \omega_c = 1.0 \)) is the frequency of the qubit (cavity) and \( n_k \) are the number operators for the excitations on the corresponding site \( k \). \( g = 0.48 \) is the coupling constant between cavity and qubit; the cavities are coupled to each other at a strength \( J \) representing the tunneling of photons. The transition from the ground (excited) state to the excited (ground) state
for the qubits is described with the operator $\sigma^+ (\sigma^-)$. The creation (annihilation) operator for the photons in the cavity is $a^\dagger (a)$. We recall that $n_{1,4} \in \mathbb{R}^{2\times 2}$ while the dimension of $n_{2,3}$ varies with the number of photons allowed in the cavity. We add a spontaneous emission from the excited state to the ground state for the qubits. The second type of system-environment interaction is the loss of photons. Both processes assume that the environment is the vacuum and only absorbs excitations, but cannot provide them to the system. We can describe this system with the following Lindblad master equation:

$$
\dot{\rho} = -i[H_{P,13}, \rho] + \sum_{k\in\{1,4\}} \gamma_k \sigma_k^- \rho \sigma_k^+ - \frac{\gamma_k}{2} \{\sigma_k^+ \sigma_k^-, \rho\} + \sum_{k\in\{2,3\}} \gamma_k a_k \rho a_k^\dagger - \frac{\gamma_k}{2} \{a_k^\dagger a_k, \rho\}.
$$

(47)

Defining the total number of excitations as $N(t) = \sum_{k=1}^4 n_k$ and choosing a constant $\gamma = \gamma_k$ for the coupling of the system to the environment via the Lindblad operator, one can show that the following relation holds:

$$
N(t) = N(t=0) \exp(-\gamma t).
$$

(48)

We turn to the error analysis. First, we choose to simulate the case $\gamma = 0$ with the open system code, which is, in fact, the evolution according to the von Neumann equation for density matrices defined in Eq. (41). This simulation should reproduce the results for pure state $|\psi\rangle$ and corresponding code. We characterize the error again in terms of the maximal distance between all single site density matrices and two site density matrices, the energy, and bond entropy as defined in Eqs. (32) to (35). Table I lists the corresponding errors for our default settings with a total time $T = 10$ with a time step of $dt = 0.01$. The errors are for the final measurement at $T = 10$. We start with one photon in the right cavity and all other sites in the ground state. Therefore, we can cut off the Hilbert space to a local dimension $d = 2$. We remark that the evolution of the density matrix does not provide the bond entropy and therefore cannot be compared to the result of the pure state. Furthermore, zero error is to be interpreted as numerical zero. Table II shows the errors for the evolution of the open system with $\gamma = 0.05$, where we introduce also the error in the total number of excitations following Eq. (48) as

$$
\epsilon_{N(t)} = |N_{EDLib}(T) - N(t=0) \exp(-\gamma T)|.
$$

(49)

We recognize the larger error for quantum trajectories in Table II. This error is due to the finite number of different trajectories. The standard deviation scales with $1/\sqrt{N_{QT}}$ according to the law of large numbers. Errors related to the algorithm are at a much lower magnitude for proper parameters and would only become visible for very large $N_{QT}$. We may illustrate this behavior with the following example: We assume there is one particle and we pick the point in time where this particle decayed with probability $0.5$ leading to a coin-toss scenario. The normalized standard deviation for the binomial distribution is then $(\sqrt{4N})^{-1}$. For 500 trajectories, we have 0.022. The standard deviation in Tab. II is within this order of magnitude. Errors here are for $T = 10$; therefore, the statistics are slightly different.

D. Error Analysis in Case Study 2: Double-Well

In the second case study, we check the error for simulations with symmetries, where we choose a Bose-Hubbard model with $U(1)$ symmetry for the number conservation. We introduced the double-well problem before in Eq. (18) with its Hamiltonian $H_{DW}$. For the comparison of the evolution of the von Neumann equation of the density matrix, see Eq. (41), to the pure state evolution we choose a system size of $L = 7$ with three bosons over a time of $T = 20$ and a time step of $dt = 0.01$. The height
Table I. Error Analysis for the Density Matrix in the von Neumann Equation. We simulate the closed system with the open system code of the library using the von Neumann equation in the Liouville space (Liou). As a reference, we use the closed system evolved with the matrix exponential (ME) to obtain errors for the maximal distances of one and two site density matrices, energy, and bond entropy. Zeros in the errors are to be interpreted as numerical zeros. Bond entropies are not calculated for density matrices and excluded for this reason from the table of errors. Data for decay of the number of excitations. Quantum trajectories run with 500 different trajectories. Zeros in the errors are to be interpreted as numerical zeros. Bond entropies are not calculated for density matrices and excluded for this reason from the table of errors. Data for decay of the number of excitations. Quantum trajectories run with 500 different trajectories. Zeros in the errors are to be interpreted as numerical zeros.

Table II. Error Analysis for the Lindblad Master Equation. We simulate the Lindblad master equation for different time evolution methods and take as reference the implementation using the matrix exponential. In addition, the simulations with $H_{PJJ}$ are compared to the exponential decay of the number of excitations. Quantum trajectories run with 500 different trajectories. Zeros in the errors are to be interpreted as numerical zeros. Bond entropies are not calculated for density matrices and excluded for this reason from the table of errors. Data for additional methods can be found in the Appendix in Table V.

VII. THERMALIZATION OF THE LONG-RANGE QUANTUM ISING MODEL WITH THE FULL-SPECTRUM LINDBLAD MASTER EQUATION

The derivation of the Lindblad master equation [19, 43] for a given Hamiltonian is subtle in the many-body scenario. One possibility, dubbed the full spectrum Lindblad operators, is the inclusion of the energy eigenstates of the system [44, 45]. This approach elegantly handles the secular approximation but is obviously restricted to many-body systems with approximately six or fewer qubits, or the size of the Hilbert space equal to six qubits. The restriction to six qubits stems from the exponential taken in Liouville space for the Lindblad master equation, defined in Eq. (39). Krylov methods or the Trotter approximation are not suitable for this huge number of non-local Lindblad operators. We use the long-range quantum Ising model with the transverse field to study the thermalization timescales based on previous research presenting the complete analysis [46], which uses the full spectrum for the thermalization timescale of the quantum Ising model with regular nearest-neighbor interactions.
The usage of the full spectrum requires knowledge of the system Hamiltonian and the interaction Hamiltonian $H_I$; the Hamiltonian of the reservoir $H_R$ is assumed to be a set of harmonic oscillators and essential to derive the actual coupling constants. These parts of the model are defined in our example in the following as:

\[
H_{LRQI} = -\sum_{k=1}^{L-1} \sum_{k'=k+1}^{L} \frac{\cos(\phi)}{(k' - k)^\alpha} \sigma_k^x \sigma_{k'}^x - \sum_{k=1}^{L} \sin(\phi) \sigma_k^x ,
\]

(51)

\[
H_R = \sum_q \omega_q n_q ,
\]

(52)

\[
H_I = \sum_{k=1}^{L} \sum_q \sigma_k^x (g_{kq} b_q + g_{kq}^* b_q^*) .
\]

(53)

The Hamiltonian of the long-range quantum Ising model $H_{LRQI}$ depends on the number of sites $L$, and different long-range interactions are addressed via a specific choice of $\alpha$. To have similar energy scales across the complete phase diagram, the coupling of the $zz$-interactions and the external field in the $x$-direction are defined as a function of $\phi$; the range $\phi \in [0, \frac{\pi}{2}]$ describes the phase diagram with the ferromagnetic limit at $\phi = 0$, the critical point in the thermodynamic limit at $\phi = \frac{\pi}{4}$, and the paramagnetic limit is $\phi = \frac{\pi}{2}$. The conversion from the standard interaction $J$ and external field coupling $g$ is $\cos(\phi) = J/\sqrt{J^2 + g^2}$ and $\sin(\phi) = g/\sqrt{J^2 + g^2}$; thus, the quantities in $H_{LRQI}$ are unitless and time and temperature are in units of $1/\sqrt{J^2 + g^2}$ and $\sqrt{J^2 + g^2}$, respectively. The two limits are the recovery of the nearest-neighbor quantum Ising model for $\alpha \to \infty$ and the completely interacting model $\alpha = 0$. The reservoir Hamiltonian $H_R$ is a large set of harmonic oscillators indexed by $q$, where the summation of $q$ is not specified due to the large set of oscillators only required during the derivation of the final coupling. The interaction $H_I$ contains the creation or annihilation of a mode in the reservoir while flipping a spin in the system, see $\sigma_k^x$. The coupling $g_{kq}$ describes the strength of the interaction based on the site in the spin chain and the mode of the harmonic oscillator. This choice of $H_I$ has interactions of each spin with one reservoir, i.e., the common reservoir can induce collective long-range interaction in addition to the ones present in the system’s Hamiltonian. Moreover, the choice of the interaction Hamiltonian conserves the $Z_2$ symmetry of the quantum Ising model and we focus on the even symmetry sector in the following.

We skip details of the complete derivation of the structure of the resulting Lindblad equation; corresponding details can be found in the original study of the quantum Ising model [46] or the references [44, 45]. The Lindblad equation following the full-spectrum then reads:

\[
\dot{\rho} = -\frac{i}{\hbar} [H'_S, \rho] + \sum_{abcd} C_{abcd} \left( L_{ab} \rho L_{cd}^\dagger - \frac{1}{2} \{ L_{cd} L_{ab}, \rho \} \right) .
\]

(54)

The Lindblad operators $L_{ab} = |a\rangle \langle b|$ are transitions between eigenstates $|a\rangle$ and $|b\rangle$ of the system’s Hamiltonian, i.e., in our example $H_{LRQI}$. The coupling $C_{abcd}$ carries the dependence on the type of reservoir, the interaction strength between system and reservoir, the operator $\sigma_k^x$ acting in the interaction Hamiltonian $H_I$ on the system in terms of the overlaps $\langle a| \sigma_k^x |b\rangle$, and the temperature of the reservoir. The Hamiltonian $H'_S$ can have corrections with regards to the system Hamiltonian $H_S$. We choose the statistics of the reservoir following a 3D electromagnetic field [46]; we truncate the corrections in $H'_S$ which are related to the imaginary part of the principal value integral while resolving the integration of the electromagnetic field.

We are interested in how the thermalization timescale changes with the tuning of the long-range interactions $\alpha$; the unique steady state of the even symmetry sector corresponds to the thermal state. One can extract the slowest possible timescale from the eigenvalues $\Lambda_i$ of the Liouville operator $L$, defined in Eq. (45), constructed based on Eq. (54). The real part identifies the steady states of the open system whenever $\Re(\Lambda_i)$ equals zero. The first non-zero, and negative, $\Re(\Lambda_i)$ defines the slowest possible timescale in the system; this timescale is labeled with subscript 1 in the following. Depending on the initial state, this timescale may not be present. We observe for a temperature $T = 1.01$ in Fig. 9(a) and (b) that for small $\phi$ there is a stronger decoherence while decreasing $\alpha$, i.e., $|\Re(\Lambda_i)|$ is increasing when moving away from the limit of nearest-neighbor interactions. In contrast, the effects of decoherence are smaller for long-range interactions close to the paramagnetic limit, which is better visible in the log-representation of $\Re(\Lambda_1)$. The system’s thermalization timescale depends on $\phi$ in the nearest-neighbor model, with no thermalization in the paramagnetic limit due to the choice of $\sigma_k^x$ as the operator acting on the system in the interaction Hamiltonian $H_I$. Figure 9(c) and (d) shows that this behavior just described for a specific temperature $T$ depends as well on the temperature of the reservoir. Thus, experimental systems such as trapped ions with tunable $\alpha$ [47–49] can exhibit a variety of timescales across the $(\alpha, T)$ parameters space.

These results illustrate why the full spectrum is worth being included in a library as it allows for the study of thermalization via a reservoir. This type of simulation is one key example where ED is preferable over tensor network methods. The method requires a knowledge of all eigenstates. The structure of the Lindblad operators $L_{ab}$ combining two eigenstates $|a\rangle$ and $|b\rangle$ via an outer product to an operator leads to a bond dimension without truncation of $2^L$ in the case of qubits. Disregarding the huge amount of Lindblad operators of this kind for a possible system and the problems arising from it, the compression without
truncation can only begin after contracting a single Lindblad operator to the state with an intermediate bond dimension of $2^{23}$; the advantages of handling such a problem with vectors given by the ED methods and adding Lindblad operators directly to the Liouville operator are apparent.

VIII. COMPARISON TO QUTIP PACKAGE FOR BENCHMARKING

We compare single features of the ED package of OSMPS with comparable features of the QuTip library [50, 51] under version 4.0.2. The QuTip library provides a far more general framework to simulate closed and open quantum systems. We focus on the benchmark of the following tasks:

- Task (1): Partial trace over a pure state $|\psi\rangle$ retaining the reduced density matrix for the first site $\rho_1$. We point out that this case does not need a permutation in OSMPS. With regards to the implementation in QuTip, the many-body case does not seem to be the focus, and the density matrix of the complete pure state is built first and then traced out over $\rho$, which leads to some overhead.

- Task (2): Partial trace over a density matrix $\rho$ retaining the reduced density matrix for the first site $\rho_1$. Since both libraries start from the density matrix, this comparison is more valuable than tracing over pure states in (1).
Table III. Benchmarking QuTip and OSMPS. The comparison of run times includes (1) the partial trace of a pure state and (2) a density matrix, both tracing over the sites 2 to \( L \) where \( L \) is the system size. Furthermore, (3) contains the time evolution for the quantum Ising model for a time-independent Hamiltonian. For OSMPS we distinguish between the 4th Trotter decomposition, i.e., (3a), and the Krylov method labeled as (3b). QuTip uses the default ODE solver. All CPU times \( T_{CPU}(L) \) are given in seconds as a function of the system size \( L \). The times are calculated cumulatively for 100 iterations of the corresponding task.

| Task | \( T_{CPU}(4) \) | \( T_{CPU}(6) \) | \( T_{CPU}(8) \) | \( T_{CPU}(10) \) | \( T_{CPU}(12) \) | \( T_{CPU}(14) \) | \( T_{CPU}(26) \) |
|------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| (1), QuTip | 0.15            | 0.16            | 0.20            | 0.42            | 2.76            | −               | −               |
| (1), OSMPS | 0.01            | 0.01            | 0.01            | 0.01            | 0.01            | 0.01            | 22.00           |
| (2), QuTip | 0.12            | 0.14            | 0.18            | 0.39            | 2.71            | −               | −               |
| (2), OSMPS | 0.01            | 0.01            | 0.02            | 0.16            | 2.61            | −               | −               |
| (3), QuTip | 3.19            | 4.50            | 7.16            | 17.00           | 57.05           | 248.60          | −               |
| (3a), OSMPS | 9.72            | 14.71           | 20.86           | 33.20           | 72.54           | 232.75          | −               |
| (3b), OSMPS | 16.84           | 20.20           | 25.60           | 31.69           | 314.91          | 973.38          | −               |

- Task (3): Time evolution of the quantum Ising model with a time-independent Hamiltonian. The QuTip library uses an ODE solver to evolve a given initial state in time, which we compare to the Krylov and fourth-order Trotter method from OSMPS. For QuTip we set up a single measurement of the identity matrix. OSMPS takes the default measurements including the measurement of the energy (Hamiltonian) and the overlap to the initial state, where the first one contains the major part of the measurement time in OSMPS. For OSMPS we consider the two approximate methods with the fourth order Trotter (3a) and the Krylov (3b). We do not use the \( \mathbb{Z}_2 \) symmetry of the system here.

We run the benchmark simulations on an Intel(R) Core(TM) i7-3610QM CPU @ 2.30GHz, 8 GB RAM and present the CPU times measured with Python’s \texttt{time.clock} package in Table III. The run times of the trace over a density matrix are comparable and show that our implementation scales as well as QuTip. The time evolutions are more difficult to compare with regards to the order of the method etc., but the time of the Trotter decomposition for large systems shows that we can offer a valid alternative. The Krylov evolution, as expected from the previous scaling analysis, e.g., in Fig. 2, takes longer.

IX. CONCLUSIONS

In this paper, we have shown the approaches for exact diagonalization methods including the use of symmetries and other evolution techniques without truncating entanglement, including the Trotter decomposition and the Krylov subspace method. The methods can be extended from closed to open systems.

We have described efficient methods for time evolutions and measurements without symmetries when the Hilbert space is built from a tensor product of local Hilbert spaces. Therefore, the state vector representing the wave function can be rewritten as a rank-\( L \) tensor. The methods include especially approaches for the Trotter time evolution, matrix-vector multiplication necessary for the Krylov time evolution, and measurements via reduced density matrices. For symmetry-adapted methods, the key is the mapping into the basis. However, methods for Trotter decomposition or matrix-vector multiplication cannot avoid looping over entries and cannot use faster matrix algebra. Reduced density matrices, in particular, suffer from this limitation and we have presented a feasible approach here, scaling with \( O(DSL) + O(d^2LD_S) \).

Open system methods according to the Lindblad master equation follow two known approaches via the quantum trajectories and the Liouville space mapping the density matrix to a vector. The evolution of the vector representing the density matrix follows a Schrödinger-like equation. While quantum trajectories can use system sizes and methods equivalent to pure states sampling over many realizations, the methods for Liouville space have to be adapted slightly for the use with density matrices. This adaption refers to the matrix exponential or its approximation. While the Hamiltonian was hermitian, the Liouville operator is not necessarily hermitian. For any propagation of the whole density matrix, we can use the rule of thumb that about half the number of sites of the pure system can be simulated. The disadvantage of quantum trajectories is the error, which was of the order \( 10^{-2} \) for the total number of excitations in the photon Josephson junctions for 500 trajectories. We point out that the number of trajectories needed might differ for various Hamiltonians. The application of the full-spectrum Lindblad operators to the long-range quantum Ising model shows that the thermalization timescale exhibits a rich landscape as a function of the strength of the long-range interaction, the temperature of the reservoir, and the position in the phase diagram.

The exact diagonalization methods presented help to develop new methods and can serve as a convergence test and benchmark. One application is the exploration of highly entangled states, which cannot be simulated with tensor network methods with the examples specified in the following. The full spectrum Lindblad equation presented in Sec. VII is too inefficient to be used in tensor network methods due to the vast amount of Lindblad operators to be stored while ED methods can add them directly to the Liouville operator stored as a matrix. Methods such as the Krylov time evolution work for both, but tensor networks need to
variational fit states in the case of long-range interaction or add MPSs and compress them; in contrast, ED can sum the vectors directly without overhead. Although tensor networks can tackle in theory equal system sizes in comparison to ED when not truncating entanglement, ED can be a more convenient approach and with less parameters to tune. Other Hamiltonians exploring only a small part of the Hilbert space might be scalable in size to more than the 25 qubits of the Ising model, where we show the scaling of Trotter decomposition up to 27 qubits. Models like the Bose-Hubbard model in \( k \)-space have long-range interactions and a large bond dimension in the Matrix Product Operators of the Hamiltonian when using tensor network methods such as MPS, and thus find in exact diagonalization methods an attractive alternative. The rule sets allow for the flexible construction of the Hamiltonians without the restriction of the MPO representation with tensor network methods, e.g., any infinite function does not have to be fitted to exponential rules. This argument might apply as well to systems with randomized interactions such as spin glasses [52, 53].

Finally, open quantum systems are another attractive field for exact diagonalization libraries despite the limitations to fewer sites due to the Liouville space. Any formulation of a master equation for \( \rho \) could be explored with the easy to program tools within our exact diagonalization codes, e.g., if Lindblad operators are non-local and/or long-range including a coupling through the environment. Especially if the full spectrum of the Hamiltonian is necessary for the Lindblad master equation, exact diagonalization is an ideal test case for any approximation truncating the number of states in the spectrum.

Therefore, these methods are not only useful as problems set in classes, but are worth being improved and optimized for exploring new frontiers in research.

**ACKNOWLEDGMENTS**

We thank I. de Vega for many discussions and insights on the full spectrum approach to the Lindblad master equation. We gratefully appreciate contributions from and discussions with L. Hillberry, S. Montangero, H. North, G. Shchedrin, and M. L. Wall. The calculations were carried out using the high performance computing resources provided by the Golden Energy Computing Organization at the Colorado School of Mines. This work has been supported by the NSF under the grants PHY-1520915, and OAC-1740130, and the AFOSR under grant FA9550-14-1-0287. This work was performed in part at the Aspen Center for Physics, which is supported by the US National Science Foundation grant PHY-1607611. We acknowledge support of the U.K. Engineering and Physical Sciences Research Council (EPSRC) through the “Quantum Science with Ultracold Molecules” Programme (Grant No. EP/P01058X/1).

[1] Steven R. White, “Density matrix formulation for quantum renormalization groups,” Phys. Rev. Lett. 69, 2863–2866 (1992).
[2] Anders W. Sandvik and Juhani Kurkijärvi, “Quantum Monte Carlo simulation method for spin systems,” Phys. Rev. B 43, 5950–5961 (1991).
[3] N. V. Prokoféev, B. V. Svistunov, and I. S. Tupitsyn, “Exact, complete, and universal continuous-time worldline Monte Carlo approach to the statistics of discrete quantum systems,” Journal of Experimental and Theoretical Physics 87, 310–321 (1998).
[4] Antoine Georges, Gabriel Kotliar, Werner Krauth, and Marcelo J. Rozenberg, “Dynamical mean-field theory of strongly correlated fermion systems and the limit of infinite dimensions,” Rev. Mod. Phys. 68, 13–125 (1996).
[5] Anatoli Polkovnikov, “Phase space representation of quantum dynamics,” Annals of Physics 325, 1790–1852 (2010).
[6] J. Schachenmayer, A. Pikovsky, and A. M. Rey, “Many-Body Quantum Spin Dynamics with Monte Carlo Trajectories on a Discrete Phase Space,” Phys. Rev. X 5, 011022 (2015).
[7] Guifré Vidal, “Efficient classical simulation of slightly entangled quantum computations,” Phys. Rev. Lett. 91, 147902 (2003).
[8] Ulrich Schollwöck, “The density-matrix renormalization group in the age of matrix product states,” Annals of Physics 326, 96 – 192 (2011), January 2011 Special Issue.
[9] Guifré Vidal, “Class of Quantum Many-Body States That Can Be Efficiently Simulated,” Phys. Rev. Lett. 101, 110501 (2008).
[10] Y.-Y. Shi, L.-M. Duan, and G. Vidal, “Classical simulation of quantum many-body systems with a tree tensor network,” Phys. Rev. A 74, 022320 (2006).
[11] Verstraete F., Murg V., and Cirac J.I., “Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems,” Advances in Physics 57, 143–224 (2008).
[12] Thomas Häner and Damian S. Steiger, “0.5 petabyte simulation of a 45-qubit quantum circuit,” in Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC ’17 (ACM, New York, NY, USA, 2017) pp. 33:1–33:10.
[13] E. Pednault, J. A. Gunnels, G. Nannicini, L. Horesh, T. Magerlein, E. Solomonik, and R. Wisnieff, “Breaking the 49-Qubit Barrier in the Simulation of Quantum Circuits,” ArXiv e-prints 1710.05867 (2017).
[14] Zhao-Yun Chen, Qi Zhou, Cheng Xue, Xia Yang, Guang-Can Guo, and Guo-Ping Guo, “64-qubit quantum circuit simulation,” Science Bulletin (2018), 10.1016/j.scib.2018.06.007.
[15] Logan E. Hillberry, *Entanglement and complexity in quantum elementary cellular automata*, Master’s thesis, Colorado School of Mines (2016), http://inside.mines.edu/~lcarr/theses/hillberry_thesis_2016.pdf.
[16] J. Watrous, “On one-dimensional quantum cellular automata,” in *Proceedings of IEEE 36th Annual Foundations of Computer Science* (1995) pp. 528–537.
Appendix A: Rule sets for exact diagonalization

As OSMPS is a many-body library, Hamiltonians are not built by the user, but through rule sets. The following rule sets are currently enabled in our open source diagonalization libraries:

\[
\text{Site rule} : \sum_{k=1}^{L} w \cdot c_k \cdot O_k
\]

\[
\text{Bond rule} : \sum_{k=1}^{L-1} w \cdot c_k \cdot O_k \otimes O'_{k+1}
\]

\[
\text{Exp rule} : \sum_{k<j}^{L} \frac{w \cdot c_k}{d_{p(j-k-1)}} \cdot O_k \otimes \left( \sum_{q=k+1}^{j-1} Q_q \right) \otimes O'_{j}
\]

\[
\text{InfiniteFunc} : \sum_{k<j}^{L} w \cdot c_k \cdot f(|j-k|) O_k \otimes \left( \sum_{q=k+1}^{j-1} Q_q \right) \otimes O'_{j}
\]

\[
\text{MBString} : \sum_{k=1}^{L-j} \sum_{j=1}^{W_{MB}} w \cdot c_k \otimes O_{j}^{k+j-1}
\]

where \( L \) is the number of sites in the one-dimensional chain, \( w \) is a general coupling, and \( c_k \) is a possibly space-dependent coupling. \( O_k, O_k', \) and \( Q_k \) are operators acting on the local Hilbert space. \( Q_k \) targets fermionic systems, which have a phase operator due to the Jordan-Wigner transformation [37, 54] mapping fermions onto the lattice. The Jordan-Wigner transformation is a mapping between spins and fermions where the transformation ensures that the corresponding commutation relations hold. The exponential rule includes a decay parameter \( d_p \leq 1 \). The function \( f(\cdot) \) describes the coupling of the two operators at an arbitrary distance for the InfiniteFunc rule. The Many-Body String (MBString) term has a width of \( W_{MB} \) sites and uses the tensor product \( \otimes \) to indicate the construction on the composite Hilbert space. Synonyms for the tensor product are "outer product" or "Kronecker product".

Appendix B: Scaling of memory resources

We have discussed in Fig. 2 the scaling of the computation time for the different time evolution methods. In this appendix, we discuss the corresponding scaling of the memory resources, shown in Fig. 10. We use the python package \texttt{resource} to track the maximal memory usage over the complete calculation.

The matrix exponential has the highest demands on memory as the complete Hamiltonian has to be stored, see Fig. 10(a). The two different set of curves distinguish the matrix exponential with and without \( \mathbb{Z}_2 \) symmetry. The symmetry has a smaller Hilbert space by a factor of 2 and thus lower memory requirements. The difference between dense and sparse methods is minimal within the quantum Ising model.

In contrast, both the Krylov and the Trotter method show higher memory requirements for the symmetry-conserving. We explain this trend with the fact that the mapping for the symmetry must be stored, overcoming the trend that the state vector itself is smaller for systems with \( \mathbb{Z}_2 \) symmetry. The lower memory for the Trotter without symmetry is due to the Krylov methods storing a set of the Krylov vectors. Figure 10(b) shows the same trend for the Bose-Hubbard model, where we recall that the \( U(1) \) symmetry takes a much smaller fraction of the Hilbert space in comparison to the quantum Ising model with and without symmetry.

We have the memory-optimized algorithms for the Krylov algorithm in our library replacing the scipy implementation. Therefore, we take a look at the memory consumption of the different Krylov modes; Fig. 10(c) considers the quantum Ising model without using the \( \mathbb{Z}_2 \) symmetry. For example, the memory requirements for 24 qubits with mode 3 are below the memory used for 20 qubits in the other modes. This increased number of qubits is a clear improvement. The Bose-Hubbard double-well problem, see Fig. 10(d), is the only simulation pushing the limits past laptop and desktop computers. Peaking at around 91GB, such simulations are limited to HPC environments. This high memory demand is related to the measurements. Therefore, we concentrate on simulations without the measurement of the site and bond entropy to distinguish the different Krylov modes. We conclude that the third mode a priori saves memory for systems without symmetry.

[55] Gavin K. Brennen and Jamie E. Williams, “Entanglement dynamics in one-dimensional quantum cellular automata,” Phys. Rev. A 68, 042311 (2003).
Figure 10. Scaling of the Memory for Time Evolution Methods. We profile five time steps for different system sizes\( L \) comparing the matrix exponential (ME) with sparse matrices (sp) and with dense matrices to the Trotter decomposition and to the default Krylov method. The maximal memory of the process in kilobytes is plotted on a lin-log-scale. (a) The nearest neighbor quantum Ising model has a\( Z_2 \) symmetry, which we use in comparison to the methods without symmetry. (b) The Bose-Hubbard model is considered with a local dimension of\( d = 5 \), and the\( U(1) \) symmetry is used with unit filling in the simulations marked with\( U/1 \). (c) The scaling of the RAM for the different Krylov modes in the quantum Ising model without using its\( Z_2 \) symmetry. The memory-optimized mode 3 clearly saves resources for large systems. (d) The memory requirements for the Bose-Hubbard model in a double-well at filling\( N = (L - 1)/2 \) with\( N \) odd are dominated by the measurement of the site and bond entropy. Without measurements, the Krylov modes show different RAM requirements.

### Appendix C: Quantum Cellular Automata and Quantum Gates

The algorithms described so far evolve the quantum state or density matrix according to the differential equation presented in Eqs. (1), (39), or (41). The methods using the full matrix exponential and the Trotter decomposition calculate the propagator, which we apply to the quantum state. But some problems in quantum physics are directly formulated in terms of propagators, i.e., quantum cellular automata [15, 16] and quantum gates. Quantum gates are especially common in quantum information theory: most researchers know the quantum Fourier transformation defined by their gates [38], but not the corresponding Hamiltonian.

As the major steps are already implemented with the application of the propagator to a state, we also provide the tools to formulate dynamics solely in terms of quantum gates defined on quasi-local gates. The time evolution is then formulated as one quantum circuit consisting of multiple sub-circuits. We measure after each sub-circuit, which can be different. Each sub-circuit is a series of gates from\( G_1 \) to\( G_K \):

\[
|\psi'\rangle = G_K G_{K-1} \cdots G_2 G_1 |\psi\rangle,
\]

where\( |\psi'\rangle \) is the wave-function after the sub-circuit. We set up a QECA rule, i.e., the SWP-rule 6 for\( L = 21 \) sites [15]. The initial state is a single spin down in the middle of the system. Figure 11 describes the evolution according to the rule. We waive a detailed study of the scaling for the quantum circuits or gates. On the one hand, such an evolution without symmetries can be easily estimated by the contraction of the gates to the state. On the other hand, even gates obeying a symmetry and acting on two sites have an upper bound of the Trotter time evolution up to some factor for the number of applications.

In conclusion, this evolution tool comes at almost no additional cost for the implementation. Although we have at present only gates in the OSMPS library, it can be generalized to Kraus operators [38] as the next step, and QECA evolutions with Kraus operators have already been suggested in [55].
Figure 11. Quantum Cellular Automata example for Quantum Circuits. Asymmetric SWP Quantum Cellular Automata rule for $L = 21$ sites is one example for the application of quantum circuits.

Figure 12. Bose-Hubbard Double-Well Problem. Two bosons occupy five sites. The first two sites represent the left well, the center site represents the barrier, and the last two sites are the right well. (a) The initial state at time $t = 0$ is the ground state of the system with a potential on the middle site (3rd site, dark blue) and the right well (4th and 5th site, light blue). (b) For times $t > 0$, we switch off the potential on the right well and the bosons oscillate between the left and right well.

Appendix D: Best-Practice Example

In this appendix, we concentrate on the setup of an actual simulation for the exact diagonalization library within OSMPS. We choose the double-well problem using the Bose-Hubbard model with the Hamiltonian introduced in Eq. (18) and compare the behavior of the closed system versus the open system. Figure 12 represents a sketch of the problem. The first step of any Python program is importing the necessary packages. We need the Python packages for MPS, the EDLib, numpy, a function for copying classes, the interface to the system parameters, and the plotting library. We show the corresponding code snippet always in the following Listing. Listing 1 shows the importing of the packages.

Listing 1. Import the packages.

```python
import MPSPyLib as mps
import MPSPyLib.EDLib as ed
import numpy as np
from copy import deepcopy
import sys
import matplotlib.pyplot as plt
```

The next parts of the code are in the main function. First, we generate the operators for the Bose-Hubbard model. In addition to the set of bosonic operators, OSMPS can provide fermionic operators and spin operators. We choose a maximum of four bosons per site leading to a local dimension of $d = 5$. Furthermore, we have to construct the interaction operator $n(n - 1)$. Matrices within `Operators` are accessed via string keys following the dictionary structure of Python. Similar new operators are defined as a new entry in the dictionary.
Listing 2. Generate the operators for the Bose-Hubbard model.

```python
# Generate bosonic operators
Operators = mps.BuildBoseOperators(4)
Operators['interaction'] = 0.5 * (np.dot(Operators['nbtotal'],
             Operators['nbtotal'])
             - Operators['nbtotal'])
```

In the next step, we build the Hamiltonian consisting of the tunneling term $b_k \dagger b_{k+1} + h.c.$, the interaction $n(n - I)$, and a site-dependent potential to model the double-well and its action through the number operator $n$. The argument `hparam` is the coupling for each term in the Hamiltonian defined as a string. The actual value is specified later on for each simulation allowing for flexibility, which is then shown in Listing 6. We define all values, which are constant for all simulations directly as variables. We make a copy of the MPO for the open system and add the Lindblad term.

Listing 3. Create the Hamiltonian via rule sets.

```python
# Define Hamiltonian MPO for closed systems (Hc) and open system (Ho)
Hc = mps.MPO(Operators)
Hc.AddMPOTerm('bond', ['bdagger', 'b'], hparam='J', weight=-1.0)
Hc.AddMPOTerm('site', 'interaction', hparam='U', weight=1.0)
Hc.AddMPOTerm('site', 'nbtotal', hparam='Vx', weight=1.0)
Ho = deepcopy(Hc)
Ho.AddMPOTerm('lind1', 'nbtotal', hparam='gamma', weight=1.0)
L = 5
J = 1.0
U = 1.0
fill = 2
Vx0 = np.array([0.0, 0.0, 1.0, 1.0, 1.0])
```

As a next step, the measurements are defined. We restrict ourselves to the local number operator, which allows us to show the oscillations between the two wells. The arguments in `AddObservable` specify the type of measurement, if applicable, the operators used, and a string identifier to access the results in a dictionary later on.

Listing 4. Define the observables.

```python
# Default measures
Obs = mps.Observables(Operators)
Obs.AddObservable('site', 'nbtotal', '<n>)
```

We define the dynamics as objects of the `QuenchList` class, one for the closed and one for the open system. We start with the potential with the barrier in the middle of the system and define a function, which returns the time-independent potential. For the time evolution, we use the Krylov method. The time-dependency `timedep` must be set by hand in this case.

Listing 5. Define the dynamics via the QuenchList class.

```python
# Define the dynamics
Vt = np.array([0.0, 0.0, 1.0, 0.0, 0.0])
def quenchfunc(t, Vt=Vt):
    return Vt
T = 100.0
dt = 0.01
tConv = mps.KrylovConvParam()
CQuenches = mps.QuenchList(Hc)
CQuenches.AddQuench(['Vx'], T, dt, [quenchfunc],
                        ConvergenceParameters=tConv,
                        stepsforoutput=10, timedep=False)
OQuenches = mps.QuenchList(Ho)
OQuenches.AddQuench(['Vx'], T, dt, [quenchfunc],
```
ConvergenceParameters=tConv,
stepsforoutput=10, timedep=False)

The simulations to be carried out are specified in a list of dictionaries. We start with the dictionary for the closed system. The necessary keys can be found in the documentation, but we point out that we now specify the hparam used in the definition of the rule sets for the Hamiltonian.

Listing 6. Add the closed system to the list of simulations.

```python
# Define the closed system simulation as a dictionary inside a list
params = [{
    'simtype' : 'Finite',
    'job_ID' : 'DoubleWell',
    'unique_ID' : '_SE',
    'Write_Directory' : 'TMP/',
    'Output_Directory' : 'OUTPUTS/',
    # System settings
    'L' : L,
    'J' : J,
    'U' : U,
    'Vx' : Vx0,
    'Abelian_generators' : ['nbtotal'],
    'Abelian_quantum_numbers' : [fill],
    'Quenches' : CQuenches,
    'DynamicsObservables' : Obs,
    'MPSObservables' : Obs
}]
```

Since we do not iterate over different system sizes, we can construct the basis states for the $U(1)$ symmetry once prior to running the simulations. Although EDLib would build the symmetry sector for each simulation automatically, it would use a general method, which is not necessary in this case. The class containing the basis is constructed via the following Listing 7.

Listing 7. Construct the basis states for a single $U(1)$ symmetry.

```python
# Construct symmetry sector
SymmSec = ed.SingleU1Symmetry(Operators, params[0])
params[0]['SymmSec'] = SymmSec
```

In the next step, we iterate two different coupling for the Lindblad operators. We can append the additional dictionaries to the list.

Listing 8. Add open system simulations with two different couplings to the list of simulations.

```python
# Add two different coupling strengths for the open system
for gamma in [0.025, 0.05]:
    params.append({
        'simtype' : 'Finite',
        'job_ID' : 'DoubleWell',
        'unique_ID' : '_ME_gamma%3.6f'%(gamma),
        'Write_Directory' : 'TMP/',
        'Output_Directory' : 'OUTPUTS/',
        # System settings
        'L' : L,
        'J' : J,
        'U' : U,
        'Vx' : Vx0,
        'gamma' : gamma,
        'Abelian_generators' : ['nbtotal'],
        'Abelian_quantum_numbers' : [fill],
        'Quenches' : CQuenches,
        'DynamicsObservables' : Obs,
        'StaticObservables' : Obs,
        'SymmSec' : SymmSec
    })
```
Since we use different rule sets for the MPO in the simulations, we need a list of MPOs with the same length as the simulation list. Afterwards, we can execute the simulations.

Listing 9. Build a list of Hamiltonians and run simulations.

```python
# Different MPO for simulations --> list of MPOs as used in params
Hlist = [Hc, Ho, Ho]

if (not PostProcess):
    MainFile = mps.WriteFiles(params, Operators, Hlist, PostProcess)
    ed.runED(params, Operators, Hlist)
    return
```

The post-processing imports the results as dictionaries. For the statics, the list contains the dictionaries directly, for dynamics the first list iterates over the different simulations and the inner list iterates over the dictionaries for each time step.

Listing 10. Reading the results and plotting them.

```python
# PostProcess
# -----------

MainFile = mps.WriteFiles(params, Operators, Hlist, PostProcess)
StatOut = mps.ReadStaticObservables(params)
DynaOut = mps.ReadDynamicObservables(params)
fig = plt.figure()
plt.rc('text', usetex=True)
plt.rc('font', family='serif', size=22)
ax1 = fig.add_subplot(121)

# Plot populations ground state (equal for all systems)
ax1.plot(range(1, L + 1), StatOut[0]['<n>'], 'bo:')
ax1.set_xlabel(r'Site index $i$')
ax1.set_ylabel(r'Population $\langle n_i \rangle$')
plt.xticks(list(range(1, L + 1)))

# Plot dynamics population left well
ax2 = fig.add_subplot(122)
label = [r'$\gamma = 0.0$', r'$\gamma = 0.025$', r'$\gamma = 0.05$']
ii = -1

for DynaOutii in DynaOut:
    ii += 1
    tlist = np.linspace(DynaOutii[0]['time'], DynaOutii[-1]['time'],
                        len(DynaOutii))
    Nleft = np.zeros((len(DynaOutii)))

    # jj is running over the different time steps
    for jj in range(len(DynaOutii)):
        Nleft[jj] = np.sum(DynaOutii[jj]['<n>'][2])
    ax2.plot(tlist, Nleft, label=label[ii])

ax2.set_xlabel(r'Time $t$')
ax2.set_ylabel(r'Population left well $N_{\text{left}}$')
ax2.legend(loc='upper right')
plt.show()
return
```

The plot obtained from this simulation is presented in Fig. 13 and shows that the coupling considered in the Lindblad master equation leads to a damping of the oscillations between the left and right well. The last lines of the code call the main function in case the script is called, which allows us in addition to import the function from another module.
Figure 13. Closed and Open System Double-Well Problem. (a) The initial state is characterized by the population per site on the left. We obtain it via an eigendecomposition of the Hamiltonian, where only the left well has a zero chemical potential. (b) The oscillations between the two wells damp faster for larger coupling $\gamma$ to the reservoir of the Lindblad master equation. The limit $\gamma = 0$ is the closed system. The time $t$ is in units of the tunneling $J$.

Listing 11. Calling the main function.

```python
if(__name__ == '__main__ '):
    PostProcess = False
    for keyval in sys.argv[1:]:
        key, val = keyval.split('=')
    if(key == '--PostProcess'): PostProcess = (val == 'True') or (val == 'T')
    main(PostProcess)
```

Appendix E: Details of Convergence Studies

In this appendix, we extend the convergence studies of the main body of this article, i.e., Sec. V. We showed the dependence of the error on the method and the time step in Fig. 7 for a time-dependent Hamiltonian in the Bose-Hubbard model. Figure 14 shows the same kind of study for the time-independent Hamiltonian corresponding to a sudden quench within the Mott insulator as described in the second scenario in Sec. V. The error from the second order time ordering vanishes and only the Trotter decompositions remain with an error depending on the time step $dt$. In addition, we present the first scenario with a time-dependent Ising Hamiltonian in Fig. 15. We calculate again the rate of convergence according to Eq. (38) and the maximal distance between the single site density matrices. Those rates are $2.07$, $2.0$ for the second order Trotter decomposition, $2.93$, $1.97$ for the fourth order Trotter decomposition, $2.08$, $2.0$ for the Krylov in mode 1, and the matrix exponential has rates of convergence of $2.08$, $2.02$. The first value corresponds to the pair $(dt = 0.01, dt' = 0.1)$ and the second value $(dt = 0.001, dt' = 0.01)$. We confirm that the convergence rate of $dt^2$ is reproduced in the Ising model due to the error in time-ordering.

For the open system analysis, we studied the error of the evolution of the von Neumann equation for the density matrix in a closed system and the Lindblad master equation for selected time evolution schemes in Tables I and II. Tables IV and V show the errors of different settings. We conclude that the $4^{th}$ order Trotter evolution reduces the error to almost machine precision. The built-in scipy Krylov method delivers a better slightly better result than the Krylov methods from the EDLib of OSMPS,
Figure 14. Convergence of Methods for the time-independent Bose-Hubbard Model. We compare the 2nd and 4th order Trotter time evolution and the Krylov method with mode 1 against the reference (ref) taken as the matrix exponential (ME) with $dt = 0.0001$ for various measures. We consider the Bose–Hubbard model in a linear quench from on-site interaction strength $U(t = 0) = 10.0$ to $U(t = 0.5) = 8.0$ for a system size of $L = 6$ and unit filling. The reference value is the matrix exponential for the smallest time step $dt = 0.0001$. The minimal error is set to $10^{-18}$ indicating machine precision. The measures are (a) the minimal distance over all single-site reduced density matrices, (b) the minimal distance over all two-site density matrices, (c) the error in energy, and (d) the error in the bond entropy for the splitting in the middle of the system.

e.g., the error in energy is about two orders (one order) of magnitude smaller for the photon Josephson junctions in the von Neumann (Lindblad master) equation, defined in Eq. (47). In Table V we point out that the errors for the quantum trajectories do not depend on the method. We explain this by the fact that the seed of the pseudo-random number generator is fed with the ID of the trajectory and therefore the trajectories for each method are equal. Also, the error from the quantum trajectory approach is bigger than the error from the time evolution method and, therefore, the latter error is not visible in the table.

Appendix F: Add-ons for Open Quantum Systems

In the following, we derive the transformation into the Liouville space given via the equation and already introduced in Eq. (44)

$$O' \rho O \rightarrow O' \otimes O^T |\rho\rangle\langle \rho|.$$  (F1)
Figure 15. Convergence of Methods for the Quantum Ising Model. We compare the 2nd and 4th order Trotter time evolution and the Krylov method in mode 1 and the matrix exponential (ME). The reference (ref) is the matrix exponential with $dt = 0.001$ for all four measures. We consider the quantum Ising model using its $Z_2$ symmetry in a linear quench from external field $h(t = 0) = 5.0$ to $h(t = 0.5) = 4.5$ for a system size of $L = 10$. The measures are (a) the minimal distance over all single-site reduced density matrices, (b) the minimal distance over all two-site density matrices, (c) the error in energy, and (d) the error in the bond entropy for the splitting in the middle of the system.

which we used to build the Liouville operator $L$ for the time evolution of an open system. First, we rewrite the equation in terms of indices$^2$ where the elements can be permuted:

$$O' \rho O = O'_{ij} \rho_{jk} O_{kl} = O'_{ij} O_{kl} \rho_{jk} = O'_{ij} O^T_{lk} \rho_{jk}. \quad (F2)$$

$\rho$ can be transformed into a vector with a combined index $(jk)$. From the two matrices the combination of indices can be achieved using the tensor product:

$$O' \rho O = O'_{ij} O^T_{lk} |\rho\rangle_{(jk)} = (O' \otimes O^T)_{(il), (jk)} |\rho\rangle_{(jk)}. \quad (F3)$$

To have the index $k$ of $O$ in the columns of the result of the tensor product, we had transposed $O$ before. We point out that this definition works independently of row- or column-major order of the memory.

$^2$ The indices are not related to the same symbols introduced previously.
Table IV. Additional Error Analysis for the Density Matrix in the von Neumann Equation. We show additional methods extending Table I for simulations of the closed system with the open system code of the library and the reference result is the closed system solving the Schrödinger equation.

| Model | Error | Liou (Trotter-4) | Liou (Krylov-0) | Liou (Krylov-2) | Liou (Krylov-3) | QT1 (Trotter-4) | QT1 (Krylov-0) |
|-------|-------|-----------------|----------------|----------------|----------------|----------------|---------------|
| $H_{PI}$ | $\epsilon_{\text{local}}$ | 0.0 | 0.0 | 2.71 $\cdot 10^{-13}$ | 2.71 $\cdot 10^{-13}$ | 0.0 | 0.0 |
|       | $\epsilon_{\text{corr}}$ | 0.0 | 0.0 | 7.34 $\cdot 10^{-13}$ | 7.34 $\cdot 10^{-13}$ | 0.0 | 0.0 |
|       | $\epsilon_{E}$ | 2.71 $\cdot 10^{-13}$ | 7.34 $\cdot 10^{-13}$ | 5.52 $\cdot 10^{-13}$ | 5.52 $\cdot 10^{-13}$ | 1.16 $\cdot 10^{-13}$ | 4.0 $\cdot 10^{-15}$ |
|       | $\epsilon_{S}$ | - | - | - | - | 2.45 $\cdot 10^{-12}$ | 8.88 $\cdot 10^{-16}$ |
| $H_{DW}$ | $\epsilon_{\text{local}}$ | 6.34 $\cdot 10^{-13}$ | 8.87 $\cdot 10^{-14}$ | 8.56 $\cdot 10^{-14}$ | 8.54 $\cdot 10^{-14}$ | 9.0 $\cdot 10^{-14}$ | 8.99 $\cdot 10^{-14}$ |
|       | $\epsilon_{\text{corr}}$ | 0.0 | 2.21 $\cdot 10^{-13}$ | 7.95 $\cdot 10^{-14}$ | 1.11 $\cdot 10^{-13}$ | 9.46 $\cdot 10^{-14}$ | 3.91 $\cdot 10^{-13}$ |
|       | $\epsilon_{E}$ | 3.53 $\cdot 10^{-11}$ | 7.81 $\cdot 10^{-13}$ | 8.04 $\cdot 10^{-13}$ | 8.13 $\cdot 10^{-13}$ | 3.48 $\cdot 10^{-11}$ | 7.74 $\cdot 10^{-13}$ |
|       | $\epsilon_{S}$ | - | - | - | - | 2.12 $\cdot 10^{-10}$ | 2.38 $\cdot 10^{-14}$ |

Table V. Additional Error Analysis for the Density Matrix in the Lindblad Master Equation. We show additional methods extending Table II for simulations of the open quantum system and the reference result is the open system using the matrix exponential.