




Fast time evolution of matrix product states using the QR decomposition

Jakob Unfried ^{*}, Johannes Hauschild , and Frank Pollmann 

Department of Physics, TFK, Technische Universität München, James-Frank-Straße 1, D-85748 Garching, Germany and Munich Center for Quantum Science and Technology (MCQST), Schellingstraße 4, D-80799 München, Germany



(Received 6 February 2023; revised 4 April 2023; accepted 5 April 2023; published 21 April 2023)

We propose and benchmark a modified time-evolving block decimation algorithm that uses a truncation scheme based on the QR decomposition instead of the singular value decomposition (SVD). The modification reduces the scaling with the dimension of the physical Hilbert space d from d^3 down to d^2 . Moreover, the QR decomposition has a lower computational complexity than the SVD and allows for highly efficient implementations on GPU hardware. In a benchmark simulation of a global quench in a quantum clock model, we observe a speedup of up to three orders of magnitude comparing QR and SVD based updates on an A100 GPU.

DOI: [10.1103/PhysRevB.107.155133](https://doi.org/10.1103/PhysRevB.107.155133)

I. INTRODUCTION

Numerical simulation of the dynamics of quantum many-body systems in and out of equilibrium is essential for the understanding of a wide range of physical phenomena. Following the success of the density matrix renormalization group (DMRG) method [1,2] for efficiently finding ground states of one-dimensional (1D) quantum systems in terms of matrix product states (MPSs), several related techniques have been developed to efficiently simulate the time evolution [3–8], with applications to classical simulation of generic quantum circuits and in particular of quantum computing [9,10]. These methods have since allowed access to experimentally relevant observables, such as dynamical correlation functions which can be compared with data from neutron scattering and ultracold atomic gases [11–13], and far out of equilibrium dynamics [14], providing profound insights into long-standing questions about quantum thermalization [15], many-body localization [16–19], and transport properties [20–25].

In a series of recent works [26–30], it has been demonstrated that accelerated linear algebra operations on graphics processing units (GPUs) and tensor processing units (TPUs) allow various numerical tasks, and in particular simulation of quantum dynamics, to be carried out not only significantly faster but also more power efficiently. However, many MPS-based algorithms heavily rely on singular value decompositions (SVDs), which are slow in the GPU implementations known to us. For example, the prominent time-evolving block decimation (TEBD) [5,31] algorithm performs an SVD following every application of a two-site gate, in order to truncate the bond dimension. In this work, we propose a modification to the TEBD algorithm for MPS time evolution, which uses QR decompositions to achieve a variational truncation, replacing the SVD. This truncation scheme

is not only faster already on CPUs as it reduces the scaling with the dimension of the physical Hilbert space d from d^3 down to d^2 , but unlike for the SVD-based scheme, significant speedups can be achieved on GPUs at the same accuracy.

This paper is organized as follows: In Sec. II, we briefly review MPS and introduce the QR-based truncation scheme. We elaborate on a way to dynamically adjust the MPS bond dimension in Sec. III. A detailed benchmark study is provided in Sec. IV, comparing results and runtimes between the different TEBD schemes, both on CPU and GPU hardware, before we conclude our findings in Sec. V.

II. QR-BASED TIME EVOLUTION ALGORITHM

We first review the isometric form of an MPS as shown schematically in Fig. 1. While the algorithm can be used both for finite as well as for uniform (i.e., infinite) MPS, in the following we only focus on the latter case and assume a unit cell of L sites. The MPS is parametrized by matrices $B^{[m]i}$, where i labels a basis of the local Hilbert space on site m and matrix indices are suppressed, such that

$$|\psi\rangle = \sum_{\{i_n\}} (\dots B^{[m]i_m} B^{[m+1]i_{m+1}} \dots) |\{i_n\}\rangle. \quad (1)$$

The transfer matrix for a unit cell starting on site m is given by $T_m = T^{[m]} T^{[m+1]} \dots T^{[m+L-1]}$, where $T_{(\alpha\alpha')(\beta\beta')}^{[m]} = \sum_i B_{\alpha\beta}^{[m]i} \bar{B}_{\alpha'\beta'}^{[m]i}$. We choose a right isometric form, in which the dominant right eigenvector $\rho^{[m]}$ of T_m is $\rho_{\beta\beta'}^{[m]} = \delta_{\beta\beta'}$, while its dominant left eigenvector is given by $\lambda_{\alpha\alpha'}^{[m]} = \sum_{\beta} \Xi_{\beta\alpha}^{[m]} \bar{\Xi}_{\beta\alpha'}^{[m]}$ with $\|\Xi^{[m]}\| = 1$ and both respective eigenvalues equal to 1. Moreover, $T^{[m]}$ translate the eigenvectors, i.e., $\lambda^{[m]} T^{[m]} = \lambda^{[m+1]}$ and $T^{[m]} \rho^{[m]} = \rho^{[m-1]}$, with superscripts modulo L . Note that this makes the $B^{[m]}$ isometric, in the sense that $\sum_{i\beta} B_{\alpha\beta}^{[m]i} \bar{B}_{\alpha'\beta}^{[m]i} = \delta_{\alpha\alpha'}$. An MPS in this form allows us to directly evaluate local expectation values. It is represented in memory by the $\{B^{[n]}\}$ and $\{\Xi^{[n]}\}$ for n running over a unit cell.

*jakob.unfried@tum.de

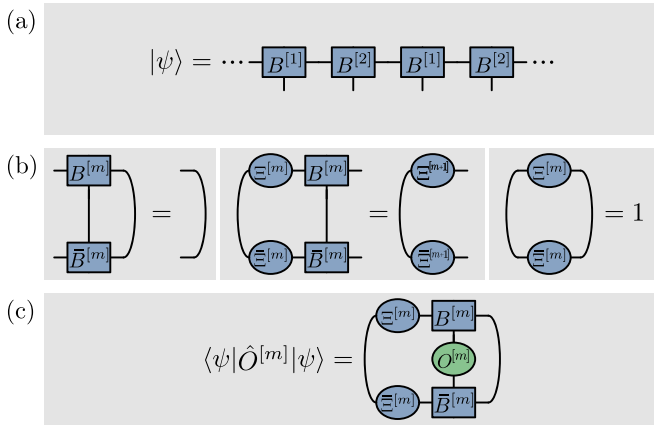


FIG. 1. (a) Uniform MPS, depicted here with a unit cell of two sites. (b) Conditions for the right isometric form; dominant right eigenvector of the transfer matrix, dominant left eigenvector, normalization choice for dominant eigenvectors. (c) The isometry conditions allow easy evaluation of local expectation values.

The isometric form does not fully fix the gauge freedom and is a weaker requirement than the canonical form of MPS [31,32], which would additionally require that the $\Xi^{[m]}$ are diagonal matrices with real positive entries in descending order, i.e., the Schmidt values Λ_α for a bipartition of the state by cutting the bond between sites m and n , $|\psi\rangle = \sum_{\alpha=1}^{\chi} |\alpha\rangle_{\leftarrow} \Lambda_\alpha |\alpha\rangle_{\rightarrow}$. Here, $|\alpha\rangle_{\leftarrow(\rightarrow)}$ denote orthonormal states on the sites left (right) of the given bond, i.e., the left (right) Schmidt states. In the isometric form we have $|\psi\rangle = \sum_{\alpha\beta} |\alpha\rangle_{\leftarrow} \Xi_{\alpha\beta}^{[n]} |\beta\rangle_{\rightarrow}$, such that the Schmidt values can be obtained as the singular values of the nondiagonal $\Xi^{[n]}$.

In order to approximate the time evolution of the MPS with respect to a Hamiltonian $H = \sum_n H_{n,n+1}$, we apply the Trotterized time evolution operator alternatingly to even and odd bonds as shown in Fig. 2(a) in the same way as in the original SVD-based infinite TEBD algorithm [5,31]. However, the update procedure for two neighboring sites m and $n = m + 1 \pmod{L}$ differs in that we do not require an SVD decomposition with a cost scaling as $d^3 \chi^3$ with the local Hilbert space dimension d and MPS bond dimension χ . Instead, the algorithm relies on two successive QR (or LQ) decompositions and scales as $d^2 \chi^3$. As shown in Fig. 2(b), the algorithm consists of three steps:

(1) We first construct a mixed representation $\theta_{\alpha\delta}^{ij} = \sum_{\beta,\gamma} \Xi_{\alpha\beta}^{[m]} B_{\beta\gamma}^{[m]i} B_{\gamma\delta}^{[n]j}$ of the state in terms of physical and virtual states. We apply the two-site gate U to the state, $\tilde{\theta}_{\alpha\delta}^{ij} = \sum_{i'j'} U_{ij}^{i'j'} \theta_{\alpha\delta}^{i'j'}$. The evolved state is then projected back into the manifold of MPS of the given bond dimension, by contracting it with the complex conjugate of the isometry $B^{[n]}$ to obtain $X_{\alpha\gamma}^i = \sum_{j,\delta} \tilde{\theta}_{\alpha\delta}^{ij} \tilde{B}_{\gamma\delta}^{[n]j}$. We group the legs of $X_{\alpha\gamma}^i \rightarrow X_{(\alpha i)\gamma}$ and perform a QR decomposition of this matrix, $X_{(\alpha i)\gamma} = \sum_{\beta} Q_{(\alpha i)\beta}^{[m]} R_{\beta\gamma}$. Ungrouping the legs $Q_{(\alpha i)\beta}^{[m]} \rightarrow Q_{\alpha\beta}^{[m]}$ yields the left isometry used in the next step.

(2) We start from the evolved state $\tilde{\theta}_{\alpha\delta}^{ij}$ and project it by contracting it with the complex conjugate of the left isometry $Q^{[m]}$ to obtain $Y_{\beta\delta}^j = \sum_{i,\alpha} \tilde{Q}_{\alpha\beta}^{[m]i} \tilde{\theta}_{\alpha\delta}^{ij}$. We group the legs of

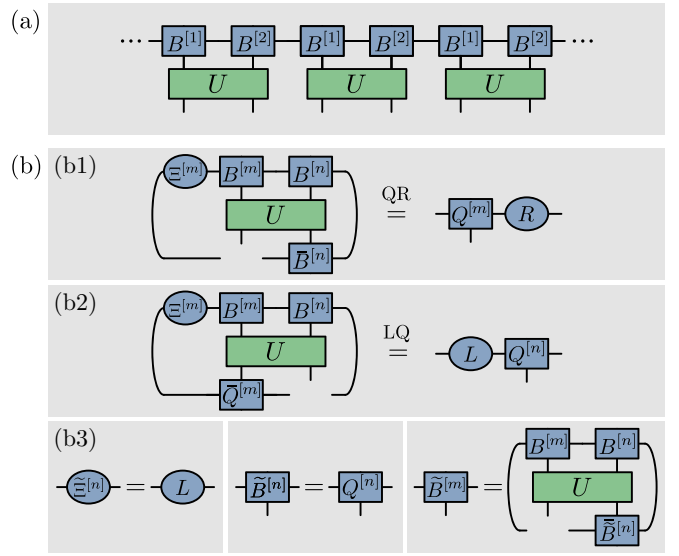


FIG. 2. Algorithm for the QR-based time evolution: (a) The time evolution is decomposed into two-site gates acting on neighboring sites. (b) Algorithm for the QR-based truncation scheme: (b1) Contraction of the time-evolved block with tensor $\tilde{B}^{[n]}$ and subsequent QR decomposition. (b2) Contraction of the time-evolved block with tensor $Q^{[m]}$ obtained in the previous step and subsequent LQ decomposition. (b3) Obtaining the updated tensors $\tilde{B}^{[m]}$, $\tilde{B}^{[n]}$, and $\tilde{\Xi}^{[n]}$.

$Y_{\beta\delta}^j \rightarrow Y_{\beta(j\delta)}$ and perform an LQ decomposition of this matrix, $Y_{\beta(j\delta)} = \sum_{\gamma} L_{\beta\gamma} Q_{\gamma(j\delta)}^{[n]}$. Ungrouping the legs $Q_{\gamma(j\delta)}^{[n]} \rightarrow Q_{\gamma\delta}^{[n]j}$ yields the right isometry used in the next step.

(3) We conclude the iteration by assigning the updated tensors: $\tilde{\Xi}_{\beta\gamma}^{[n]} = L_{\beta\gamma}$, $\tilde{B}_{\gamma\delta}^{[n]j} = Q_{\gamma\delta}^{[n]j}$, and $\tilde{B}_{\alpha\beta}^{[m]i} = \sum_{\gamma\delta i'j'} U_{ij}^{i'j'} B_{\alpha\gamma}^{[m]i'} B_{\gamma\delta}^{[n]j'} \tilde{B}_{\beta\delta}^{[n]j}$.

A few comments are in order. First, we can understand the truncation scheme as an iterative solver for finding the optimal approximation of $\tilde{\theta}$ with reduced rank $\tilde{\chi}$, i.e., $\tilde{\theta}_{(\alpha i)(j\delta)} \approx \sum_{\gamma=1}^{\tilde{\chi}} X_{(\alpha i)\gamma} Y_{\gamma(j\delta)}$. Keeping one of the components, e.g., Y , constant and demanding that it be an isometry, the optimal update for X which minimizes the distance $\|\tilde{\theta} - XY\|$ is given by $\tilde{\theta} Y^\dagger$. Before we can analogously update Y , we perform a gauge transformation via the QR decomposition, i.e., $(X, Y) \mapsto (Q, RY)$, which makes the first matrix an isometry while leaving the distance invariant. After updating Y , its LQ decomposition yields an approximation of $\tilde{\theta}$ in a suitable isometric form. In order to approximate a generic matrix $\tilde{\theta}$, we expect that these updates need to be iterated until convergence. In the specific case of Trotterized time evolution we have $U = \mathbb{1} + O(\delta t)$, where δt is small to control the Trotter error, such that the initial guess $Y_0 = B^{[n]}$ is already close to optimal, and we find that a single sweep is sufficient. The approach of iteratively solving a local (here single-site) problem, shifting the orthogonality center, and repeating until convergence is widely used for MPS compression [8], in the context of PEPS contraction [33,34], and is closely related to the DMRG algorithm. We emphasize that this provides only a heuristic intuition that the resulting approximation is sound, while reliable evidence is obtained only *a posteriori* by ob-

servicing a small truncation error. Like in the SVD-based TEBD algorithm, we have achieved a truncation of the evolved wave function which is (approximately) optimal in the local Frobenius norm of the tensor $\tilde{\theta}$. The isometric form of the MPS then guarantees that it is also (approximately) optimal in the Hilbert space norm and thus the algorithm simulates a controlled approximation to the true dynamics within the manifold of MPS of the given bond dimension $\tilde{\chi}$.

Second, the update for $\tilde{B}^{[m]}$ in the last step is motivated by Hastings' modified TEBD [35]. From our truncation scheme, just as from SVD-based truncation, we get the left MPS tensor in left isometric form, i.e., $\tilde{A}^{[m]} = Q^{[m]}$, and Hastings' modification allows us to form $\tilde{B}^{[mi]} = (\Xi^{[m]})^{-1} \tilde{A}^{[mi]} \Xi^{[n]}$ without explicit matrix inversion, which would in practice often be ill conditioned.

Lastly, the algorithm, as presented above, yields an MPS with the same bond dimension χ as the MPS before the time step. A simple heuristic method to grow the bond dimension is to replace $\tilde{B}^{[n]}$ in step (i) with an isometry Y_0 to a larger virtual space, with a dimension $\eta \in [\chi, d\chi]$ which is determined *a priori*, e.g., $\eta = \min(\chi_{\max}, d\chi)$. In practice, we could take an arbitrary η -dimensional slice of the left leg pair of $\tilde{\theta}$. A controlled method to increase the bond dimension dynamically, based on a desired bound on the truncation error, is given below.

III. CONTROLLED BOND EXPANSION

We discuss now how the MPS bond dimension can be adjusted dynamically, e.g., based on the Schmidt values of the state, as can be done in the SVD-based truncation scheme. This is in analogy to the ideas of controlled bond expansion [36–38], which originated in the context of single-site DMRG [39,40] and improves upon the uncontrolled bond expansion scheme outlined in the previous section. The algorithm is illustrated schematically in Fig. 3. We choose—*a priori*—a bond dimension $\eta = \chi + \Delta\chi \leq d\chi$ at which we perform the variational QR-based decomposition, then truncate to $\tilde{\chi} \leq \eta$, based on the Schmidt values. The optimal value of $\Delta\chi$ is model dependent and has to be chosen empirically as the sweet spot in a trade-off between computational cost, which scales as $d^2\eta\chi^2$, and the amount of entanglement which can be represented, for which η gives an upper bound. In practice we find that an increase of $\sim 10\%$ at each time step is sufficient for the cases considered. Next, we require an initial guess for the MPS tensor on site n , which allows us to enlarge the dimension of the virtual Hilbert space. In step (i), we choose an arbitrary η -dimensional slice on the left leg pair of the time-evolved block, i.e., $(Y_0)_{\alpha'\beta}^j = \sum_{\alpha i} (P_\eta)_{\alpha'\alpha i} \tilde{\theta}_{(\alpha i)(\beta j)}$, with the $\eta \times d\chi$ projection matrix $(P_\eta)_{\alpha\beta} = \delta_{\alpha\beta}$. The following steps (ii) and (iii) involve a QR and an LQ decomposition and are performed exactly as described in the preceding section. In step (iv), we diagonalize the Hermitian matrix $L^\dagger L = V^\dagger S^2 V$, where S^2 is a diagonal matrix containing the real, non-negative eigenvalues. Note that S are the singular values of L , i.e., the Schmidt values of the state, and we could have computed S and V via an SVD of L . This appears to be significantly slower on the GPU, however. By discarding the smallest singular values in S , along with

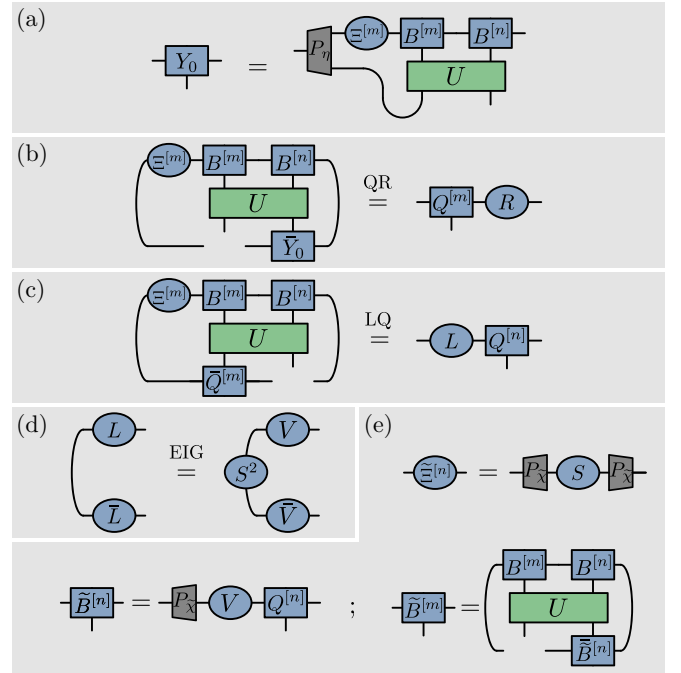


FIG. 3. Algorithm for the QR-based time evolution with controlled bond expansion: (a) Obtain an initial guess Y_0 by projecting/slicing the evolved wave function. (b), (c) Decomposition of the evolved wave function, similar to Sec. II and Fig. 2. (d) Diagonalization of $L^\dagger L$ yields two of the three matrices comprising the SVD of L . (e) Truncation of the implicit SVD to bond dimension $\tilde{\chi}$, obtaining the updated tensors.

corresponding rows of V , a truncation to a bond dimension of $\tilde{\chi}$ is achieved, controlled, e.g., by a desired truncation error and/or a threshold below which Schmidt values are neglected. In step (v), we finally absorb the projected unitary V into $Q^{[n]}$, then update the MPS tensors as in the previous section, that is, $\tilde{B}_{\alpha\delta}^{[nj]} = \sum_{\beta\gamma} (P_{\tilde{\chi}})_{\alpha\beta} V_{\beta\gamma} Q_{\gamma\delta}^{[nj]}$, $\tilde{\Xi}_{\alpha\delta}^{[n]} = (P_{\tilde{\chi}})_{\alpha\beta} S_{\beta\gamma} (P_{\tilde{\chi}})_{\gamma\delta}$, and $\tilde{B}_{\alpha\beta}^{[mi]} = \sum_{\gamma\delta i' j' j} U_{ij}^{i' j'} B_{\alpha\gamma}^{[mi] i'} B_{\gamma\delta}^{[n] j'} \tilde{B}_{\beta\delta}^{[nj]}$, where $P_{\tilde{\chi}}$ is the projection matrix realizing the truncation, i.e., keeping the largest $\tilde{\chi}$ Schmidt values. If applied to all bonds and if truncation errors are negligible, the bond expansion scheme with its implicit SVD brings the MPS to the canonical form, where the $\Xi^{[m]}$ are diagonal matrices containing the Schmidt values.

The controlled bond expansion is crucial when exploiting symmetries via a block structure which these impose [41,42]. The scheme as outlined in Sec. II would need to make a choice about the block structure, in particular the size of the individual blocks *a priori*. By using $Y_0 = B^{[n]}$, for example, they are chosen to be the same as before the time step. The controlled bond expansion, however, allows us to dynamically choose the block dimensions χ_i optimally, in the sense of minimal truncation error, just like in the SVD-based TEBD scheme. Therefore, even when the total bond dimension $\chi = \sum_i \chi_i$ is already saturated, one can expand each block $\chi_i \rightarrow \chi_i + \Delta\chi_i$ and subsequently truncate back by keeping only at most χ dominant contributions—this allows us to dynamically adjust the size of the individual blocks.

IV. BENCHMARK

We choose the d -state quantum clock model to benchmark the algorithm. This model is generically nonintegrable ($d > 2$) and allows us to highlight the scaling with the physical Hilbert space dimension d . The Hamiltonian reads

$$H = - \sum_n (Z_n Z_{n+1}^\dagger + \text{H.c.}) - g \sum_n (X_n + \text{H.c.}), \quad (2)$$

where the clock operators

$$Z = \begin{pmatrix} 1 & & & & \\ & \omega & & & \\ & & \omega^2 & & \\ & & & \ddots & \\ & & & & \omega^{d-1} \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & 0 & \ddots & \\ & & & \ddots & 1 \\ 1 & & & & 0 \end{pmatrix} \quad (3)$$

are $d \times d$ generalization of Pauli matrices and $\omega = e^{2\pi i/d}$. The model has a global \mathbb{Z}_d symmetry generated by $\prod_i X_i$, which we do not exploit in the numerical simulations. For $d \leq 4$, the model has a critical point at $g = 1$, while there is an extended critical region around $g = 1$ for $d \geq 5$ [43,44]. We start with the $Z = 1$ product state and evolve it in time with the $g = 2$ Hamiltonian.

We consider four algorithmic variations of the truncation scheme: (i) via SVD, $\tilde{\theta} = USV^\dagger$, (ii) the same decomposition but numerically evaluated by diagonalizing $\tilde{\theta}^\dagger \tilde{\theta} = VS^2V^\dagger$ (note that U is not actually required), which we dub EIG, (iii) the simple QR-based scheme we have introduced in Sec. II, and (iv) the QR scheme with controlled bond expansion (QR+CBE) as described in Sec. III. We run the benchmark on a NVIDIA A100 GPU (80 GB RAM) with CUDA version 11.7, as well as an AMD EPYC 7763 CPU with 64 physical cores and MKL version 2019.0.5. The two units have similar power consumption: 300 W and 280 W thermal design power, respectively. All simulations are performed in double precision (i.e., complex128). The implementation used for the benchmark and the data are available in the Supplemental Material [45].

In Fig. 4, we perform full TEBD simulations of the quench protocol for a $d = 5$ clock model. We run the simulation beyond times where the approximation of the evolved state as an MPS of the given bond dimension breaks down, as quantified by a large truncation error. In the time regime of acceptable error $\epsilon_{\text{trunc}} \lesssim 10^{-5}$, that is, until $t \lesssim 2$ depending on bond dimension, we observe excellent agreement between the different TEBD schemes in the extracted expectation values $\langle Z \rangle$ and entanglement entropy S_{vN} up to relative deviations of $10^{-11} \sim 10^{-12}$. For the QR-based scheme, we do not have access to all singular values of $\tilde{\theta}$, from which the truncation error is extracted in SVD-based TEBD. We instead explicitly compute the distance between the evolved wave function $\tilde{\theta}$ and its low rank approximation.

In Fig. 5, we benchmark runtimes for the core algorithmic step of contracting and subsequently decomposing the evolved wave function $\tilde{\theta}$ for all combinations of truncation scheme and hardware, as well as a range of Hilbert space dimensions d . We clearly observe the improved scaling of the QR-based algorithm, which is quadratic in d instead of

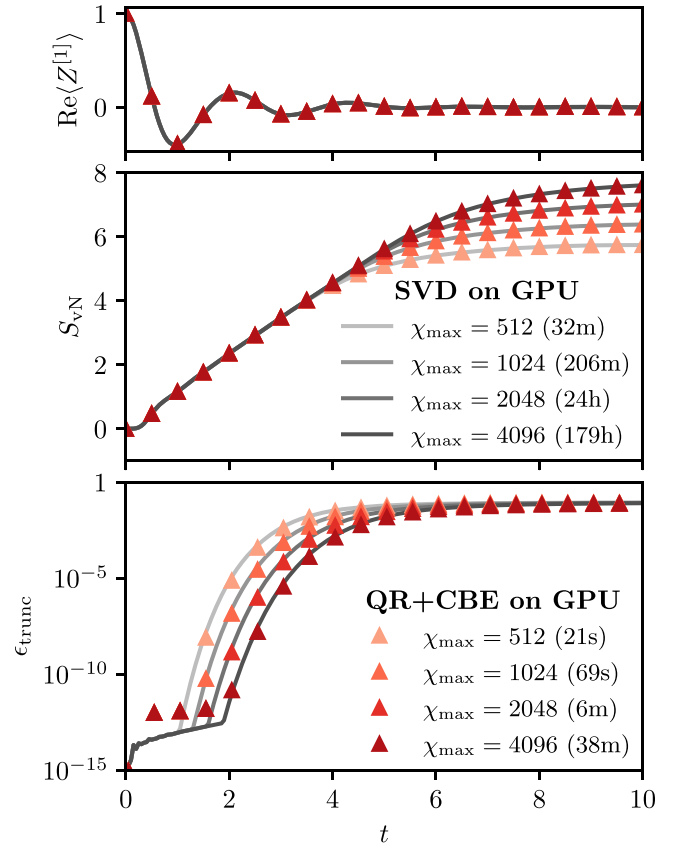


FIG. 4. TEBD simulation of a global quench in the $d = 5$ quantum clock model from $g = 0$ to $g = 2$ with a time step of $\delta t = 0.05$. We show a local Z expectation value (top), the half-chain von Neumann entanglement entropy (center), and truncation error (bottom). We compare data from SVD-based (solid lines) and QR-based (triangles) TEBD simulations at a range of bond dimensions χ_{max} (colors). For the QR-based scheme, we employ controlled bond expansion with $\eta = \max(100, 1.1\chi)$ and plot only every tenth data point. For both schemes, we discard Schmidt values smaller than 10^{-14} and keep at most χ_{max} of them. Time in the legend denotes the total wall time needed for each simulation, i.e., to generate the shown data from scratch.

cubic, as well as a speedup of one to two orders of magnitude from hardware acceleration for EIG and QR based algorithms. For example, the QR-based truncation scheme on the GPU with $\chi = 1024$, $d = 20$ reaches a speedup factor of 2700 compared to the SVD-based scheme on the same GPU and 750 compared to SVD on CPU.

V. CONCLUSION

We proposed and benchmarked a modified time-evolving block decimation (TEBD) algorithm that uses a variational truncation scheme based on the QR decomposition instead of the singular value decomposition (SVD). We demonstrated that the QR-based truncation scheme allows simulation of the time evolution of MPS to the same degree of accuracy, but compared to the SVD-based scheme drastically decreases runtime and power consumption needed to obtain

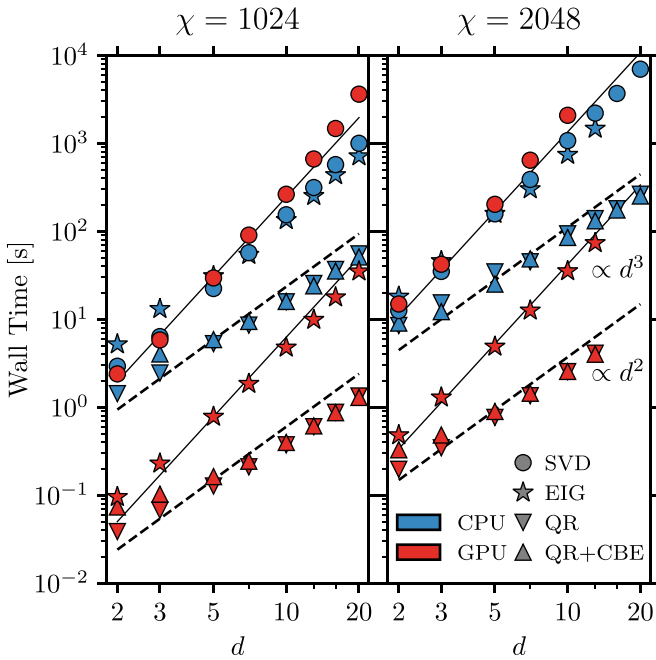


FIG. 5. Timing benchmark for the application of a single gate to an MPS for different hardware (marker colors) and truncation schemes (marker shapes). We give the average wall time needed to compute the updated tensors $\tilde{B}^{[m]}$, $\tilde{\Xi}^{[m]}$, $\tilde{B}^{[n]}$ from the old tensors $\Xi^{[m]}$, $B^{[m]}$, $B^{[n]}$ and U . For the QR-based algorithm with (without) controlled bond expansion (CBE), these are the steps illustrated in Fig. 3 (Fig. 2) and Ξ are diagonal (nondiagonal) matrices. In the case of SVD (EIG) based truncation, the task consists of first contracting $\tilde{\theta}$, performing an SVD of $\tilde{\theta}$ (diagonalizing $\tilde{\theta}^\dagger \tilde{\theta}$), and finally contracting $\tilde{B}^{[m]}$. For CBE, we choose $\Delta\chi = 0.1\chi$, the same expansion rate as for Fig. 4. The initial MPS has a bond dimension χ and the evolved state is truncated to $\tilde{\chi} = \chi$. Solid (dashed) lines are power laws with the expected cubic (quadratic) scaling with the physical dimension d . The missing data points for large d in the right panel were not possible to obtain due to memory limitations.

the same results, especially when run on GPU hardware. The improved scaling with the local Hilbert space dimension d implies substantial performance increase even on CPU for large d , e.g., in simulations of open systems [46] or bosonic systems.

We expect that with small changes, the algorithm can be used to accelerate MPS truncation in a broader class of algorithmic settings, e.g., to apply long-range gates arising from interactions beyond nearest neighbors or in the effective 1D description of two-dimensional models, time evolution based on applying MPOs [47], or DMRG [1]. An application to the simulation of quantum circuits would need to be investigated in further detail, since unlike for the Trotterized time evolution with small time steps, the unitary gates in a generic quantum circuit need not be close to unity. Hardware acceleration on the heavily specialized tensor processing units (TPUs) [30,48] may yield an even greater performance increase and make larger bond dimensions accessible via large memory and distributed linear algebra, allowing the simulation to represent more entanglement. For the simulation of finite systems, parallel gate application can provide further performance increase, as demonstrated in Ref. [49] for the time-dependent variational principle.

ACKNOWLEDGMENTS

This research was financially supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program under Grant Agreement No. 771537. F.P. acknowledges the support of the Deutsche Forschungsgemeinschaft (DFG; German Research Foundation) under Germany's Excellence Strategy EXC-2111-390814868. F.P.'s research is part of the Munich Quantum Valley, which is supported by the Bavarian state government with funds from the Hightech Agenda Bayern Plus.

- [1] S. R. White, *Phys. Rev. Lett.* **69**, 2863 (1992).
- [2] U. Schollwöck, *Ann. Phys.* **326**, 96 (2011).
- [3] S. R. White and A. E. Feiguin, *Phys. Rev. Lett.* **93**, 076401 (2004).
- [4] A. J. Daley, C. Kollath, U. Schollwöck, and G. Vidal, *J. Stat. Mech.: Theory Exp.* (2004) P04005.
- [5] G. Vidal, *Phys. Rev. Lett.* **93**, 040502 (2004).
- [6] J. Haegeman, J. I. Cirac, T. J. Osborne, I. Pižorn, H. Verschelde, and F. Verstraete, *Phys. Rev. Lett.* **107**, 070601 (2011).
- [7] J. Haegeman, C. Lubich, I. Oseledets, B. Vandereycken, and F. Verstraete, *Phys. Rev. B* **94**, 165116 (2016).
- [8] S. Paeckel, T. Köhler, A. Swoboda, S. R. Manmana, U. Schollwöck, and C. Hubig, *Ann. Phys.* **411**, 167998 (2019).
- [9] A. Dang, C. D. Hill, and L. C. L. Hollenberg, *Quantum* **3**, 116 (2019).
- [10] Y. Zhou, E. M. Stoudenmire, and X. Waintal, *Phys. Rev. X* **10**, 041038 (2020).
- [11] M. Gohlke, R. Verresen, R. Moessner, and F. Pollmann, *Phys. Rev. Lett.* **119**, 157203 (2017).
- [12] W. Kadow, L. Vanderstraeten, and M. Knap, *Phys. Rev. B* **106**, 094417 (2022).
- [13] P. N. Jepsen, W. W. Ho, J. Amato-Grill, I. Dimitrova, E. Demler, and W. Ketterle, *Phys. Rev. X* **11**, 041054 (2021).
- [14] C. Kollath, A. M. Läuchli, and E. Altman, *Phys. Rev. Lett.* **98**, 180601 (2007).
- [15] S. Trotzky, Y.-A. Chen, A. Flesch, I. P. McCulloch, U. Schollwöck, J. Eisert, and I. Bloch, *Nat. Phys.* **8**, 325 (2012).
- [16] J. H. Bardarson, F. Pollmann, and J. E. Moore, *Phys. Rev. Lett.* **109**, 017202 (2012).
- [17] T. Chanda, P. Sierant, and J. Zakrzewski, *Phys. Rev. B* **101**, 035148 (2020).
- [18] E. V. Doggen, I. V. Gornyi, A. D. Mirlin, and D. G. Polyakov, *Ann. Phys.* **435**, 168437 (2021).
- [19] A. Nietner, A. Kshetrimayum, J. Eisert, and B. Lake, *arXiv:2207.10696*.
- [20] T. Prosen and M. Žnidarič, *J. Stat. Mech.: Theory Exp.* (2009) P02035.
- [21] T. Rakovszky, C. W. von Keyserlingk, and F. Pollmann, *Phys. Rev. B* **105**, 075131 (2022).

- [22] B. Bertini, F. Heidrich-Meisner, C. Karrasch, T. Prosen, R. Steinigeweg, and M. Žnidarič, *Rev. Mod. Phys.* **93**, 025003 (2021).
- [23] M. Schulz, S. R. Taylor, C. A. Hooley, and A. Scardicchio, *Phys. Rev. B* **98**, 180201(R) (2018).
- [24] B. Kloss and Y. Bar Lev, *Phys. Rev. B* **102**, 060201(R) (2020).
- [25] N. Darkwah Oppong, G. Pasqualetti, O. Bettermann, P. Zechmann, M. Knap, I. Bloch, and S. Fölling, *Phys. Rev. X* **12**, 031026 (2022).
- [26] W. Li, J. Ren, and Z. Shuai, *J. Chem. Phys.* **152**, 024127 (2020).
- [27] F. Pan and P. Zhang, *Phys. Rev. Lett.* **128**, 030501 (2022).
- [28] M. Hauru, A. Morningstar, J. Beall, M. Ganahl, A. Lewis, and G. Vidal, [arXiv:2111.10466](https://arxiv.org/abs/2111.10466).
- [29] A. Morningstar, M. Hauru, J. Beall, M. Ganahl, A. G. M. Lewis, V. Khemani, and G. Vidal, *PRX Quantum* **3**, 020331 (2022).
- [30] M. Ganahl, J. Beall, M. Hauru, A. G. Lewis, J. H. Yoo, Y. Zou, and G. Vidal, *PRX Quantum* **4**, 010317 (2023).
- [31] G. Vidal, *Phys. Rev. Lett.* **98**, 070201 (2007).
- [32] G. Vidal, J. I. Latorre, E. Rico, and A. Kitaev, *Phys. Rev. Lett.* **90**, 227902 (2003).
- [33] F. Verstraete and J. I. Cirac, [arXiv:cond-mat/0407066](https://arxiv.org/abs/cond-mat/0407066).
- [34] M. Lubasch, J. I. Cirac, and M.-C. Bañuls, *New J. Phys.* **16**, 033014 (2014).
- [35] M. B. Hastings, *J. Math. Phys.* **50**, 095207 (2009).
- [36] A. Gleis, J.-W. Li, and J. Von Delft, [arXiv:2207.14712](https://arxiv.org/abs/2207.14712).
- [37] J.-W. Li, A. Gleis, and J. Von Delft, [arXiv:2208.10972](https://arxiv.org/abs/2208.10972).
- [38] A. Gleis, J.-W. Li, and J. von Delft, *Phys. Rev. B* **106**, 195138 (2022).
- [39] S. R. White, *Phys. Rev. B* **72**, 180403(R) (2005).
- [40] C. Hubig, I. P. McCulloch, U. Schollwöck, and F. A. Wolf, *Phys. Rev. B* **91**, 155115 (2015).
- [41] S. Singh, R. N. C. Pfeifer, and G. Vidal, *Phys. Rev. A* **82**, 050301(R) (2010).
- [42] S. Singh, R. N. C. Pfeifer, and G. Vidal, *Phys. Rev. B* **83**, 115125 (2011).
- [43] G. Sun, T. Vekua, E. Cobanera, and G. Ortiz, *Phys. Rev. B* **100**, 094428 (2019).
- [44] G. Ortiz, E. Cobanera, and Z. Nussinov, *Nucl. Phys. B* **854**, 780 (2012).
- [45] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/PhysRevB.107.155133> for an example PYTHON implementation and raw data of the benchmark presented in Figs. 4 and 5.
- [46] F. Verstraete, J. J. García-Ripoll, and J. I. Cirac, *Phys. Rev. Lett.* **93**, 207204 (2004).
- [47] M. P. Zaletel, R. S. K. Mong, C. Karrasch, J. E. Moore, and F. Pollmann, *Phys. Rev. B* **91**, 165112 (2015).
- [48] A. G. Lewis, J. Beall, M. Ganahl, M. Hauru, S. B. Mallick, and G. Vidal, *Proc. Natl. Acad. Sci. USA* **119**, e2122762119 (2022).
- [49] P. Secular, N. Gourianov, M. Lubasch, S. Dolgov, S. R. Clark, and D. Jaksch, *Phys. Rev. B* **101**, 235123 (2020).