PAPER • OPEN ACCESS

Propagating large open quantum systems towards their asymptotic states: cluster implementation of the time-evolving block decimation scheme

To cite this article: Valentin Volokitin *et al* 2019 *J. Phys.: Conf. Ser.* **1392** 012061

View the article online for updates and enhancements.

Propagating large open quantum systems towards their asymptotic states: cluster implementation of the time-evolving block decimation scheme.

Valentin Volokitin[1], **Ihor Vakulchyk**[2,3], **Evgeny Kozinov**[1], **Alexey Liniov**[1], **Iosif Meyerov**[1], **Michail Ivanchenko**[1], **Tatyana Laptyeva**[1], **Sergey Denisov**[4]

[1]Lobachevsky State University of Nizhny Novgorod, Russia
[2]Institute for Basic Science, Daejeon, Korea
[3]Korea University of Science and Technology, Daejeon, Korea
[4]Oslo Metropolitan University, N-0130 Oslo, Norway

E-mail: valentin.volokitin@itlab.unn.ru

Abstract. Many-body quantum systems are subjected to the Curse of Dimensionality: The dimension of the Hilbert space $\mathcal{H}$, where these systems live in, grows exponentially with number of their components. However, with some systems, it is possible to escape the curse by using a low-rank tensor approximation known as "matrix-product state/operator (MPS/O) representation" in the quantum community and "tensor-train decomposition" among applied mathematicians. Motivated by recent advances in computational quantum physics, we consider chains of N spins coupled by nearest-neighbor interactions. The spins are subjected to an action coming from the environment. Spatially disordered interaction and environment-induced decoherence drive systems into non-trivial asymptotic states. The dissipative evolution is modeled with a Markovian master equation in the Lindblad form. By implementing the MPO technique and propagating system states with the time-evolving block decimation scheme, which allows keeping the length of the state descriptions fixed, it is in principle possible to reach the asymptotic states. We propose and realize a cluster implementation of this idea. The implementation on four nodes allowed us to resolve the asymptotic states of the model systems with $N = 128$ spins (total dimension of the Hilbert space $\dim\mathcal{H} = 2^{128} \approx 10^{39}$).

1. Introduction

Many-body systems are at the focus of the current research in theoretical and experimental quantum physics. In addition to their fundamental importance for quantum thermodynamics and information [1], these systems are perspective from the technological point of view; e.g., all manufactured (by now) quantum computers are based on arrays of interacting superconducting qubits [2].

All real-life quantum systems are open, meaning that they interact – to a different extent – with their environments [3]. This 'action from outside', termed "decoherence" or "dissipation", works together with the unitary evolution stemming from system's Hamiltonians and, on large time scales, these joint efforts result in the creation of an asymptotic stationary state. The evolution of an open quantum system towards its asymptotic states is usually modeled with a Markovian master equation, which describes the dynamics of the system density operator

$\varrho(t)$, $\dot{\varrho}(t) = \mathcal{L}\varrho(t)$ [3]. Formally, similar to the Schroedinger equation used to describe unitary evolution of an isolated quantum system, this is a linear differential equation which can be solved numerically, e.g., by diagonalizing generator of evolution $\mathcal{L}$.

However, computational studies of many-body quantum systems are limited by the so-called Course of Dimensionality: the total length L of description (number of parameters required to specify a state) of an isolated quantum system consisting of N components (spins, qubits, ions, etc.), each one with d degrees of freedom, scales as $L(N) \sim d^N$. To specify an *arbitrary* state of a system of 50 qubits one needs $2^{50} \approx 10^{15}$ complex-valued parameters. This exceeds the memory capacity of the supercomputer "Titan" [4]. In the case of open quantum systems, the complexity squares: to describe a density operator one needs $L(N) \sim d^{2N}$ real-valued parameters.

This is a famous problem in modern data science – manipulations (or even simply storing) with data tensors becomes impossible when the data are sorted in high-dimensional spaces. The attempts to break the curse led to the development of a variety of low-rank tensor approximation algorithms [5]. These algorithms are used now in signal processing, computer vision, data mining, and neuroscience [6]. The most robust algorithms are based on Singular Value Decomposition (SVD), and one particularly efficient for multilinear algebra manipulations is the so-called Tensor-Train (TT) decomposition [7]. In physical literature, it is commonly referred to as Matrix Product State (MPS) [or Matrix Product Operator (MPO)] representation [8]. While these two names are used simultaneously (though in different fields), the underlying mathematical structure is the same [9]. The MPS/MPO/TT approach allows to reduce descriptions of *some* many-body states to a linear scaling $L(N) \sim N$[7].

The MPS/MPO representation allows for effective propagation of quantum many-body systems in time by using the so-called Time-Evolving Block Decimation (TEBD) scheme [10]. In short, this is a procedure to reduce the description of the state, obtained after every propagation step, to a given fixed length L_{cut}. The accuracy of the propagation is controllable through L_{cut}: If the information is thrown out after the restriction is substantial, the used TEBD propagation is bad and leads to a wrong description. Otherwise, it is good. Some many-body systems 'behave' well during the TEBD propagation and so the amount of the neglected information is tolerable (we are not going to discuss physical properties underlying such a 'good behavior' and refer the reader to an extensive literature on the subject; see. e.g., Ref. [8]). Important is that the MPO/TT-TEBD scheme can be used to propagate open systems [11] and thus get in touch with the corresponding asymptotic states [12, 13]. It is crucial therefore to estimate computational resources needed for the realization of this program. Here we report the results of our studies in these directions.

2. The algorithm

2.1. Tensor-Train Decomposition and Time-Evoling Block Decimation Propagation

Here we mainly follow works [7] and [8]; for more details, we refer the interested reader to them.

We start with the TT representation of a N-dimensional complex-valued tensor $A^{i_1,i_2 \dots i_N}$ with $i_k = 1, 2, \dots M$ [7],

$$A\left[i_1, i_2 \dots i_N\right] = \sum_{\alpha_1 \dots \alpha_{N-1}} \Gamma^{[1]i_1}_{\alpha_0,\alpha_1} \lambda^{[1]}_{\alpha_1} \Gamma^{[2]i_2}_{\alpha_1,\alpha_2} \lambda^{[2]}_{\alpha_2} \dots \Gamma^{[N-2]i_{N-1}}_{\alpha_{N-2},\alpha_{N-1}} \lambda^{[N-1]}_{\alpha_{N-1}} \Gamma^{[N]i_N}_{\alpha_{N-1},\alpha_N}. \tag{1}$$

One may interpret this structure as a "train" of Γ's that encode local structure in each dimension, and λ's that quantify correlations between them. Each $\Gamma^{[k]}$ is an array of M matrices $r_{k-1} \times r_k$ with restrictions $r_j \le M \max\left(r_{j-1}, r_{j+1}\right)$ with boundary conditions $r_0 = r_N = 1$. Thus, the dimensions of the matrices are $1 \times M$, $M \times M^2$, $M^2 \times M^3 \dots M^2 \times M$, $M \times 1$, which corresponds to the full representation with M^N complex parameters. The construction of the TT representation invlove SVD operations (their number is proportional to the number

Algorithm 1 : TEBD method implementation

```
1: upload: system & method parameters (N, $T_j[i_j, i'_j]$, $T_{j,j+1}[i_j, i'_j, i_{j+1}, i'_{j+1}]$, $dt$, $T_{\max}$, $R$),
   initial state ($\Gamma^{[j]i_j}_{\alpha_{j-1}\alpha_j}$, $\lambda^{[j]}_{\alpha_j}$)
2: for $t = 0$ to $T_{\max}$ do
3:     propagate all $\Gamma^{[j]i_j}_{\alpha_{j-1}\alpha_j}$ on $[t; t + dt/2]$
4:     propagate $\Gamma^{[j]i_j}_{\alpha_{j-1}\alpha_j}$, $\lambda^{[j]}_{\alpha_j}$, $\Gamma^{[j+1]i_{j+1}}_{\alpha_j\alpha_{j+1}}$ with odd $j$ on $[t; t + dt/2]$ ▷ 4-6 do with hard cutoff of the local bond dimension
5:     propagate $\Gamma^{[j]i_j}_{\alpha_{j-1}\alpha_j}$, $\lambda^{[j]}_{\alpha_j}$, $\Gamma^{[j+1]i_{j+1}}_{\alpha_j\alpha_{j+1}}$ with even $j$ on $[t; t + dt]$
6:     propagate $\Gamma^{[j]i_j}_{\alpha_{j-1}\alpha_j}$, $\lambda^{[j]}_{\alpha_j}$, $\Gamma^{[j+1]i_{j+1}}_{\alpha_j\alpha_{j+1}}$ with odd $j$ on $[t + dt/2; t + dt]$
7:     propagate all $\Gamma^{[j]i_j}_{\alpha_{j-1}\alpha_j}$ on $[t + dt/2; t + dt]$
8: end for
9: save results
10: release memory
```

of the system components). When SVD is performed, one can keep only certain singular values based on the approximation criterion. One possibility is to discard all values smaller than a fixed number. An alternative approach is to introduce a so-called **bond dimension** R, a cut-off value such that on each bound i only singular values $\lambda^{[i]}_j$, $j \leq R$, are kept and the rest are truncated. We use the latter option. Each such local approximation on the set of singular value $\{\lambda^{[i]}\}$ introduces a truncation error, $E_i(R) = \sum_{j>R} \left(\lambda^{[i]}_j\right)^2$.

The TT representation provides a basis for an approximate tensor propagation algorithms. Here we use Time-Evolving Block Decimation (TEBD) scheme [14, 11], which was specifically designed for quantum systems but applicable also in the general case. Consider a tensor flow governed by an evolution generator consisting only of operations acting on one or two adjacent dimensions,

$$\frac{d}{dt}A[i_1 \ldots i_N] = \sum_j \sum_{i_j} T^{[1]j}_{i_j} A[i_1 \ldots i_j \ldots i_N] + \sum_{j_1,j_2} \sum_{i_{j_1},i_{j_2}} T^{[2]j}_{i_{j_1},i_{j_2}} A[i_1 \ldots i_{j_1} i_{j_2} \ldots i_N], \tag{2}$$

where operations $T^{[1]}$ and $T^{[2]}$ act only on the i-th component and a pair of components, respectively. We use standard time discretization to iteratively integrate this equation (starting from some initial tensor). In terms of operations the solution reads

$$A(t+dt) = L(dt)A(t) = \; = \exp\left[\left(\sum_j \hat{T}^{[1]j} + \hat{T}^{[2]j}\right) dt\right] A(t). \tag{3}$$

As T operators generally do not commute, we have to approximate the matrix exponents. To minimize the error, it is convinient to separate the operators into groups as large as possible and such that all the operators belonging to one group commute with each other. All one-dimension operators commute by default, and two-dimension acting on odd/even pairs commute within their oddity groups. We use modified second order Suzuki-Trotter decomposition [8]. As all operators are commuting by construction, corresponding computation can be parallelized. Each two-index operator may include a cut-off if after the reorthogonalizationm, the number of singular value exceeds bound dimension R. Corresponding accumulated truncation error $E(t, R)$ is then calculated as a sum of local errors over all the operation during evolution up to time t.

Computations are dominated by SVD, so resulting complexity is $\mathcal{O}(NR^3)$, where R is the bond dimension. With $\mathcal{O}(N)$ cores available, it becomes $\mathcal{O}(R^3)$ and thus the computational task is perfectly scalable.

2.2. Lindblad Equation

We apply both TT and TEBD methods to evolve numerically many-body open quantum models. The state of such systems is described by a density matrix $\varrho(t)$ of the size $M^N \times M^N$, where N is the number of particles/spins and M is number of the local states, which we put to $M = 2$ for a 1/2-spins that we consider in the paper. Evolution of a quantum system in contact with the environment is governed by Lindblad equation [3]

$$\dot{\varrho}(t) = \mathcal{L}\varrho(t) = \mathcal{L}_H\varrho(t) + \mathcal{L}_{\text{dis}}\varrho(t) = -i\,[H, \varrho(t)] + \sum_{s=1}^{M} \gamma_s \left[D_s\varrho(t)D_s^\dagger - \frac{1}{2}\{D_s^\dagger D_s, \varrho(t)\} \right], \quad (4)$$

where $\mathcal{L}$ is the Lindblad superoperator consisting of conservative $\mathcal{L}_H$ and dissipative $\mathcal{L}_{\text{dis}}$ parts, H is Hamiltonian, D_s are dissipation operators, and γ_s are corresponding dissipation rates. There is a stationary state solution for any Lindblad superoperator $\mathcal{L}\varrho(\infty) = 0$ which is unique (aside of special cases of symmetries which we do not address here). Finally, many-body density operator ϱ can be represented as an $2N$-dimensional tensor $\varrho\,[i_1, i_2 \dots i_N; i'_1, i'_2 \dots i'_N]$ where every pair of indexes i_j, i'_j (each one runs from 1 to 2) correspond to the j-th qubit/spin.

3. Implementation

The method described in Section II is implemented as shown in Algorithm 1. The algorithm is implemented using the C++ programming language. We found, that the matrix operations (mainly SVD) are the most time-consuming parts of the algorithm. In this regard, we employ the Armadillo software library integrated with highly optimized mathematical routines from the Intel Math Kernel Library to improve performance. Finally, Armadillo/MKL routines take about 50-80% of computation time during the propagation step depending on the current system state.

The algorithm assumes performing a set of integration operations for individual components of the system at every time step. These operations are not independent but can be ordered according to their dependencies for the organization of parallel computations. In particular, all one- and two-particle interactions can be performed completely in parallel.

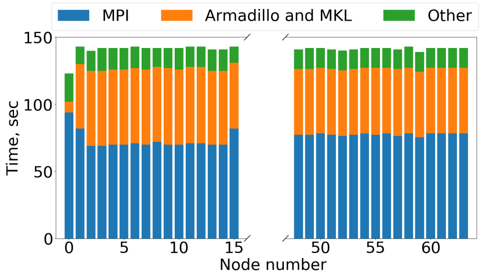

Figure 1. Distribution of computational and communication functions run time. 64 MPI-processes were executed on four nodes of the cluster.

The cluster parallelization is done by using the MPI technology. We apply the classic master-worker scheme for parallelization of the algorithm. For that, the single managing MPI-process (master) forms separate tasks for one– and two–particle interactions, monitors their dependencies from each other and readiness, distributes tasks to all other processes (workers) and accumulates the results.

All computational experiments have been done on the Lobachevsky cluster with a 2 × 8-core Intel Xeon CPU E5-2660, 2.20GHz, 64 GB RAM, Infiniband QDR interconnect. The code was compiled with the Intel C++ Compiler, Intel Math Kernel Library and Intel MPI from the Intel Parallel Studio XE suite of development tools and the Armadillo library.

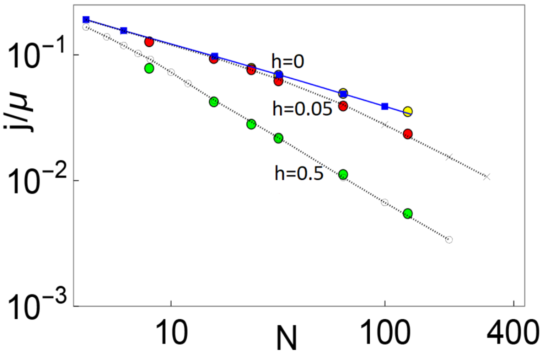

Figure 2. Scaling of the spin current j through a disordered spin chain with N spins for different values of disorder strength h. Our results (big colored circles) are plotted on top of the results reported in Ref. [12]). The maximal size of the model system used in our simulations is $N = 128$. For every set of parameters, we performed averaging over 20 disorder realizations. The propagation time step $dt = 0.1$ and bond dimension $R = 50$.

4. Results

As test-beds we used two models of open spin chains with next-neigbor couplings [12, 13].

We integrate model from Ref. [12] with following parameters: $N = 128$, $R = 50$, $T_{\max} = 50$, $dt = 0.1$. Parallel code was run on four computational nodes of the cluster (1 MPI-process per CPU core, 64 MPI-processes overall). Total computation time was 143 s. The resulted diagram for the distribution of computational and communication functions run time is presented in Fig. 1. It is shown that the calculations are fairly well balanced, which is an undoubted advantage of the parallelization scheme. However, MPI communications take a significant part of the computation time, while further increasing the number of cluster nodes used will not significantly speed up the calculations, which is a limitation of the scheme. Computational efficiency (ratio of computation time to total execution time) was 47%.

We find that it is possible to reproduce - with high accuracy – the results reported in Ref. [12] by using bond-dimension $R = 50$. On Fig. 2 we present a comparison of the results of the sampling we perform with our code (big circles; yellow, red and green) with the results by Žnidarič and his co-authors. We use propagation step $dt = 0.1$ and propagate every system up to $t = 10^4$, irrespectively of its size. For every value of N and disorder strength h, we

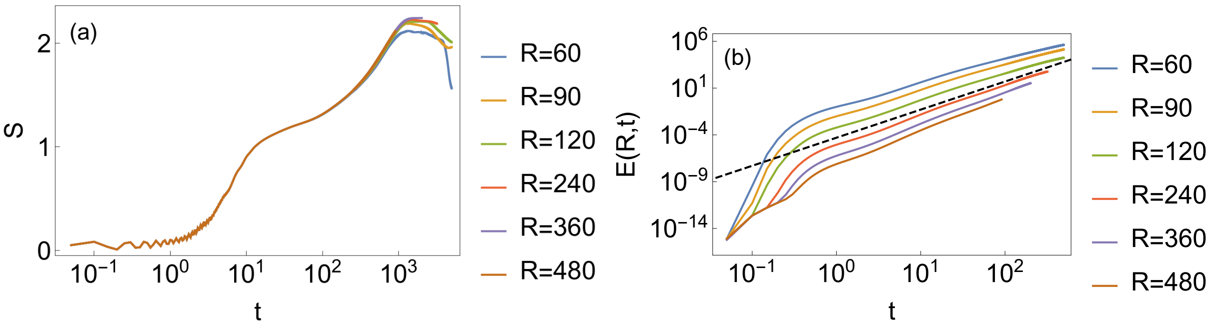

Figure 3. (a) Evolution of the operator entanglement entropy S for a single disorder realization for the model from Ref. [13], for different values of bond dimension R. The propagation time step $dt = 0.1$ and the system size is $N = 128$. Note that for $R = 480$ we did not reach the asymptotic 'plateau' because it was not possible to numerically propagate system further (we hit the two-week limit). (b) Increase of the accumulated truncation error in time.

additionally performed averaging for 20 disorder realizations. Each realization took from 2 minutes to 2 hours depending on the size of the system and involved up to two nodes (for large system sizes, $N > 64$).

For the model from Ref. [13] we study the dynamics of the operator entanglement entropy (for a fixed disorder realization) for different bond dimensions as shown on Fig. 3a. We found that $R_c = 360$ constitutes a threshold value after which the asymptotic entropy does not change upon further increase of the bond dimension. The calculation time for this value of bond dimension was four weeks of continuous propagation on four computing nodes. We also analyze also the evolution of accumulated error $E(R, t)$ in this case. The operator entropy, which signals the arrival to the asymptotic state, is not accompanied by the saturation of the error. The latter continues to grow in a power-law manner, see the dashed line in Fig. 3b. This means that MPO states – even with $R = 480$ – are still different from the genuine asymptotic state of the model (which is the zero-value eigenelement of the corresponding Lindbladian).

5. Conclusions

We presented a parallel implementation of the MPO-TEBD algorithm to propagate many-body open quantum systems. Parallelization is performed using the MPI technology and employs the master-worker scheme for computational tasks distribution. High-performance implementations of linear algebra from the Intel MKL were used to better utilize computational resources of modern hardware.

The performance tests on the Lobachevsky cluster demonstrated that 64 MPI processes running on four computational nodes is the optimal configuration for the model systems with $N = 128$ spins. As a next step, we plan to explore the possibility of further improvements of the parallelization by reducing the communications and increasing the efficiency of using computational resources. After that, we hope to reach the limit $N \simeq 400$ with the test-bed models.

6. Acknowledgments

The authors acknowledge support of the Russian Foundation for Basic Research and the Government of the Nizhni Novgorod region of the Russian Federation, grant # 18-41-520004. IV acknowledges support by the Institute for Basic Science, Project Code (IBS-R024-D1), and by the Korea University of Science and Technology Overseas Training program.

References

[1] J. Eisert, M. Friesdorf, and C. Gogolin, Nature Phys. **11**, 124 (2015).
[2] J. M. Gambetta, J. M. Chow, and M. Steffen, npj Quant. Inf. **3**, 2 (2017).
[3] Breuer, H.-P., and F. Petruccione, *The Theory of Open Quantum Systems* (Oxford University Press, Oxford, 2002).
[4] https://en.wikipedia.org/wiki/Titan (supercomputer)
[5] T. G. Kolda and B. W. Bader, SIAM Rev. **51**, 455 (2009).
[6] A. Cichocki, *Era of Big Data processing: a new approach via Tensor Networks and Tensor Decompositions*, https://arxiv.org/abs/1403.2048.
[7] I. V. Oseledets, SIAM J. Sci. Comput. **33**, 2295 (2011).
[8] U. Schollwoeck, Ann. of Phys. **326**(1) (2010).
[9] J. Haegeman, C. Lubich, I. Oseledets, B. Vandereycken, and F. Verstraete, Phys. Rev. B **94**, 165116 (2016).
[10] G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003).
[11] M. Zwolak and G. Vidal, Phys. Rev. Lett. **93**, 207205 (2004).
[12] M. Žnidarič, A. Scardicchio, and V. K. Varma, Phys. Rev. Lett. **117**, 040601 (2016).
[13] I. Vakulchyk, I. Yusipov, M. Ivanchenko, S. Flach, and S. Denisov, Phys. Rev. B **98**, 020202(R).
[14] G. Vidal, Phys. Rev. Lett. **91**(14) (2003)
[15] T. Prosen and I. Pižorn, Phys. Rev. A. **76**, 032316 (2007).