The Journal of Open Source Software

mpnum: A matrix product representation library for Python

Daniel Suess[1] **and Milan Holzäpfel**[2]

1 University of Cologne **2** Ulm University

DOI: 10.21105/joss.00465

Software

- Review
- Repository
- Archive

Submitted: 25 October 2017
Published: 15 December 2017

Licence
Authors of JOSS papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC-BY).

Summary

Tensors – or high-dimensional arrays – are ubiquitous in science and provide the foundation for numerous numerical algorithms in scientific computing, machine learning, signal processing, and other fields. With their high demands in memory and computational time, tensor computations constitute the bottleneck of many such algorithms. This has led to the development of sparse and low-rank tensor decompositions (Kolda and Bader 2009). One such decomposition, which was first developed under the name *"matrix product state"* (MPS) in the study of entanglement in quantum physics (Fannes, Nachtergaele, and Werner 1992), is the *matrix product* or *tensor train* (TT) representation (Schollwöck 2011; Oseledets 2011).

The matrix product tensor format is often used in practice (Latorre 2005; Savostyanov et al. 2014; Szalay et al. 2015; Zhang et al. 2015; Novikov et al. 2015; Stoudenmire and Schwab 2016) for two reasons: On the one hand, it captures the low-dimensional structure of many problems well. Therefore, it can be used model those problems computationally in an efficient way. On the other hand, the matrix product tensor format also allows for performing crucial tensor operations – such as addition, contraction, or low-rank approximation – efficiently (Schollwöck 2011; Oseledets 2011; Orús 2014; Bridgeman and Chubb 2017).

The library **mpnum** (Suess and Holzäpfel 2017) provides a flexible, user-friendly, and expandable toolbox for prototyping algorithms based on the matrix-product tensor format. Its fundamental data structure is the `MPArray` which represents a tensor with an arbitrary number of dimensions and local structure. Based on the `MPArray`, **mpnum** implements basic linear algebraic operations such as addition, contraction, approximate eigenvalue computation, etc. as well as specialized matrix-product decomposition operations such as compression or canonicalization. With these facilities, the user can express algorithms in high-level, readable code. Examples from quantum physics include matrix-product state (MPS) and matrix-product operator (MPO) computations, DMRG, low-rank tensor recovery, and efficient quantum state estimation.

Acknowledgements

The work of DS has been supported by the Excellence Initiative of the German Federal and State Governments (Grants ZUK 81), the ARO under contract W911NF-14-1-0098 (Quantum Characterization, Verification, and Validation), and the DFG projects GRO 4334/1,2 (SPP1798 CoSIP) The work of MH has been supported by an Alexander von Humboldt Professorship, the ERC Synergy grant BioQ, the EU projects QUCHIP, and the US Army Research Office Grant No. W911NF-14-1-0133. We furthermore acknowledge the provision of computational resources by the Regional Computing Center of the

University of Cologne (RRZK) through the DFG-funded HPC system CHEOPS, the state of Baden-Württemberg through bwHPC, and the German Research Foundation (DFG) through grant no INST 40/467-1 FUGG.

References

Bridgeman, J. C., and C. T. Chubb. 2017. "Hand-Waving and Interpretive Dance: An Introductory Course on Tensor Networks." *Journal of Physics A: Mathematical and Theoretical* 50 (22):223001. https://doi.org/10.1088/1751-8121/aa6dc3.

Fannes, M., B. Nachtergaele, and R. F. Werner. 1992. "Finitely Correlated States on Quantum Spin Chains." *Communications in Mathematical Physics* 144 (3):443–90. https://doi.org/10.1007/BF02099178.

Kolda, T., and B. Bader. 2009. "Tensor Decompositions and Applications." *SIAM Review* 51 (3):455–500. https://doi.org/10.1137/07070111X.

Latorre, J.I. 2005. "Image Compression and Entanglement." *arXiv Preprint Quant-Ph/0510031*. https://arxiv.org/abs/quant-ph/0510031.

Novikov, A., D. Podoprikhin, A. Osokin, and D. P. Vetrov. 2015. "Tensorizing Neural Networks." In *Advances in Neural Information Processing Systems 28*, edited by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, 442–50. Curran Associates, Inc. http://papers.nips.cc/paper/5787-tensorizing-neural-networks.pdf.

Orús, R. 2014. "A Practical Introduction to Tensor Networks: Matrix Product States and Projected Entangled Pair States." *Annals of Physics* 349:117–58. https://doi.org/10.1016/j.aop.2014.06.013.

Oseledets, I. 2011. "Tensor-Train Decomposition." *SIAM Journal on Scientific Computing* 33 (5):2295–2317. https://doi.org/10.1137/090752286.

Savostyanov, D.V., S. V. Dolgov, J. M. Werner, and I. Kuprov. 2014. "Exact Nmr Simulation of Protein-Size Spin Systems Using Tensor Train Formalism." *Phys. Rev. B* 90 (8). American Physical Society:085139. https://doi.org/10.1103/PhysRevB.90.085139.

Schollwöck, U. 2011. "The Density-Matrix Renormalization Group in the Age of Matrix Product States." *Annals of Physics*, January 2011 special issue, 326 (1):96–192. https://doi.org/10.1016/j.aop.2010.09.012.

Stoudenmire, E., and D. J. Schwab. 2016. "Supervised Learning with Tensor Networks." In *Advances in Neural Information Processing Systems 29*, 4799. Curran Associates, Inc. https://papers.nips.cc/paper/6211-supervised-learning-with-tensor-networks.

Suess, D., and M. Holzäpfel. 2017. *mpnum: Matrix Product Representation Library for Python*. https://github.com/dseuss/mpnum.

Szalay, S., M. Pfeffer, V. Murg, G. Barcza, F. Verstraete, R. Schneider, and Ö. Legeza. 2015. "Tensor Product Methods and Entanglement Optimization for Ab Initio Quantum Chemistry." *International Journal of Quantum Chemistry* 115 (19):1342–91. https://doi.org/10.1002/qua.24898.

Zhang, Z., X. Yang, I. Oseledets, G.E. Karniadakis, and L. Daniel. 2015. "Enabling High-Dimensional Hierarchical Uncertainty Quantification by Anova and Tensor-Train Decomposition." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 34 (1). IEEE:63–76. https://doi.org/10.1109/TCAD.2014.2369505.