



HAL
open science

The software landscape for the density matrix renormalization group

Per Sehlstedt, Jan Brandejs, Paolo Bientinesi, Lars Karlsson

► To cite this version:

Per Sehlstedt, Jan Brandejs, Paolo Bientinesi, Lars Karlsson. The software landscape for the density matrix renormalization group. *Computer Physics Communications*, 2026, 324, pp.110136. <10.1016/j.cpc.2026.110136>. <hal-05614255>

HAL Id: hal-05614255

<https://hal.science/hal-05614255v1>

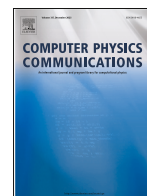
Submitted on 6 May 2026

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License



Computational Physics

The software landscape for the density matrix renormalization group

Per Sehlstedt ^{a,*}, Jan Brandejs ^b, Paolo Bientinesi ^a, Lars Karlsson ^a^a Department of Computing Science, Umeå University, Umeå, 901 87, Sweden^b Laboratoire de Chimie et Physique Quantiques, CNRS, Toulouse, 310 62, France

ARTICLE INFO

Editor: Prof. Blum Volker

Keywords:
DMRG
Survey
Modularity

ABSTRACT

The density matrix renormalization group (DMRG) algorithm is a cornerstone computational method for studying quantum many-body systems, renowned for its accuracy and adaptability. Because DMRG provides a general framework applicable across various fields such as materials science, quantum chemistry, and quantum computing, one might expect a shared, flexible library to serve most users. Nevertheless, numerous independent implementations continue to appear, resulting in significant duplication of effort. To identify collaboration opportunities that can promote a more unified approach, we map the rapidly expanding DMRG software landscape and provide a comprehensive comparison of features across 37 existing packages. When comparing key features, such as parallelism strategies for high-performance computing and symmetry-adapted formulations that enhance efficiency, we found significant overlap among the packages. This overlap suggests opportunities for collaboration to modularize common functionality—e.g., tensor operations, symmetry representations, and eigensolvers—as the packages are mostly independent and share few third-party library dependencies. More collaboration on modularization could reduce duplication of effort, improve interoperability, and enable prioritization and quicker spread of new advances. We believe the current lack of modularity is more socially driven than a technical issue; hence, we see raising awareness about the existing implementations as a first step in the right direction. Ultimately, this work emphasizes the value of greater cohesion through modularity, which would benefit DMRG software and related tensor-network-centered software, enabling the solution of more complex and ambitious problems.

1. Introduction

Modern scientific discovery across an increasingly diverse range of fields depends on numerical software. This survey examines the diverse and rapidly expanding software landscape for the density matrix renormalization group (DMRG). We provide a map of existing packages and their functionalities that can serve as a valuable resource for researchers and developers interested in the topic. To allow everyone to benefit from the full potential of the algorithm and its various extensions, we encourage greater community collaboration to build a more unified foundation based on a well-structured modular hierarchy.

Tensor network algorithms have become indispensable for computational science and engineering, providing powerful frameworks for efficiently representing and simulating quantum many-body systems [1–4]. One of the most successful examples is the modern tensor-based DMRG formulation [5–8]. DMRG's accuracy and versatility have established it as a cornerstone method, inspiring numerous adaptations and extensions far beyond its original scope [5,9,10]. These advancements have been applied to address problems across various research fields, includ-

ing quantum chemistry [11], materials science [12], quantum computing [13], and machine learning [14–16], among others, covering both one and higher dimensions [17].

Various DMRG flavors, characterized by their sweeping variational optimization, remain relevant and widely used today. Furthermore, the growing interest in simulating larger and more complex systems, coupled with the algorithm's significant time and space complexity, induces a need to embrace parallelism and high-performance computing (HPC) platforms [18–20]. Additionally, high-performing DMRG must exploit inherent symmetries in the systems, as symmetries introduce a block-sparsity structure that eliminates redundant computations, accelerates convergence, and enhances numerical stability and accuracy [9,21]. Symmetry-adapted DMRG is therefore another key component enabling simulations of previously intractable systems.

DMRG's broad applicability has driven the development of numerous software packages. However, researchers often develop these packages independently, motivated by their needs and tailoring them to specific applications. This application-focused and uncoordinated approach has resulted in a lack of standardized modularity. Additionally, it has led

* Corresponding author.

E-mail address: pers@cs.umu.se (P. Sehlstedt).<https://doi.org/10.1016/j.cpc.2026.110136>

Received 14 June 2025; Received in revised form 13 March 2026; Accepted 16 March 2026

Available online 20 March 2026

0010-4655/© 2026 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

to significant duplication of efforts when multiple groups implement similar functionalities. The limited modularity, in particular, poses significant challenges when adapting these codes to HPC platforms.

To identify opportunities for collaboration that can promote a more unified and structured approach—one that reduces duplication, enhances modularity, and promotes advancements that leverage HPC platforms for increasingly ambitious challenges in quantum many-body physics and beyond—we map the current DMRG software landscape to highlight similarities and differences across key features. Through this comparative perspective, we aim to reveal patterns that cut across implementations and point toward a shared foundation for future development.

The remainder of this paper is structured as follows: We briefly introduce the DMRG algorithm in Section 2. Since we focus on software, we assume the reader is reasonably familiar with the DMRG method. Many excellent reviews are available for those seeking a thorough introduction [5,9,10,22,23]. We also highlight some developments since its inception, as these have significantly contributed to the proliferation of software packages. We describe our survey’s method in Section 3. We present all the packages we identified in Section 4 and compare their features in Section 5. Finally, we discuss and conclude in Section 6.

2. Foundations and advancements

2.1. The DMRG algorithm

The DMRG algorithm [24,25] is a numerical method for approximating the dominant eigenvector of a matrix H , particularly the ground state of a quantum many-body system where H represents the system’s Hamiltonian. The matrix is often large enough to present significant computational challenges, as the Hamiltonian is defined on a Hilbert space which grows exponentially with the system’s size. However, the ground state of most quantum many-body systems obeys a boundary law (also known as area law), possibly with a multiplicative logarithmic correction [26–31]. Roughly speaking, the boundary law states that a subsystem’s entanglement entropy scales with the size of its boundary rather than its volume. This implies that valid ground state candidates are confined to a relatively tiny subspace of the Hilbert space. Ultimately, this allows for data-sparse representations that make computations tractable despite the exponential size of the Hilbert space.

The exact ground state of a Hamiltonian minimizes the energy expectation value E . Thus, by the variational principle, DMRG can be framed as a variational optimization [32], directly seeking to minimize E over the set of states spanned by a variational ansatz $|\psi\rangle$:

$$\arg \min_{|\psi\rangle} E(|\psi\rangle) = \arg \min_{|\psi\rangle} \frac{\langle \psi | H | \psi \rangle}{\langle \psi | \psi \rangle}. \quad (1)$$

Minimization is achieved through an iterative sweeping procedure that updates some elements while keeping all others fixed. The update at each sweeping step refines the approximation using an iterative eigensolver such as Lanczos [33] or Davidson [34]. These projection-based solvers are preferred as they allow for matrix-free implementations that avoid the simultaneous storage of Hamiltonian components in memory.

Originally, DMRG relied on reduced density matrices to identify and track the crucial degrees of freedom defining the system, hence the algorithm’s name. However, it was later acknowledged that the singular value decomposition could achieve a similar goal and that a natural choice for the variational ansatz takes the form of a matrix product state (MPS) [5]:

$$|\psi_{\text{MPS}}\rangle = \sum_{\sigma_1, \dots, \sigma_L} A_1^{\sigma_1} A_2^{\sigma_2} \dots A_L^{\sigma_L} |\sigma_1 \dots \sigma_L\rangle, \quad (2)$$

where L is the number of sites and each $A_i^{\sigma_i}$ is a matrix of parameters. Single-site space σ_i is typically of small dimension, e.g., a chemical orbital of dimension 4.

Typical options for DMRG calculations include distinctions based on system type (infinite versus finite) and boundary conditions (open,

periodic [32,35], smooth [36,37], or infinite [38]) tailored to a range of lattice geometries from one-dimensional chains to two-dimensional square [39,40], triangular [41,42], CAVO [43], honeycomb [44], or Kagome [45–47] lattices [17].

Enhancements such as density matrix perturbation [48], subspace expansion [49–52], and the recent controlled bond expansion [53] have notably enabled reliable single-site formulations of DMRG, sometimes delivering significant speed-ups and improved convergence, especially for systems with long-range interactions. DMRG is typically implemented using a two-site formulation, which naturally allows for adaptive precision refinements of $|\psi\rangle$, often helping to avoid convergence issues. Traditional single-site formulations significantly reduce computation and memory costs but do not allow for such adaptivity, putting the simulation at risk of getting stuck in metastable configurations, i.e., local minima that differ from the ground state [48]. These enhancements aim to balance the computational advantages of the single-site approach while reintroducing a degree of adaptivity and mitigating its risks.

Implementations commonly have the capability to exploit symmetries inherent to many quantum models. Symmetry-adapted DMRG can yield significant performance improvements [9,21]. Incorporating abelian symmetries enables block-sparse representations that reduce computational overhead by eliminating redundant calculations. Furthermore, leveraging non-abelian symmetries can result in even more significant efficiency gains by capturing a richer system structure. However, this entails a significantly increased implementation effort due to the complexity of managing non-commutative group representations, often relying on Clebsch-Gordan coefficients and the Wigner-Eckart theorem to construct and manipulate symmetry-adapted tensor structures properly and efficiently. Despite these challenges, the potential performance improvements offered by non-abelian symmetry-adapted methods make them a powerful tool for tackling complex, highly symmetric systems.

2.2. Development and evolution

Since its inception in 1992 by Steven R. White [24,25], the DMRG algorithm has continuously developed and evolved to accommodate new applications and system types. It was initially conceived within the framework of renormalization group (RG) methods, building upon earlier ideas in condensed matter physics. However, over time, the language describing DMRG shifted from the RG perspective to tensor networks, particularly matrix product states (MPS) [5–8] and matrix product operators (MPO) [54–58]. The MPS/MPO framework has since become the foundation for understanding and generalizing DMRG.

The applicability of DMRG has spread across several scientific domains. For example, in ab initio quantum chemistry, it has been applied to solve the electronic structure problem in molecules [59–63]. The algorithm’s ability to capture static correlations makes it highly effective, particularly in cases where traditional methods fall short. However, unlike traditional methods, DMRG often struggles to capture dynamic correlations. These trade-offs have resulted in hybrid quantum chemistry methods integrating DMRG with traditional multi-reference techniques. For instance, DMRG has been integrated with CASSCF, CASPT2, MRCI, CT, and MRDSRG to combine the methods’ strengths and effectively address static and dynamic electron correlations.

DMRG has also been extended to tackle dynamic behavior and properties [64,65]. Time-dependent DMRG methods, including time-evolving block-decimation (TEBD) [66–69], the time-dependent variational principle (TDVP) [70–73], local Krylov [74,75] and global Krylov [76,77] techniques, and MPO $W^{1,1}$ methods [78], have opened pathways for real- and imaginary-time evolution studies. Additionally, DMRG has been extended for excited-state calculations [79,80], finite-temperature simulations [54,81,82], and non-equilibrium systems [83]. Furthermore, it has inspired multigrid DMRG [84] for systems with

multiple length scales and the variational uniform MPS (VUMPS) algorithm [85] for systems approaching the thermodynamic limit.

In parallel, the conceptual framework of tensor network states has evolved to become recognized as a natural language to describe quantum states of matter [1–3,26,86,87]. The so-called bond dimension parameterizes tensor network states and controls the accuracy, with higher values yielding higher accuracy. In principle, MPS can represent any quantum state provided that the bond dimension χ is large enough; however, capturing the entanglement structure of higher-dimensional or critical systems requires exponentially large bond dimensions, leading to a rapid increase in computational cost. The reason for the exponential scaling is that MPS only satisfies a one-dimensional boundary law, meaning entanglement saturates to a constant for a fixed χ as the size of the system increases, making them most effective when representing systems generally corresponding to ground states of local one-dimensional gapped Hamiltonians [88]. This limit explains MPS's challenge with higher-dimensional systems, which often obey higher-dimensional boundary laws, and critical systems, which often obey boundary laws with a multiplicative logarithmic correction.

Alternative tensor network states have been developed to overcome MPS's inherent limitations, mainly to capture a higher scaling of entanglement entropy effectively. Among these are projected entangled-pair states/operators (PEPS/PEPO) [89], which can be seen as higher-dimensional generalizations of MPS/MPO. Moreover, there are the multi-scale entanglement renormalization ansatz (MERA) [90,91] and branching MERA [92,93]. Another alternative is the tree tensor network states/operators (TTNS/TTNO) [94,95]. Similar to MPS, TTNS, on average, satisfies a one-dimensional boundary law and is also inherently one-dimensional since it is loop-free (lacking cycles). Still, the hierarchical structure offers different benefits to MPS.

However, while these alternatives can effectively capture a higher scaling of entanglement entropy using lower bond dimensions, optimizing them can in total require more work due to increased algorithmic complexity. For reference, the typical computational cost for DMRG scales as $O(\chi_{\text{MPS}}^3)$ [17]. TTNS typically scales as $O(\chi_{\text{TTNS}}^4)$ [94,95]. MERA methods, depending on their structure, exhibit computational costs ranging from $O(\chi_{\text{MERA}}^7)$ to $O(\chi_{\text{MERA}}^{16})$, with higher costs associated with two-dimensional and branching variants [3,96]. Similarly, PEPS will have a typical computational effort that scales as $O(\chi_{\text{PEPS}}^{10})$ to $O(\chi_{\text{PEPS}}^{12})$ [17]. This stark contrast highlights a trade-off between low bond dimension and high algorithmic complexity.

3. Methodology

The survey was performed in two steps. First, we searched to identify all packages that implement DMRG, and then filtered out a relevant subset for further investigation (see Section 4 for the results). Second, we studied the selected packages to classify their features (see Section 5 for the results).

To identify all packages that implement DMRG, we surveyed relevant literature, particularly software release articles, searched online sources and code hosts (e.g., GitHub), and traced citations and related projects. The Tensor Network website [97] was a particularly useful starting point. We believe our coverage of open source packages is relatively complete. However, a recurring issue was the lack of transparency in articles regarding the software used for the DMRG calculations. This leads us to believe that there could potentially be a large number of packages, especially private ones, that we are not aware of.

To classify the package features, we gathered information from documentation, research articles, and other publicly accessible sources. We then approached the developers with questions regarding the features, dependencies, and parallelization strategies of their software to verify and complement the initial information. The input from the developers proved invaluable, as the veracity and recency of publicly available information are often lacking. Despite our best efforts, errors and omissions may still occur in the presented information. The errors could be

due to evolving software features, differing interpretations, or a lack of information.

We maintain updated tables on GitHub [98] and encourage feedback from package developers and users to help keep the information complete and up-to-date.

4. Packages implementing DMRG

The following are brief descriptions of the packages we found and compared in Section 5, including relevant references describing their implementations.

- ALPS (Algorithms and Libraries for Physics Simulations) [99–102] is a software package aiming to provide standardized components for numerical simulations of condensed matter systems. It has received contributions from numerous researchers worldwide. It contains two separate DMRG packages: ALPS DMRG [99,100] and ALPS MPS [100,103].
- BAGEL (Brilliantly Advanced General Electronic-structure Library) [104–106] is an electronic structure package maintained by the company QSimulate [107,108]. In its DMRG implementation, sites are individual molecules that collect multiple orbitals [109].
- BLOCK2 [110–114] aims to provide a comprehensive set of DMRG algorithms for use in electronic structure methods and other applications. The package is developed and maintained by the Garnet Chan group at Caltech and the Initiative for Computational Catalysis at the Flatiron Institute. It can interface with other quantum chemistry packages to enable hybrid DMRG methods, including PYSCF [115,116], OPENMOLCAS [117], and FORTE [118]. Finally, it facilitates transformations of MPS and/or MPO objects from various packages.
- CHEMPS2 [60,119–123] is a library containing a DMRG implementation for ab initio quantum chemistry primarily developed by Sebastian Wouters. It can interface with quantum chemistry packages that can handle R(O)HF calculations and molecular orbital matrix elements, and this has been done for PSI4 [124] and PYSCF [115,116].
- CHEMTENSOR [125] is a package for tensor network algorithms centered around chemical systems developed by the Quantum Computing Group at the Technical University of Munich [126,127].
- Chen et al. [128,129] presents a hybrid parallel implementation of DMRG with custom subroutines for vector operations on the GPU for improved performance. More recently, they also presented a novel approach to real-space parallel DMRG [130], building upon the work of Stoudenmire and White [131].
- CYTNX [132,133] is a tensor network library designed for classical and quantum physics simulations. Kai-Hsin Wu is the creator and manages the core team that develops and maintains the package. They also see significant contributions from individuals outside the team. The package has interfaces similar to popular libraries like NumPy [134], SciPy [135], and PyTorch [136].
- DMRG-BUDAPEST [137] is a package for general quantum many-body problems. Örs Legeza leads the Strongly Correlated Systems “Lendület” Research Group [138] that develops and maintains the code. It features various novel in-house optimization techniques to improve performance [139–144].
- DMRG++ [145–150] implements the DMRG algorithm with an emphasis on generic programming and minimal software dependencies. The main developer is Gonzalo Alvarez. Additionally, the package has a plug-in to extend its capabilities to GPUs [151,152].
- DMRGPY [153] is a library to simulate quasi-one-dimensional spin chains and fermionic systems, mainly developed by Jose Lado. The library relies on ITENSOR [154–156] (either the Julia or the C++ version).
- FOCUS [157] implements an ab initio DMRG algorithm. Zhendong Li originally introduced the package with a pilot implementation of a DMRG-like sweep algorithm for variational optimization of comb

- tensor network states [158]. It supports both non-relativistic and relativistic Hamiltonians, utilizing time-reversal symmetry in the latter case to reduce computational costs [159].
- Hong et al. [160] presents a DMRG implementation using various optimization strategies, specifically targeting GPUs with tensor cores.
 - ITENSOR [154–156] is a package for programming tensor network calculations, allowing users to focus on the connectivity of a tensor network without manually tracking indices. Matt Fishman and Miles Stoudenmire are the primary developers and receive contributions from collaborators. ITENSOR was initially implemented in C++ but later fully ported to Julia, with most new features first being developed there. The C++ version [161] includes DMRG in the main library, while the Julia version provides DMRG through the ITENSORMPS.JL [162] extension of ITENSORS.JL [163].
 - KYLIN [164–166] is an ab initio quantum chemistry software package for estimating the electronic structures of molecular systems. The package is developed by the Ma Group [167] at Shandong University, and it is specially designed for calculations with large active spaces.
 - MOLMPS [168] is a parallel implementation of DMRG for quantum chemistry. The Libor Veis Research Group [169] at the Heyrovský Institute [170] develops and maintains the code. The parallel scheme is based on an in-house MPI global memory library and supports various hybrid electronic structure methods.
 - MPSKIT.JL [171,172] contains tensor network algorithms for one-dimensional quantum and two-dimensional statistical mechanics problems. It is developed under the QuantumKitHub organization [173], where many related software packages focusing on quantum many-body physics through tensor networks are publicly available, with contributions mainly from the Quantum Group at Ghent University [174]. It mainly builds upon TENSORKIT.JL [175,176], which provides functionality for generic symmetries, and its capabilities can be extended with packages like SUNREPRESENTATIONS.JL [177] and MPSKITMODELS.JL [178].
 - MPTOOLKIT (Matrix Product Toolkit) [179,180] is a package for creating and manipulating MPS. Ian McCulloch started the project around 2002, initially envisioning a “next-generation” DMRG code that integrates non-abelian symmetries and emphasizes a flexible, general approach to constructing Hamiltonian operators and measuring observables [21].
 - OSMPS (Open Source MPS) [181–185] is a collection of tensor network algorithms for simulating entangled one-dimensional many-body quantum systems, focusing on long-range interaction necessary for the molecular systems. The package is developed by Michael L. Wall, Daniel Jaschke, Matthew Jones, and the Carr Theoretical Physics Research Group [186]. The initial focus was to offer a broad set of methods for quantum simulators based on atomic, molecular, and optical physics architectures.
 - PYTENET [187,188] implements quantum tensor network operations and simulations structured around MPS and MPO classes. The Quantum Computing Group at the Technical University of Munich [126,127] develops the package. It acts as a facilitator of algorithmic experimentation.
 - PYTTN [189,190] is a package for working with generic TTNS to compute dynamical properties of quantum systems, primarily developed by Lachlan Lindoy. It also provides functionality for merging multiple physical modes into a single site, which can be useful when handling weakly correlated subsystems with strong correlations between the degrees of freedom within each subsystem.
 - QCMAQUIS [191–195] is a SCINE module [196,197] that builds upon the ALPS MPS [100,103] code and implements various DMRG-based algorithms. The Reiher Research Group at ETH Zurich [197,198] develops the package. Key features include vibrational [199,200], time-dependent [201,202], and nuclear-electron [203,204] DMRG, and it accommodates non-relativistic and relativistic electronic structure calculations [205].
 - QSPACE [206–209] is a tensor library designed to incorporate arbitrary combinations of abelian and continuous non-abelian symmetries, down to no symmetries at all. It can perform all tensor operations identically by incorporating Clebsch-Gordan tensors, easing development for users. It has a long-standing history, with Andreas Weichselbaum as the primary developer. It was recently made open source, with extensive documentation provided in the release article [208].
 - QUANTUM TEA (Quantum Tensor network Emulator Applications) [210–212] is a set of tensor network packages for quantum simulation, circuit emulation, and machine learning applications. The Quantum Information and Matter group at the University of Padova [213] develops the package. Supplementary libraries extend their capabilities with diverse tensor backends to enable computations on CPUs, GPUs, and TPUs [214].
 - QUIMB [215,216] is a library for quantum information and many-body calculations, focusing primarily on tensor networks, developed mainly by Johnnie Gray. While its DMRG routine does not currently support features commonly found in other packages, QUIMB is compatible with complementary libraries like COTENGRA [217,218], AUTORAY [219], and SYMMRAY [220] to support efficient tensor network contraction, and various backend array libraries supporting block-sparse, abelian-symmetric, and fermionic representations.
 - RENORMALIZER [65,221–224] is a tensor network package with a focus on electron-phonon quantum dynamics developed by the Zhigang Shuai group [225,226]. It implements an original MPO/TTNO construction algorithm that leverages bipartite graph theory to automate and optimize the selection of normal and complementary operators for custom Hamiltonians in the sum-of-products form [227, 228], a feature later adopted by several other packages.
 - SEEMPS2 [229–231] is the second iteration of the SELF-Explaining Matrix-Product-State library [232]. Juan José García Ripoll and Paula García Molina in the Quantum Information and Foundations Group at the Institute of Fundamental Physics, in the Spanish Research Council (CSIC) [233] develop the package. It aims to enable rapid prototyping of MPS and DMRG-inspired algorithms.
 - SUNDMRG.JL [234] is a DMRG implementation specializing in a full $SU(n)$ symmetry and is developed by Masahiko G. Yamada, James R. Garrison, and Ryan V. Mishmash.
 - SYMMPS [235,236] builds upon the Scientific Parallel Algorithms Library (SciPAL) [237] and is developed by Sebastian Paeckel and Thomas Kölnner. It features an original construction scheme for MPO representations of arbitrary $U(1)$ -symmetric operators whenever there is an expression of the local structure in terms of a finite-state machine [238].
 - SYTEN [239,240] aims to be a tensor network toolkit with standard MPS, binary TTNS, and infinite PEPS utilities. Claudius Hubig created the package, and Sebastian Paeckel currently maintains it. It has also benefited from significant contributions from many others, primarily members of the group of Uli Schollwöck at LMU Munich [241].
 - TENPY (Tensor Network Python) [242–244] is a library for simulating strongly correlated quantum systems with tensor networks. Johannes Hauschild and Jakob Unfried are the primary developers, and many others have made significant contributions. Their stated philosophy is to balance readability and usability for newcomers while providing powerful algorithms for experts.
 - TENSOR-TOOLS [245,246] is a tensor network library that builds upon the Cyclops Tensor Framework (CTF) [247]. Ryan Levy developed the parallel version, which builds upon the serial version created by Xiongjie Yu, both of which Bryan Clark guided as they were part of the Clark Research Group [248]. The package utilizes multiple ITENSOR [154–156] systems and design concepts to improve interoperability between the two.
 - TENSORTRACK [249,250] is a package implementing various elementary algorithms that arise in the context of tensor networks and

is developed by the Quantum Group at Ghent University [174]. The package supports generic symmetries, including symmetry groups with multiplicities, which can have either bosonic or fermionic braiding rules.

- **ULTRADMRG** [251] is a library for performing large-scale calculations using one-dimensional tensor network algorithms developed by Hao-Xin Wang. It uses TENSORTOOLKIT [252], also developed by Hao-Xin Wang, for tensor operations and is specifically designed to tackle two-dimensional strongly correlated electron systems.
- **xDMRG++** [253] is a package that includes various MPS algorithms for studying one-dimensional quantum spin chains. David Aceituno at KTH Condensed Matter Physics group developed the package.
- **YASTN** [254,255] (Yet Another Symmetric Tensor Network) is a package for differentiable linear algebra with block-sparse tensors, supporting abelian symmetries. Marek Rams and Juraj Hasik are the primary developers. The package mainly focuses on PEPS but also supports a range of MPS algorithms.

We also identified a few additional packages, which we decided to exclude from Section 5 for various reasons. We mention them here for completeness. The following appear to no longer be actively maintained and/or used: DMRG.X [256,257], MPS [258], MPS++ [259], MPSTOOLS [260], ORZ [261], PARALLELDMRG [131,262], PYDMRG [263], PYTEN [264], QC-MPO-DMRG [265], REGO [266], SNAKE DMRG [267,268], TENSORNETWORK [269,270], TNT [271], and ZMPO-DMRG [272]. Next are those that have been superseded by other packages: BLOCK [273] and STACKBLOCK [274] are precursors to BLOCK2 [110,114], GRACEQ/MPS2 [275] is a precursor to ULTRADMRG [251], and UNI10 [276,277] is a precursor to CYTNX [132,133]. SIMPLE DMRG [278], SOPHISTICATED-DMRG [279], Glen Evenbly's example codes [280], and G. Catarina's implementation [22] were designed primarily as tutorials rather than scalable research tools. Finally, ITENSORNETWORKS.JL [281] and TENPY v2 [244] are currently pre-releases and were therefore excluded.

5. Feature comparisons

This section presents and compares multiple features of the packages. Rather than serving as a qualitative assessment of their effectiveness, we emphasize that our classification focuses on the packages' functionality within the context of DMRG. Hence, a package with more features is not necessarily more versatile or higher-performing than others. Still, this does not hinder us from identifying common patterns that point toward opportunities for modularization and a shared foundation across implementations.

Additionally, packages may offer unique features and adaptations suited to specific research needs, sometimes extending well beyond DMRG. These may include other tensor-network algorithms, such as those mentioned in Section 2.2—e.g., time-dependent approaches (real and imaginary time), methods for targeting excited states, and finite-temperature schemes. While these extensions lie outside the scope of the present survey, which focuses on ground-state DMRG implementations, many of the concepts and practical considerations discussed here are directly relevant and transferable to these related methods.

5.1. High-level overview

We present some high-level aspects of each package in Table 1. We highlight the implementation language and scheme, support for symmetries, and HPC capabilities. The columns in Table 1 are as follows:

- **ID**: A unique identifier for cross-referencing purposes.
- **Name**: The official name of the package; if there is one.
- **Language**: The programming languages used for implementation and interfaces. Languages marked with ¹-superscripts are those primarily used as interfaces.

Table 1

High-level overview of the packages included in the survey. Note that support for multiple features does not imply that they can be utilized simultaneously.

ID	Name	Language	OSS	Gen.	Sym.		HPC		
					A	NA	SM	DM	GPU
1	ALPS DMRG	C++, Python ¹	[102]	1 st	✓	–	*	–	–
2	ALPS MPS	C++, Python ¹	[102]	2 nd	✓	–	✓	–	–
3	BAGEL	C++	[106]	1 st	✓	–	✓	✓	–
4	Block2	C++, Python ¹	[114]	2 nd ‡	✓	✓	✓	✓	–
5	CheMPS2	C++, Python ¹	[123]	1 st	✓	✓	✓	✓	–
6	ChemTensor	C, Python ¹	[125]	2 nd †	✓	✓	✓	–	–
7	Chen et al.	C++	–	1 st	✓	–	✓	–	S
8	Cytnx	C++, Python ¹	[133]	2 nd †	✓	–	✓	–	S
9	DMRG-Budapest	C++, MATLAB ¹	–	2 nd ‡	✓	✓	✓	✓	M
10	DMRG++	C++	[150]	1 st	✓	–	✓	–	S
11	DMRGPy	Python	[153]	2 nd	–	–	*	–	–
12	FOCUS	C++	–	1 st	✓	✓	✓	✓	M
13	Hong et al.	C++	–	2 nd	–	–	✓	–	S
14	ITensor	C++	[161]	2 nd	✓	–	✓	–	–
15	ITensorMPS.jl	Julia	[162]	2 nd	✓	–	✓	–	S
16	Kylin	C++	–	2 nd	✓	✓	✓	–	–
17	MOLMPS	C++	–	1 st	✓	–	✓	✓	S
18	MPSKit.jl	Julia	[172]	2 nd	✓	✓	✓	–	–
19	MPToolkit	C++	[180]	2 nd	✓	✓	✓	–	–
20	OSMPS	Fortran, Python ¹	[185]	2 nd	✓	–	✓	✓	–
21	PyTeNet	Python	[188]	2 nd	✓	–	*	–	–
22	pyTTN	C++, Python ¹	[190]	2 nd †	–	–	✓	–	–
23	QCMAquis	C++, Python ¹	[195]	2 nd	✓	✓	✓	–	–
24	QSpace	C++, MATLAB ¹	[209]	2 nd ‡	✓	✓	✓	–	–
25	Quantum TEA	Fortran, Python ¹	[212]	2 nd †	✓	–	*	–	S
26	quimb	Python	[216]	2 nd	–	–	✓	–	–
27	Renormalizer	Python	[224]	2 nd †	✓	–	*	–	S
28	SeeMPS2	Python	[231]	2 nd	–	–	*	–	–
29	SUNDMRG.jl	Julia	[234]	1 st	–	–	✓	✓	S
30	SymMPS	C++	[236]	2 nd	✓	–	✓	–	–
31	SyTen	C++, Python ¹	–	2 nd	✓	✓	✓	✓	S
32	TenPy	Python	[244]	2 nd	✓	–	✓	✓	–
33	tensor-tools	C++	[246]	2 nd	✓	–	✓	✓	–
34	TensorTrack	MATLAB	[250]	2 nd	✓	✓	*	–	–
35	UltraDMRG	C++	[251]	2 nd	✓	–	✓	✓	S
36	xDMRG++	C++	[253]	2 nd	–	–	*	–	–
37	YASTN	Python	[255]	2 nd	✓	–	*	–	S

- **Open Source Software (OSS)**: Indicates whether the package is open source and available for download.
- **Implementation Formalism (Gen.)**: Specifies which view the DMRG implementation is based on: 1st for the traditional RG perspective using renormalized operators and 2nd for the tensor network perspective using MPO/MPS. Additionally, ‡-superscripts mark those who support both views, and †-superscripts mark those who support the usage of TTNO/TTNS.
- **Symmetry Support (Sym.)**: Indicates whether the package is symmetry-adapted to support abelian (A) and/or non-abelian (NA) symmetries or not.
- **High-Performance Computing (HPC)**: Indicates support for HPC platforms, categorized as shared-memory parallelism (SM), distributed-memory parallelism (DM), and single- (S) or multi- (M) GPU acceleration (GPU). We mark trivial support with *, see Section 5.3 for further details.

To provide further clarity regarding Table 1, firstly, most developers of packages listed as not open source are willing to share their work in collaborations on specific projects or may offer it for free to academics.

Secondly, DMRG expressed with tensor network operators/states is sometimes called a second-generation formalism and allows for more flexibility in finding the optimal state with a broader range of applications [192], hence the distinction. However, it is essential to note that both can be equally efficient, and to a certain extent, the difference between them is mainly semantic in the context of DMRG. In an expectation value optimization, the first-generation renormalized operators naturally correspond to partial traces of the second-generation MPO. In

DMRG, these are the only ones needed; the explicit MPO itself never needs to appear. However, for calculations of higher moments, for example, an MPO is demonstrably more accurate as it exactly represents the Hamiltonian, unlike the renormalized operators [5,62].

Thirdly, we classify packages based on whether they support symmetry in any form, without distinguishing between those limited to specific models and those that allow arbitrary symmetry combinations. Table 3 provides more fine-grained analysis.

Finally, similar to the symmetries, we classify packages based on whether they support HPC platforms in any form, regardless of their use. Table 2 provides a more detailed examination of the different strategies and optimization techniques.

We observe that most packages are implemented in C++, often exposing Python interfaces to enhance accessibility. Pure Python implementations are relatively rare, and only a few packages rely on languages such as Julia, Fortran, or MATLAB. Support for simpler symmetries and shared-memory parallelism is widespread, reflecting their maturity and ease of integration. In contrast, distributed-memory and GPU acceleration remain relatively rare, highlighting current development challenges in large-scale or heterogeneous computing.

5.2. Dependencies

All surveyed packages have external software dependencies. Some are mandatory while others are optional. Some packages are used in tandem with others that provide additional features, even though one is not technically dependent on the other. For example, the package MP-SKIT.JL can be used with the package MPSKITMODELS.JL, which provides concrete operators and models. For the purposes of this section, we also consider these relationships as dependencies.

We present the dependencies of each package in Fig. 1. Only those dependencies most relevant for DMRG calculations are included. For example, dependencies related to parsing and serialization were excluded. Thus, despite their prevalence, dependencies such as HDF5 are omitted. Similarly, Boost is not marked as a dependency when the documentation states that it is only used for parsing and serialization. We do not distinguish between required and optional dependencies. We also exclude edges between third-party libraries as long as this does not affect the set of dependencies reachable from any package.

Since all packages depend on some implementation of BLAS and LAPACK, we excluded them to reduce clutter. Furthermore, we excluded dependencies associated with MPI and threading (e.g., OpenMP). These dependencies are implicitly captured in Tables 1 and 2. We grouped CUDA-related software into a single node, covering bespoke CUDA kernels, tensor and other numerical libraries (cuTT, cuTENSOR, cuQUANTUM, cuBLAS, cuSOLVER), communication (NCCL), and core components (CUDA Runtime, CUDA Toolkit).

We conclude from this analysis that the packages are mostly independent, with little common functionality factored out into third-party libraries. The main exceptions are BLAS/LAPACK, which all packages rely on directly or indirectly, and the standard parallel support frameworks (e.g., OpenMP, MPI, CUDA). The Python-based packages commonly use SciPy and NumPy. Some libraries provide higher-level tensor operations, but they are mostly used by one or two packages.

5.3. Parallelism and mixed-precision

We now expand on the HPC aspects introduced in Section 5.1 to describe parallelization strategies and other optimization techniques. We closely follow the classification summary and five-level hierarchy of parallelism first introduced by Zhai and Chan [111] and then further discussed by Tian and Ma [18]. Table 2 presents the following:

- *Parallel strategies (Parallelism)*: Indicates support for the following types of parallelism:

- *Parallelism within matrix operations (i)*: The most fine-grained and lowest-level source of parallelism is the data parallelism found primarily within the matrix–matrix and matrix–vector multiplications that underpin the tensor algebra in the DMRG algorithm. We mark those who only support parallel matrix–matrix multiplications with *.
- *Parallelism over symmetry sectors (ii)*: The block-sparse tensor representations in symmetry-adapted DMRG implementations enable parallelism across symmetry sectors. The calculations on different blocks are independent, allowing them to run simultaneously [247,266].
- *Parallelism over normal and complementary operators (iii)*: In ab initio DMRG, the left-right decomposition of the Hamiltonian is expressed as a summation of products of normal and complementary operators. These operators are distributed over threads and processors, processing the corresponding calculations concurrently [60,282].
- *Parallelism over a sum of sub-Hamiltonians (iv)*: The Hamiltonian is rewritten as a sum of sub-Hamiltonians such that each term can be manipulated independently [62]. This coarse-grained, high-level parallelism is easily expressed in the MPO formalism and adds little communication overhead.
- *Parallelism over sites (v)*: The DMRG sequentially sweeps back and forth along a path. The path can be partitioned into sections that can be processed simultaneously at the expense of extra communication at the section boundaries [130,131]. This coarse-grained parallelism is especially useful for large systems with many sites.
- *Mixed-precision (MP)*: Indicates support for the recent mixed-precision optimization technique introduced by Tian et al. [283], where the initial sweeps are accelerated by using reduced floating-point precision. Later sweeps restore full precision by switching to full floating-point precision.

For a more comprehensive understanding of parallelism in matrix operations, the DMRG algorithm uses tensor data structures and related tensor algebra, including contraction, diagonalization, and decomposition. The sparse matrix–vector multiplication is the dominant operation of the diagonalization done by the iterative eigensolver and can be decomposed into several dense matrix–matrix multiplications, which can be computed in parallel [247,284]. Since all packages depend on BLAS/LAPACK in some way, simply linking to a multi-threaded or GPU-accelerated version will enable parallelization of dense matrix–matrix multiplications trivially.

Moreover, the matrices that form the operands for matrix–matrix multiplication calls are often relatively small in DMRG, particularly when exploiting symmetries, leading to non-negligible parallelization overhead and thus poor parallel efficiency [65,284]. The overhead can be somewhat mitigated by using batched matrix–matrix operations; however, we do not indicate such support here.

Regarding strategy *iii*, we observe it being highly prioritized by packages primarily related to quantum chemistry. The prioritization is likely due to the strategy being regarded as the largest source of parallelism for typical ab initio problems in quantum chemistry [18,111].

Regarding strategy *v*, Chen et al. [130] recently presented an improvement of the original scheme by Stoudenmire et al. [131], allowing section boundaries to be adaptively distributed. However, we do not distinguish which specific scheme is implemented in Table 2.

Regarding mixed-precision, the clean-up sweeps may use a mixed-precision diagonalization method, which leads to a two-level mixed-precision strategy [283]. However, we do not make this distinction in Table 2.

5.4. Symmetries

Next, we elaborate on the symmetry aspects introduced in Section 5.1. As before, we classify packages based on whether they

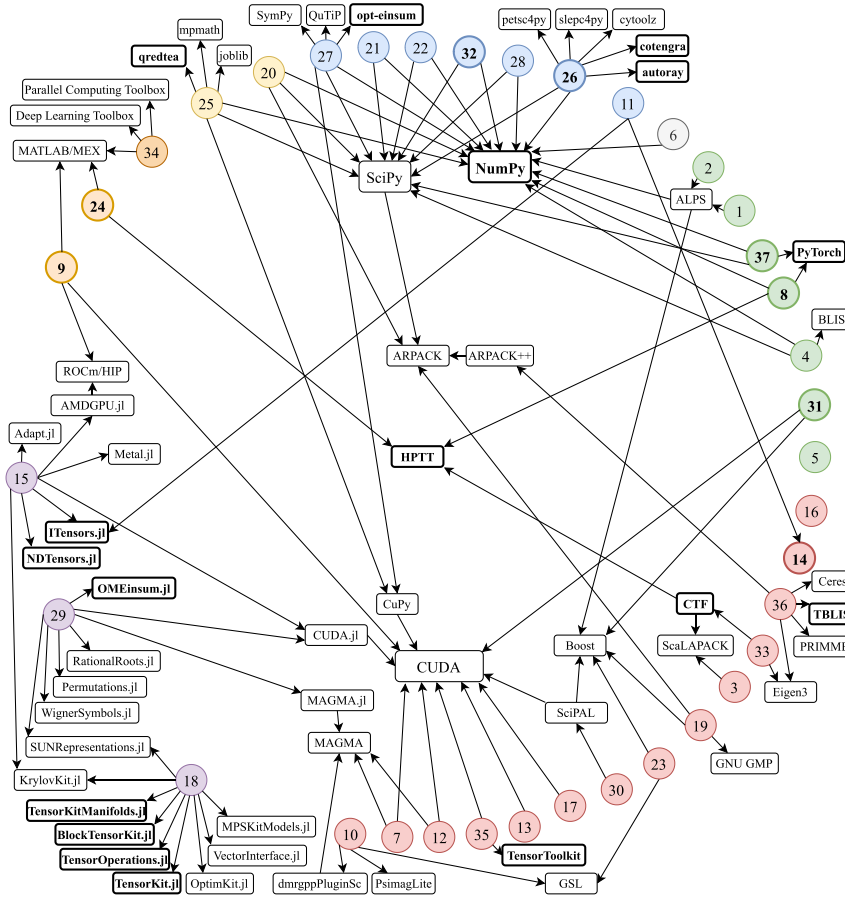


Fig. 1. Graph showing the required and optional dependencies, and compatible software relevant to DMRG calculations for the different packages. The circle numbers correspond to those in Table 1. The circles are color-coded based on the Language-column in Table 1. We have highlighted tensor-related libraries with a bold font and thicker outline.

support each symmetry in any form, without distinguishing between those whose support is limited to specific built-in models and those that allow them to be arbitrarily combined. Table 3 indicates support for the following symmetries:

- *The unitary group* $U(1)$: Indicates support for the unitary group of degree one, which is the most commonly used symmetry. It typically enforces the conservation of particle numbers (total electrons or separate spin counts) and total spin magnetization.
- *Cyclic groups* (\mathbb{Z}): Indicates support for either \mathbb{Z}_2 or more general \mathbb{Z}_n symmetries. A common usage for \mathbb{Z}_2 is to enforce the spin-flip (parity) symmetry in the transverse-field Ising model, where flipping all spins leaves the Hamiltonian unchanged. The more general \mathbb{Z}_n group is common in models with an n -fold rotational or cyclic invariance.
- *Special unitary groups* (SU): Indicates support for either $SU(2)$ or more generally $SU(n)$. The $SU(2)$ group is the most common and is often applied to preserve the total spin, often referred to as spin $SU(2)$. An example of $SU(n)$ is the channel $SU(3)$ symmetry considered by Weichselbaum [206].
- *Point groups* (P): Indicates support for abelian point group symmetries $P \in \{C_1, C_i, C_2, C_s, C_{2h}, D_2, C_{2v}, D_{2h}\}$ with real-valued character tables. They are often used for molecular systems to enforce spatial symmetries [60].
- *Fermion parity* (\mathbb{Z}_2^f): Indicates support for the fermion parity, which captures the anti-commuting nature of fermionic degrees of freedom. The parity is a \mathbb{Z}_2 quantum number that yields an additional sign when exchanging two fermions (each carrying odd parity).
- *Other*: Indicates support for less common symmetries, such as the special orthogonal group $SO(n)$, the symplectic group $Sp(2n)$, and

Anyonic symmetries [206]. Anyons are particles with non-trivial exchange statistics that are neither fermions nor bosons.

The fermion parity is often implemented using the Jordan-Wigner transformation, which maps fermionic creation and annihilation operators to (bosonic) spin operators. However, newer approaches incorporate it directly in tensor networks on arbitrary graphs using graded Hilbert spaces [285–287].

Finally, some software packages support user-defined symmetry groups, where the user specifies the properties/rules of the symmetry by providing appropriate Clebsch-Gordan coefficients or encoding them via an abstract tensor interface available in the package, for example. However, this type often requires complex workflows and a deep understanding of the software, as the interfaces tend to be complex. Additionally, it can be challenging to categorize and determine the threshold for what qualifies as supporting user-defined symmetries. Hence, we excluded them from Table 3 to maintain clarity and consistency.

We conclude that $U(1)$ and \mathbb{Z}_2^f are the most frequently supported forms of symmetries. Support for non-abelian symmetries is mainly restricted to $SU(2)$.

5.5. Hamiltonians

Formulating the Hamiltonian lies at the heart of applying DMRG. Here, we investigate the types of Hamiltonians available in the selected packages. We strive to provide broad insight into each package’s flexibility in defining system interactions. However, the classifications in this section are somewhat up for interpretation due to variations in perspective. Table 4 presents the following information:

Table 2

Support for parallelism strategies and mixed-precision optimization techniques. Note that support for multiple features does not imply that they can be utilized simultaneously.

ID	Name	HPC			Parallelism					MP
		SM	DM	GPU	i	ii	iii	iv	v	
1	ALPS DMRG	*	-	-	*	-	-	-	-	-
2	ALPS MPS	✓	-	-	*	✓	-	-	-	-
3	BAGEL	✓	✓	-	✓	-	-	-	-	-
4	Block2	✓	✓	-	✓	✓	✓	✓	✓	✓
5	CheMPS2	✓	✓	-	✓	✓	✓	-	-	-
6	ChemTensor	✓	-	-	✓	✓	-	✓	-	-
7	Chen et al.	✓	-	S	✓	✓	-	-	✓	-
8	Cytnx	✓	-	S	*	✓	✓	-	-	-
9	DMRG-Budapest	✓	✓	M	✓	✓	✓	✓	-	✓
10	DMRG++	✓	-	S	✓	✓	-	-	-	-
11	DMRGPy	*	-	-	*	✓	-	-	-	-
12	FOCUS	✓	✓	M	*	✓	✓	-	-	-
13	Hong et al.	✓	-	S	✓	-	-	-	-	-
14	ITensor	-	-	-	*	✓	-	-	-	-
15	ITensorMPS.jl	✓	-	S	*	✓	-	-	-	✓
16	Kylin	✓	-	-	✓	✓	✓	-	-	✓
17	MOLMPS	✓	✓	S	✓	✓	✓	-	-	-
18	MPSKit.jl	✓	-	-	*	✓	-	-	✓	-
19	MPToolkit	✓	-	-	*	✓	-	-	-	-
20	OSMPS	✓	✓	-	*	✓	-	-	-	-
21	PyTeNet	*	-	-	*	-	-	-	-	-
22	pyTTN	✓	-	-	*	-	-	✓	-	-
23	QCMAquis	✓	-	-	✓	✓	✓	✓	-	-
24	QSpace	✓	-	-	✓	✓	-	-	-	✓
25	Quantum TEA	*	-	S	*	-	-	-	-	✓
26	quimb	✓	-	-	✓	-	-	-	-	-
27	Renormalizer	*	-	S	*	-	-	-	-	-
28	SeeMPS2	*	-	-	*	-	-	-	-	-
29	SUNDMRG.jl	✓	✓	S	*	✓	-	-	-	-
30	SymMPS	✓	-	-	✓	✓	-	-	-	-
31	SyTen	✓	✓	S	✓	✓	-	✓	✓	-
32	TeNPy	✓	✓	-	✓	✓	-	✓	-	-
33	tensor-tools	✓	✓	-	✓	-	-	-	-	-
34	TensorTrack	*	-	-	*	-	-	-	-	-
35	UltraDMRG	✓	✓	S	*	✓	-	✓	-	-
36	xDMRG++	*	-	-	*	-	-	-	-	-
37	YASTN	*	-	S	*	-	-	-	-	-

- **Construction:** Indicates if and how a user can define custom Hamiltonians.
 - *Custom:* Indicates support for some generic custom Hamiltonian construction interface.
 - *Operator:* Indicates support for constructing custom Hamiltonians using local n -body operators defined on the different site-indices upon which the operators act.
- **Compression:** Indicates support for compression of the Hamiltonian, classified as lossless (L) and/or lossy, with lossy schemes specified with fixed (F) or user-defined (U) tolerance.
- **Built-in:** Indicates whether the package contains built-in Hamiltonians that target a specific or broad range of fields.

To provide further clarity regarding Table 4, we first note that we only indicate the presence of an interface for custom Hamiltonian construction and do not distinguish between implementations based on their specific details. Moreover, standard operators include spin, spin exchange, bosonic, and fermionic operators for the operator-type construction interfaces. However, we do not make distinctions based on the number or type of operators each package offers.

Secondly, we acknowledge that classifying the variety of built-in Hamiltonians is somewhat subjective. Currently, the column reflects the developers' views on their package. We provided them with the following examples of what we considered fields: quantum chemistry, condensed matter physics, and nuclear structure.

Regarding the compression, many packages allow the Hamiltonian to be initially defined as a lazy sum of operators or a symbolic expres-

Table 3

Support for various types of symmetries. Note that support for multiple features does not imply that they can be utilized simultaneously.

ID	Name	U(1)	Z	SU	P	Z ₂ ^f	Other
1	ALPS DMRG	✓	-	-	-	✓	-
2	ALPS MPS	✓	2	-	-	✓	-
3	BAGEL	✓	-	-	-	✓	-
4	Block2	✓	n	2	✓	✓	-
5	CheMPS2	✓	-	2	✓	✓	-
6	ChemTensor	✓	-	2	-	-	-
7	Chen et al.	✓	-	-	-	✓	-
8	Cytnx	✓	n	-	-	✓	-
9	DMRG-Budapest	✓	n	n	✓	✓	-
10	DMRG++	✓	-	-	-	✓	-
11	DMRGPy	-	-	-	-	✓	-
12	FOCUS	✓	-	2	-	✓	-
13	Hong et al.	-	-	-	-	-	-
14	ITensor	✓	n	-	-	✓	-
15	ITensorMPS.jl	✓	n	-	-	✓	-
16	Kylin	✓	-	2	✓	✓	-
17	MOLMPS	✓	-	-	✓	✓	-
18	MPSKit.jl	✓	n	n	-	✓	Anyonic/Categoric
19	MPToolkit	✓	n	2	-	✓	-
20	OSMPS	✓	2	-	-	✓	-
21	PyTeNet	✓	-	-	-	-	-
22	pyTTN	-	-	-	-	✓	-
23	QCMAquis	✓	2	2	✓	✓	-
24	QSpace	✓	n	n	-	✓	SO(n), Sp($2n$)
25	Quantum TEA	✓	n	-	-	✓	-
26	quimb	-	-	-	-	✓	-
27	Renormalizer	✓	-	-	-	✓	-
28	SeeMPS2	-	-	-	-	-	-
29	SUNDMRG.jl	-	-	n	-	-	-
30	SymMPS	✓	-	-	-	✓	-
31	SyTen	✓	n	2	-	✓	-
32	TeNPy	✓	n	-	-	✓	-
33	tensor-tools	✓	-	-	-	✓	-
34	TensorTrack	✓	n	n	-	✓	-
35	UltraDMRG	✓	n	-	-	✓	-
36	xDMRG++	-	-	-	-	✓	-
37	YASTN	✓	n	-	-	✓	-

sion, as indicated in the *Operator* column. Later, during the conversion to the final form, many of these packages can ensure that the representation is compressed—typically in a lossless manner—to minimize, for example, the bond dimension when constructing an MPO. However, many packages also allow further compression, either during or after the conversion, where users can control the tolerance.

Continuing with the conversion from initial to final representation of the Hamiltonian, the different developers reported using various schemes. The most commonly referenced works were those of Ren et al. [227] for MPO and Li et al. [228] for TTNO. Other works referenced were Crosswhite et al. [288], Wouters et al. [119], Szalay et al. [289], Keller et al. [192], Chan et al. [62], Paecel et al. [238], Hubig et al. [58], Chen et al. [290], Jaschke et al. [182,184], Parker et al. [291], Felser [292], and Çakır et al. [293].

5.6. Eigensolvers

Eigensolving is central to DMRG; here, we present various aspects of the packages' implementations and user interfaces. More specifically, Table 5 shows:

- **Eigensolver:** Indicates if the eigensolver is a bespoke implementation or provided by some external dependency. Some packages include both, often implementing their own sparse solver and complementing it with an external dense solver.
- **Method:** Primary method(s) used, mainly relevant for packages that implement their own eigensolver, as they can be less flexible regarding method changes. Common methods include: Lanczos (L), Arnoldi (A), Davidson (D), Jacobi–Davidson (JD), and Krylov–Schur (KS).

Table 4

Package support for custom Hamiltonian construction, operator-based definitions, compression, and the variety of fields for built-in models.

ID	Name	Construction		Compression	Built-in
		Custom	Operator		
1	ALPS DMRG	✓	✓	L, U	Specific
2	ALPS MPS	–	–	–	Specific
3	BAGEL	–	–	–	Specific
4	Block2	✓	✓	L, U	Broad
5	CheMPS2	–	–	L	Specific
6	ChemTensor	✓	✓	L	Broad
7	Chen et al.	–	–	–	Broad
8	Cytnx	✓	✓	U	–
9	DMRG-Budapest	✓	✓	L	Broad
10	DMRG + +	✓	–	–	Specific
11	DMRGPy	✓	✓	L, U	–
12	FOCUS	–	–	–	Specific
13	Hong et al.	–	–	–	Specific
14	ITensor	✓	✓	L, U	–
15	ITensorMPS.jl	✓	✓	L, U	–
16	Kylin	–	–	L	Specific
17	MOLMPS	✓	–	–	Broad
18	MPSKit.jl	✓	✓	L, U	Specific
19	MPToolkit	✓	–	–	Broad
20	OSMPS	✓	✓	L	–
21	PyTeNet	✓	✓	L	Broad
22	pyTTN	✓	✓	L	Broad
23	QCMAquis	–	–	L	Broad
24	QSpace	✓	✓	L	Specific
25	Quantum TEA	✓	✓	L	Specific
26	quimb	✓	✓	L, U	Specific
27	Renormalizer	✓	✓	L, U	Broad
28	SeeMPS2	✓	✓	–	–
29	SUNDMRG.jl	–	–	–	Specific
30	SymMPS	✓	–	L, U	–
31	SyTen	✓	–	L, U	–
32	TeNPy	✓	✓	L	Broad
33	tensor-tools	✓	✓	L, U	–
34	TensorTrack	✓	✓	–	Specific
35	UltraDMRG	✓	✓	L	–
36	xDMRG + +	✓	–	L, F	Specific
37	YASTN	✓	✓	L, U	–

- *Parameters:*

- *Access:* Specifies the level of user access to the eigensolver’s internal parameters.
- *Dynamic:* Indicates if the parameters can be updated dynamically, either through automatic-tuning (Auto) or user control (User).

Typical options for parameter access generally refer to how users can control solver-specific settings such as convergence thresholds, maximum subspace dimensions, restart frequencies, or preconditioner choices. Some packages also allow users to define how solver parameters evolve dynamically during the iterative process. For example, the convergence tolerance can be decreased depending on the iteration count. However, few include any form of automated tuning. As a result, users often bear the responsibility of selecting appropriate parameter values through manual experimentation, which can be highly system-dependent and thus time-consuming.

We observe that most packages include custom eigensolver implementations, with the majority based on either the Lanczos or Davidson algorithms. However, while things may look similar at the surface, we emphasize that there is still considerable room for variation among implementations. For instance, some implementations may incorporate alternative restart strategies—implicit or thick restarts—to balance subspace growth against convergence speed. Others may differ in their orthogonalization schemes—using full, partial, or selective reorthogonalization—to control numerical stability and computational overhead. Additional nuances include choices in how residual vectors are updated, whether preconditioning is applied within the subspace iterations, or how loss of orthogonality is monitored and corrected. Col-

Table 5

Eigensolvers used by each package, with implementation source, method, and parameter accessibility.

ID	Name	Eigensolver	Method	Parameters	
				Access	Dynamic
1	ALPS DMRG	Own	L	Most	Auto
2	ALPS MPS	IETL in ALPS	JD	Most	Auto
3	BAGEL	Own	D	All	–
4	Block2	Own	D	All	User
5	CheMPS2	Own	D	Most	User
6	ChemTensor	Own	L	Most	–
7	Chen et al.	Own	D	All	–
8	Cytnx	Own	L	Most	Auto
9	DMRG-Budapest	Own, ARPACK	L, D, etc.	All	Auto
10	DMRG + +	Own	L	Most	Auto, User
11	DMRGPy	ITensor C++/Julia	–	Most	User
12	FOCUS	Own	D	Most	User
13	Hong et al.	Own	L	All	–
14	ITensor	Own	D	All	–
15	ITensorMPS.jl	KrylovKit.jl	L, A	All	–
16	Kylin	Own	L, A	Some	User
17	MOLMPS	Own	D	Most	–
18	MPSKit.jl	KrylovKit.jl	L, A	All	Auto, User
19	MPToolkit	ARPACK	L, D	All	User
20	OSMPS	Own, ARPACK	L	Some	–
21	PyTeNet	Own	L	Most	–
22	pyTTN	Own	A	All	User
23	QCMAquis	ALPS	JD	All	User
24	QSpace	Own	L, A	Most	User
25	Quantum TEA	Own, SciPy	L	All	User
26	quimb	SciPy, slepc4py	KS, D, etc.	Most	User
27	Renormalizer	Own, SciPy	D	Most	–
28	SeeMPS2	SciPy	L	Some	–
29	SUNDMRG.jl	Own	L	Some	–
30	SymMPS	Own	L	All	User
31	SyTen	Own	L, D	All	User
32	TeNPy	Own, SciPy	L	All	User
33	tensor-tools	Own	D	All	User
34	TensorTrack	Own	KS	Some	–
35	UltraDMRG	Own	L	Most	–
36	xDMRG + +	PRIMME	A, D, etc.	Most	Auto, User
37	YASTN	Own	L	All	User

lectively, these design decisions can lead to notable differences in convergence behavior, memory footprint, and overall solver efficiency, underscoring the critical importance of modular architectures. By designing with modular components in mind, we gain the flexibility to adapt the solver to a wide range of problems, experiment with optimizations, and incorporate future algorithmic improvements without requiring an overhaul of the entire implementation.

5.7. Configurations

Here, we present typical configuration options available in the selected packages. Table 6 presents the following information:

- *System types (Sys.):* Indicates support for finite (F) and infinite (I) system types.
- *Boundary conditions (BC):* Indicates support for different types of boundary conditions, such as open (O) and periodic (P).
- *Update types (Site):* Indicates support for local optimization schemes used at each step, such as single-site (1) and two-site (2).
- *Enhancements:* Indicates support for different types of enhancements, including density matrix perturbation (DMP) [48], subspace expansion (SSE) schemes [49], and dynamically extended active space (DEAS) procedure [294].
- *Out-of-core computations (OoC):* Indicates support for out-of-core computations, such as writing edge tensors to disk when they become too large to all be held in memory.
- *Observable (Obs):* Indicates whether users can measure observables other than energy during the calculation.

Table 6
Typical configuration options for each package.

ID	Name	Sys.	BC	Site	Enhancements	OoC	Obs
1	ALPS DMRG	F	O, P	2	DMP	✓	✓
2	ALPS MPS	F	O, P	1, 2	DMP	✓	✓
3	BAGEL	F	O	1	DMP	–	–
4	Block2	F	O, P	1, 2	DMP, SSE	✓	✓
5	CheMPS2	F	O	2	DMP	✓	✓
6	ChemTensor	F	O	1, 2	–	–	✓
7	Chen et al.	F	O	2	–	–	–
8	Cytnx	F, I	O, P	1, 2	–	–	✓
9	DMRG-Budapest	F, I	O, P	1, 2	DMP, SSE, DEAS	✓	✓
10	DMRG++	F	O, P	1, 2	DMP, SSE	✓	✓
11	DMRGPy	F	O	2	DMP	–	–
12	FOCUS	F	O	1, 2	DMP	✓	✓
13	Hong et al.	F	O	1	–	–	–
14	ITensor	F, I	O, P	2	DMP	✓	✓
15	ITensorMPS.jl	F, I	O, P	2	DMP	✓	✓
16	Kylin	F	O, P	1, 2	SSE	✓	–
17	MOLMPS	F, I	O, P	1, 2	–	✓	✓
18	MPSKit.jl	F, I	O, P	1, 2	SSE	–	✓
19	MPToolkit	F, I	O, P	1, 2	DMP	✓	✓
20	OSMPS	F, I	O	1, 2	–	–	–
21	PyTeNet	F	O	1, 2	–	–	✓
22	pyTTN	F	O	1	SSE	–	✓
23	QCMAquis	F	O	2	DMP	✓	✓
24	QSpace	F	O, P	1, 2	–	✓	✓
25	Quantum TEA	F	O, P	1, 2	SSE	✓	–
26	quimb	F	O, P	1, 2	–	–	✓
27	Renormalizer	F	O, P	1, 2	SSE	–	–
28	SeeMPS2	F	O	2	–	–	✓
29	SUNDMRG.jl	F	O	2	DMP	–	–
30	SymMPS	F	O, P	1, 2	–	–	✓
31	SyTen	F	O, P	1, 2	SSE	✓	✓
32	TeNPy	F, I	O, P	1, 2	DMP, SSE	✓	✓
33	tensor-tools	F	O, P	2	–	–	–
34	TensorTrack	F, I	O	1, 2	SSE	–	–
35	UltraDMRG	F	O	1, 2	SSE	✓	–
36	xDMRG++	F, I	O	1, 2	SSE	–	✓
37	YASTN	F	O	1, 2	–	–	✓

Note that the *Site* column in Table 6 refers strictly to the local update type and does not denote physical orbital sites, blocks of the reduced density matrix, or partitioning used in some quantum-chemistry DMRG implementations.

Observables refer to quantities that can be measured during a calculation, allowing the user to monitor the system’s state or compute physical properties. In typical implementations, this feature enables the user to specify local or global operators, such as spin, particle number, or correlation functions, to be measured after each sweep or optimization step. Depending on the package, observables may be evaluated by supplying a callback function, by overloading a function with a predefined signature, or by using a generator-style interface that yields access to the current state at each step. This approach provides flexibility, allowing custom measurements to be incorporated into the workflow without requiring modifications to the core algorithm. It can be beneficial for tracking convergence, computing expectation values, or performing intermediate analyses during long-running calculations.

6. Discussion

We systematically surveyed the software landscape for the DMRG algorithm and identified more than 50 packages, 37 of which we compared across multiple features. We revealed significant overlap among the supported features and showed that, despite the overlap, most developments are largely independent, each relying on its own implementation of core components. This independence suggests opportunities for collaboration to establish standards for modular components, thereby enabling increased use of shared software infrastructure.

Independently developed core components face several challenges. The current lack of standardization limits interoperability, making mi-

gration between packages costly and time-consuming. Moreover, new techniques must be re-implemented in each package, thereby slowing adoption and limiting the immediate spread of its benefits. Furthermore, implementations tailored to specific use cases may be rigid if they lack modularity, potentially making them difficult to adapt to new methods or technologies. These challenges become particularly acute when addressing complex features, such as GPU support, multi-level parallelism, non-abelian symmetries, and auto-tuning, which is likely why we do not observe any package implementing all of them.

Increased use of shared software infrastructure can concentrate maintenance efforts, reduce duplication, and improve long-term stability, interoperability, and developer productivity. Additionally, it can help new advances and features spread more quickly. For example, all packages rely on BLAS rather than re-implementing matrix multiplication, so improvements to BLAS immediately accelerate all packages. Just as BLAS provides a foundation for matrix operations, a standardized, modular approach could be established for other core components, including tensor operations, symmetry representations, and eigensolvers. Initial efforts in this area have already started, such as through the Tensor Algebra Processing Primitives (TAPP) initiative [295].

The current landscape, with an abundance of largely independent packages and significant functional overlap, thus raises two questions: Why has this proliferation emerged, and why has it not been accompanied by greater standardization and modularity of core components to promote shared software infrastructure?

Proliferation. We find part of an explanation for the abundance of packages in Section 4, with different packages serving distinct purposes: some are research-oriented, developed to prototype and explore novel ideas; others are application-focused, specializing in specific real-world problems; and still others are commercially oriented, for enterprise use. While this distinction helps explain why a single all-purpose package is neither realistic nor necessarily desirable, it does not fully account for the recurrent independence or the limited modularity.

A further explanation is that most packages were created within a research group by a small team to address their particular needs, and what may have begun as a modest, internally developed proof-of-concept gradually evolved over many years into a sophisticated package eventually released as open source. Having complete control over the implementation can allow the developers to tailor it more easily to their needs, which can be advantageous when studying a specific application.

Standardization. As for the lack of standardization and modularity in core components, one reason may be the complexity of some components. Tensor network algorithms as a whole are still in a relatively exploratory stage, undergoing rapid development, and it is not yet fully understood how to handle, for example, symmetries in the most efficient way, which is central to DMRG. Still, this exploratory nature can underscore the importance of standardization and modularity. Flexible, robust interfaces for modular components can provide a stable foundation, enabling researchers to develop, test, and compare new ideas—such as improved symmetry handling—more efficiently without rewriting entire codebases, thereby potentially accelerating exploration.

Another reason is likely a lack of awareness of parallel developments, which occur across various fields. Information about existing DMRG implementations is scattered across journals, publications, code repositories, and informal documentation, making it difficult to obtain a clear overview of available approaches or to build directly on prior work. Therefore, we view this survey as a step toward consolidating knowledge, increasing the visibility of existing solutions, and fostering more coordinated progress.

Conclusions. Taken together, our survey suggests that the diversity of DMRG implementations is driven more by social than technical factors. By mapping the landscape, we have shown that many packages share overlapping features, revealing opportunities for modularization

and shared software infrastructure. Building on this insight, we encourage broader collaboration across projects and disciplines, combining bottom-up modularization of shared functionality with top-down consolidation of similar projects. Such collaboration would benefit DMRG and related tensor-network-centered software, allowing these powerful algorithms to tackle increasingly complex and ambitious problems.

Recent efforts indicate a growing recognition of the benefits of modularization and collaborative development, as seen in the modular designs of ITENSORS.JL [163], TENSORKIT.JL [176], and QUIMB [216], and the collaborative effort of CYTNX [133] and TENPY [244] to create CYTEN [296]—a shared underlying platform that combines and extends the linear algebra of symmetric tensors from both packages. Moreover, we observe efforts to enhance interoperability, for example, BLOCK2 [114], which facilitates the transformation of MPS and/or MPO objects from various packages. Finally, several ongoing efforts specifically aim to bring together developers of DMRG and related software. Examples include the Reusable Libraries in Quantum Chemistry meeting [297,298] and the Toulouse Tensor Workshop [295,299].

CRedit authorship contribution statement

Per Sehlstedt: Writing – original draft, Visualization, Investigation, Conceptualization; **Jan Brandejs:** Writing – review & editing, Investigation, Conceptualization; **Paolo Bientinesi:** Writing – review & editing, Supervision, Conceptualization; **Lars Karlsson:** Writing – review & editing, Supervision, Conceptualization.

Data availability

No data was used for the research described in the article.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We thank the developers of the surveyed DMRG packages for taking the time to answer our questions, as an essential part of this work has been based on the information they provided.

Supported by the eSENCE Programme under the Swedish Government's Strategic Research Initiative.

J. Brandejs has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 101019907).

References

- [1] R. Orús, A practical introduction to tensor networks: matrix product states and projected entangled pair states, *Ann. Phys.* (2014) 117–158. <https://doi.org/10.1016/j.aop.2014.06.013>
- [2] R. Orús, Advances on tensor network theory: symmetries, fermions, entanglement, and holography, *Eur. Phys. J. B* 87 (11) (2014). <https://doi.org/10.1140/epjb/e2014-50502-9>
- [3] R. Orús, Tensor networks for complex quantum systems, *Nature Rev. Phys.* 1 (9) (2019) 538–550. <https://doi.org/10.1038/s42254-019-0086-7>
- [4] K. Okunishi, T. Nishino, H. Ueda, Developments in the tensor network — from statistical mechanics to quantum entanglement, *J. Phys. Soc. Jpn.* 91 (6) (2022). <https://doi.org/10.7566/jpsj.91.062001>
- [5] U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, *Ann. Phys.* 326 (1) (2011) 96–192. <https://doi.org/10.1016/j.aop.2010.09.012>
- [6] S. Östlund, S. Rommer, Thermodynamic limit of density matrix renormalization, *Phys. Rev. Lett.* 75 (19) (1995) 3537–3540. <https://doi.org/10.1103/physrevlett.75.3537>
- [7] S. Rommer, S. Östlund, Class of ansatz wave functions for one-dimensional spin systems and their relation to the density matrix renormalization group, *Phys. Rev. B* 55 (4) (1997) 2164–2181. <https://doi.org/10.1103/physrevb.55.2164>

- [8] J. Dukelsky, M.A. Martín-Delgado, T. Nishino, G. Sierra, Equivalence of the variational matrix product method and the density matrix renormalization group applied to spin chains, *Europhys. Lett.* 43 (4) (1998) 457–462. <https://doi.org/10.1209/epl/11998-00381-x>
- [9] U. Schollwöck, The density-matrix renormalization group, *Rev. Mod. Phys.* 77 (1) (2005) 259–315. <https://doi.org/10.1103/revmodphys.77.259>
- [10] K.A. Hallberg, New trends in density matrix renormalization, *Adv. Phys.* 55 (5–6) (2006) 477–526. <https://doi.org/10.1080/00018730600766432>
- [11] S.R. White, R.L. Martin, Ab initio quantum chemistry using the density matrix renormalization group, *J. Chem. Phys.* 110 (9) (1999) 4127–4130. <https://doi.org/10.1063/1.478295>
- [12] B. Bauer, S. Bravyi, M. Motta, G.K.-L. Chan, Quantum algorithms for quantum chemistry and quantum materials science, *Chem. Rev.* 120 (22) (2020) 12685–12717. <https://doi.org/10.1021/acs.chemrev.9b00829>
- [13] T. Ayrál, T. Louvet, Y. Zhou, C. Lambert, E.M. Stoudenmire, X. Waintal, Density-matrix renormalization group algorithm for simulating quantum circuits with a finite fidelity, *PRX Quantum* 4 (2) (2023). <https://doi.org/10.1103/prxquantum.4.020304>
- [14] E. Stoudenmire, D.J. Schwab, Supervised learning with tensor networks, in: D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, 29, Curran Associates, Inc., 2016, pp. 4799–4807.
- [15] A. Novikov, M. Trofimov, I. Oseledets, Exponential machines, 2016, <https://doi.org/10.48550/ARXIV.1605.03795>
- [16] W. Huggins, P. Patil, B. Mitchell, K.B. Whaley, E.M. Stoudenmire, Towards quantum machine learning with tensor networks, *Quantum Sci. Technol.* 4 (2) (2019) 024001. <https://doi.org/10.1088/2058-9565/aaea94>
- [17] E.M. Stoudenmire, S.R. White, Studying two-dimensional systems with the density matrix renormalization group, *Annu. Rev. Condens. Matter Phys.* 3 (1) (2012) 111–128. <https://doi.org/10.1146/annurev-conmatphys-020911-125018>
- [18] Y. Tian, H. Ma, High-performance computing for density matrix renormalization group, *Current Chinese Sci.* 3 (3) (2023) 178–186. <https://doi.org/10.2174/2210298103666221125162959>
- [19] M. Ganahl, J. Beall, M. Hauru, A.G.M. Lewis, T. Wojno, J.H. Yoo, Y. Zou, G. Vidal, Density matrix renormalization group with tensor processing units, *PRX Quantum* 4 (1) (2023). <https://doi.org/10.1103/prxquantum.4.010317>
- [20] C. Brower, S.R. Bernabeu, J. Hammond, J. Gunnels, S.S. Xanthea, M. Ganahl, A. Menczer, O. Legeza, Mixed-precision ab initio tensor network state methods adapted for NVIDIA Blackwell technology via emulated FP64 arithmetic, 2025. <https://doi.org/10.48550/ARXIV.2510.04795>
- [21] I.P. McCulloch, M. Gulácsi, The non-Abelian density matrix renormalization group algorithm, *Europhys. Lett.* 57 (6) (2002) 852–858. <https://doi.org/10.1209/epl/i2002-00393-0>
- [22] G. Catarina, B. Murta, Density-matrix renormalization group: a pedagogical introduction, *Eur. Phys. J. B* 96 (8) (2023). <https://doi.org/10.1140/epjb/s10051-023-00575-2>
- [23] M.A. Werner, A. Menczer, O. Legeza, Tensor network state methods and quantum information theory for strongly correlated molecular systems, 2025. <https://doi.org/10.48550/ARXIV.2501.18263>
- [24] S.R. White, Density matrix formulation for quantum renormalization groups, *Phys. Rev. Lett.* 69 (19) (1992) 2863–2866. <https://doi.org/10.1103/physrevlett.69.2863>
- [25] S.R. White, Density-matrix algorithms for quantum renormalization groups, *Phys. Rev. B* 48 (14) (1993) 10345–10356. <https://doi.org/10.1103/physrevb.48.10345>
- [26] G. Evenbly, G. Vidal, Scaling of entanglement entropy in the (branching) multiscale entanglement renormalization ansatz, *Phys. Rev. B* 89 (23) (2014). <https://doi.org/10.1103/physrevb.89.235113>
- [27] M. Srednicki, Entropy and area, *Phys. Rev. Lett.* 71 (5) (1993) 666–669. <https://doi.org/10.1103/physrevlett.71.666>
- [28] J. Eisert, T.J. Osborne, General entanglement scaling laws from time evolution, *Phys. Rev. Lett.* 97 (15) (2006). <https://doi.org/10.1103/physrevlett.97.150404>
- [29] M.B. Hastings, An area law for one-dimensional quantum systems, *J. Stat. Mech: Theory Exp.* 2007 (08) (2007) P08024. <https://doi.org/10.1088/1742-5468/2007/08/p08024>
- [30] L. Masanes, Area law for the entropy of low-energy states, *Phys. Rev. A* 80 (5) (2009). <https://doi.org/10.1103/physreva.80.052104>
- [31] J. Eisert, M. Cramer, M.B. Plenio, Colloquium: area laws for the entanglement entropy, *Rev. Mod. Phys.* 82 (1) (2010) 277–306. <https://doi.org/10.1103/revmodphys.82.277>
- [32] F. Verstraete, D. Porras, J.I. Cirac, Density matrix renormalization group and periodic boundary conditions: a quantum information perspective, *Phys. Rev. Lett.* 93 (22) (2004). <https://doi.org/10.1103/physrevlett.93.227205>
- [33] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, *J. Res. Natl. Bur. Stand.* 45 (4) (1950) 255. <https://doi.org/10.6028/jres.045.026>
- [34] E.R. Davidson, The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices, *J. Comput. Phys.* 17 (1) (1975) 87–94. [https://doi.org/10.1016/0021-9991\(75\)90065-0](https://doi.org/10.1016/0021-9991(75)90065-0)
- [35] P. Pippan, S.R. White, H.G. Evertz, Efficient matrix-product state method for periodic boundary conditions, *Phys. Rev. B* 81 (8) (2010). <https://doi.org/10.1103/physrevb.81.081103>
- [36] M. Vekić, S.R. White, Smooth boundary conditions for quantum lattice systems, *Phys. Rev. Lett.* 71 (26) (1993) 4283–4286. <https://doi.org/10.1103/physrevlett.71.4283>
- [37] M. Vekić, S.R. White, Hubbard model with smooth boundary conditions, *Phys. Rev. B* 53 (21) (1996) 14552–14557. <https://doi.org/10.1103/physrevb.53.14552>

- [38] H.N. Phien, G. Vidal, I.P. McCulloch, Infinite boundary conditions for matrix product state calculations, *Phys. Rev. B* 86 (24) (2012). <https://doi.org/10.1103/physrevb.86.245107>
- [39] E. Jeckelmann, S.R. White, Density-matrix renormalization-group study of the polaron problem in the Holstein model, *Phys. Rev. B* 57 (11) (1998) 6376–6385. <https://doi.org/10.1103/physrevb.57.6376>
- [40] S.R. White, A.L. Chernyshev, Néel order in square and triangular lattice heisenberg models, *Phys. Rev. Lett.* 99 (12) (2007). <https://doi.org/10.1103/physrevlett.99.127004>
- [41] S.-i. Yoshikawa, K. Okunishi, M. Senda, S. Miyashita, Quantum fluctuation-induced phase transition in $s=1/2$ XY-like heisenberg antiferromagnets on the triangular lattice, *J. Phys. Soc. Jpn.* 73 (7) (2004) 1798–1804. <https://doi.org/10.1143/jpsj.73.1798>
- [42] M.Q. Weng, D.N. Sheng, Z.Y. Weng, R.J. Bursill, Spin-liquid phase in an anisotropic triangular-lattice heisenberg model: exact diagonalization and density-matrix renormalization group calculations, *Phys. Rev. B* 74 (1) (2006). <https://doi.org/10.1103/physrevb.74.012407>
- [43] S.R. White, Spin gaps in a frustrated heisenberg model for CaV_2O_7 , *Phys. Rev. Lett.* 77 (17) (1996) 3633–3636. <https://doi.org/10.1103/physrevlett.77.3633>
- [44] H.-C. Jiang, Z.-C. Gu, X.-L. Qi, S. Trebst, Possible proximity of the mott insulating iridate Na_2IrO_3 to a topological phase: phase diagram of the heisenberg-Kitaev model in a magnetic field, *Phys. Rev. B* 83 (24) (2011). <https://doi.org/10.1103/physrevb.83.245104>
- [45] S. Sachdev, Kagomé – and triangular-lattice Heisenberg antiferromagnets: ordering from quantum fluctuations and quantum-disordered ground states with unconfined bosonic spinons, *Phys. Rev. B* 45 (21) (1992) 12377–12396. <https://doi.org/10.1103/physrevb.45.12377>
- [46] H.C. Jiang, Z.Y. Weng, D.N. Sheng, Density matrix renormalization group numerical study of the kagome antiferromagnet, *Phys. Rev. Lett.* 101 (11) (2008). <https://doi.org/10.1103/physrevlett.101.117203>
- [47] S. Yan, D.A. Huse, S.R. White, Spin-liquid ground state of the $s=1/2$ kagome heisenberg antiferromagnet, *Science* 332 (6034) (2011) 1173–1176. <https://doi.org/10.1126/science.1201080>
- [48] S.R. White, Density matrix renormalization group algorithms with a single center site, *Phys. Rev. B* 72 (18) (2005). <https://doi.org/10.1103/physrevb.72.180403>
- [49] C. Hubig, I.P. McCulloch, U. Schollwöck, F.A. Wolf, Strictly single-site DMRG algorithm with subspace expansion, *Phys. Rev. B* 91 (15) (2015). <https://doi.org/10.1103/physrevb.91.155115>
- [50] S.V. Dolgov, D.V. Savostyanov, Alternating minimal energy methods for linear systems in higher dimensions, *SIAM J. Sci. Comput.* 36 (5) (2014) A2248–A2271. <https://doi.org/10.1137/140953289>
- [51] S.V. Dolgov, D.V. Savostyanov, Corrected one-site density matrix renormalization group and alternating minimal energy algorithm, in: A. Abdulle, S. Deparis, D. Kressner, F. Nobile, M. Picasso (Eds.), *Numerical Mathematics and Advanced Applications - ENUMATH 2013*, Springer International Publishing, Cham, 2015, pp. 335–343. https://doi.org/10.1007/978-3-319-10705-9_33
- [52] M. Yang, S.R. White, Time-dependent variational principle with ancillary krylov subspace, *Phys. Rev. B* 102 (9) (2020). <https://doi.org/10.1103/physrevb.102.094315>
- [53] A. Gleis, J.-W. Li, J. von Delft, Controlled bond expansion for density matrix renormalization group ground state search at single-site costs, *Phys. Rev. Lett.* 130 (24) (2023). <https://doi.org/10.1103/physrevlett.130.246402>
- [54] F. Verstraete, J.J. Garcia-Ripoll, J.I. Cirac, Matrix product density operators: simulation of finite-temperature and dissipative systems, *Phys. Rev. Lett.* 93 (20) (2004). <https://doi.org/10.1103/physrevlett.93.207204>
- [55] M. Zwolek, G. Vidal, Mixed-State dynamics in one-Dimensional quantum lattice systems: a time-Dependent superoperator renormalization algorithm, *Phys. Rev. Lett.* 93 (20) (2004). <https://doi.org/10.1103/physrevlett.93.207205>
- [56] I.P. McCulloch, From density-matrix renormalization group to matrix product states, *J. Stat. Mech. Theory Exp.* 2007 (10) (2007) P10014. <https://doi.org/10.1088/1742-5468/2007/10/p10014>
- [57] B. Pirvu, V. Murg, J.I. Cirac, F. Verstraete, Matrix product operator representations, *New J. Phys.* 12 (2) (2010) 025012. <https://doi.org/10.1088/1367-2630/12/2/025012>
- [58] C. Hubig, I.P. McCulloch, U. Schollwöck, Generic construction of efficient matrix product operators, *Phys. Rev. B* 95 (3) (2017). <https://doi.org/10.1103/physrevb.95.035129>
- [59] G.K.-L. Chan, S. Sharma, The density matrix renormalization group in quantum chemistry, *Annu. Rev. Phys. Chem.* 62 (1) (2011) 465–481. <https://doi.org/10.1146/annurev-physchem-032210-103338>
- [60] S. Wouters, D. Van Neck, The density matrix renormalization group for ab initio quantum chemistry, *Eur. Phys. Journal. D* 68 (9) (2014). <https://doi.org/10.1140/epjd/e2014-50500-1>
- [61] T. Yanai, Y. Kurashige, W. Mizukami, J. Chalupský, T.N. Lan, M. Saitow, Density matrix renormalization group for ab initio calculations and associated dynamic correlation methods: a review of theory and applications, *Int. J. Quantum Chem.* 115 (5) (2014) 283–299. <https://doi.org/10.1002/qua.24808>
- [62] G.K.-L. Chan, A. Keselman, N. Nakatani, Z. Li, S.R. White, Matrix product operators, matrix product states, and ab initio density matrix renormalization group algorithms, *J. Chem. Phys.* 145 (1) (2016). <https://doi.org/10.1063/1.4955108>
- [63] A. Baiardi, M. Reiher, The density matrix renormalization group in chemistry and molecular physics: recent developments and new challenges, *J. Chem. Phys.* 152 (4) (2020). <https://doi.org/10.1063/1.5129672>
- [64] S. Paeckel, T. Köhler, A. Swoboda, S.R. Manmana, U. Schollwöck, C. Hubig, Time-evolution methods for matrix-product states, *Ann. Phys.* 411 (2019) 167998. <https://doi.org/10.1016/j.aop.2019.167998>
- [65] W. Li, J. Ren, Z. Shuai, Numerical assessment for accuracy and GPU acceleration of TD-DMRG time evolution schemes, *J. Chem. Phys.* 152 (2) (2020). <https://doi.org/10.1063/1.5135363>
- [66] A.J. Daley, C. Kollath, U. Schollwöck, G. Vidal, Time-dependent density-matrix renormalization-group using adaptive effective Hilbert spaces, *J. Stat. Mech. Theory Exp.* 2004 (04) (2004) P04005. <https://doi.org/10.1088/1742-5468/2004/04/p04005>
- [67] G. Vidal, Efficient classical simulation of slightly entangled quantum computations, *Phys. Rev. Lett.* 91 (14) (2003). <https://doi.org/10.1103/physrevlett.91.147902>
- [68] G. Vidal, Efficient simulation of one-dimensional quantum many-body systems, *Phys. Rev. Lett.* 93 (4) (2004). <https://doi.org/10.1103/physrevlett.93.040502>
- [69] S.R. White, A.E. Feiguin, Real-time evolution using the density matrix renormalization group, *Phys. Rev. Lett.* 93 (7) (2004). <https://doi.org/10.1103/physrevlett.93.076401>
- [70] J. Haegeman, J.I. Cirac, T.J. Osborne, I. Pižorn, H. Verschelde, F. Verstraete, Time-dependent variational principle for quantum lattices, *Phys. Rev. Lett.* 107 (7) (2011). <https://doi.org/10.1103/physrevlett.107.070601>
- [71] A. Milsted, J. Haegeman, T.J. Osborne, F. Verstraete, Variational matrix product ansatz for nonuniform dynamics in the thermodynamic limit, *Phys. Rev. B* 88 (15) (2013). <https://doi.org/10.1103/physrevb.88.155116>
- [72] J. Haegeman, C. Lubich, I. Oseledecs, B. Vandereycken, F. Verstraete, Unifying time evolution and optimization with matrix product states, *Phys. Rev. B* 94 (16) (2016). <https://doi.org/10.1103/physrevb.94.165116>
- [73] B. Kloss, Y.B. Lev, D. Reichman, Time-dependent variational principle in matrix-product state manifolds: pitfalls and potential, *Phys. Rev. B* 97 (2) (2018). <https://doi.org/10.1103/physrevb.97.024307>
- [74] P. Schmitteckert, Nonequilibrium electron transport using the density matrix renormalization group method, *Phys. Rev. B* 70 (12) (2004). <https://doi.org/10.1103/physrevb.70.121302>
- [75] A.E. Feiguin, S.R. White, Time-step targeting methods for real-time dynamics using the density matrix renormalization group, *Phys. Rev. B* 72 (2) (2005). <https://doi.org/10.1103/physrevb.72.020404>
- [76] J.J. Garcia-Ripoll, Time evolution of matrix product states, *New J. Phys.* 8 (12) (2006) 305. <https://doi.org/10.1088/1367-2630/8/12/305>
- [77] P.E. Dargel, A. Wöllert, A. Honecker, I.P. McCulloch, U. Schollwöck, T. Pruschke, Lanczos algorithm with matrix product states for dynamical correlation functions, *Phys. Rev. B* 85 (20) (2012). <https://doi.org/10.1103/physrevb.85.205119>
- [78] M.P. Zaletel, R.S.K. Mong, C. Karrasch, J.E. Moore, F. Pollmann, Time-evolving a matrix product state with long-ranged interactions, *Phys. Rev. B* 91 (16) (2015). <https://doi.org/10.1103/physrevb.91.165112>
- [79] M. Chandross, J.C. Hicks, Density-matrix renormalization-group method for excited states, *Phys. Rev. B* 59 (15) (1999) 9699–9702. <https://doi.org/10.1103/physrevb.59.9699>
- [80] V. Khemani, F. Pollmann, S. Sondhi, Obtaining highly excited eigenstates of many-body localized hamiltonians by the density matrix renormalization group approach, *Phys. Rev. Lett.* 116 (24) (2016). <https://doi.org/10.1103/physrevlett.116.247204>
- [81] R.J. Bursill, T. Xiang, G.A. Gehring, The density matrix renormalization group for a quantum spin chain at non-zero temperature, *J. Phys.: Condens. Matter* 8 (40) (1996) L583–L590. <https://doi.org/10.1088/0953-8984/8/40/003>
- [82] X. Wang, T. Xiang, Transfer-matrix density-matrix renormalization-group theory for thermodynamics of one-dimensional quantum systems, *Phys. Rev. B* 56 (9) (1997) 5061–5064. <https://doi.org/10.1103/physrevb.56.5061>
- [83] Y. Hieida, Application of the density matrix renormalization group method to a non-Equilibrium problem, *J. Phys. Soc. Jpn.* 67 (2) (1998) 369–372. <https://doi.org/10.1143/jpsj.67.369>
- [84] M. Dolfi, B. Bauer, M. Troyer, Z. Ristivojevic, Multigrid algorithms for tensor network states, *Phys. Rev. Lett.* 109 (2) (2012). <https://doi.org/10.1103/physrevlett.109.020604>
- [85] V. Zauner-Stauber, L. Vanderstraeten, M.T. Fishman, F. Verstraete, J. Haegeman, Variational optimization algorithms for uniform matrix product states, *Phys. Rev. B* 97 (4) (2018). <https://doi.org/10.1103/physrevb.97.045145>
- [86] G. Evenbly, G. Vidal, Tensor network states and geometry, *J. Stat. Phys.* 145 (4) (2011) 891–918. <https://doi.org/10.1007/s10955-011-0237-4>
- [87] M.C. Bañuls, Tensor network algorithms: a route map, *Annu. Rev. Condens. Matter Phys.* 14 (1) (2023) 173–191. <https://doi.org/10.1146/annurev-conmatphys-040721-022705>
- [88] M.B. Hastings, Solving gapped Hamiltonians locally, *Phys. Rev. B* 73 (8) (2006). <https://doi.org/10.1103/physrevb.73.085115>
- [89] F. Verstraete, J.I. Cirac, Renormalization algorithms for Quantum-Many Body Systems in two and higher dimensions, 2004, <https://doi.org/10.48550/ARXIV.COND-MAT/0407066>
- [90] G. Vidal, Entanglement renormalization, *Phys. Rev. Lett.* 99 (22) (2007). <https://doi.org/10.1103/physrevlett.99.220405>
- [91] G. Vidal, Class of quantum many-Body states that can be efficiently simulated, *Phys. Rev. Lett.* 101 (11) (2008). <https://doi.org/10.1103/physrevlett.101.110501>
- [92] G. Evenbly, G. Vidal, Real-space decoupling transformation for quantum many-Body systems, *Phys. Rev. Lett.* 112 (22) (2014). <https://doi.org/10.1103/physrevlett.112.220502>
- [93] G. Evenbly, G. Vidal, Class of highly entangled many-Body states that can be efficiently simulated, *Phys. Rev. Lett.* 112 (24) (2014). <https://doi.org/10.1103/physrevlett.112.240502>
- [94] Y.-Y. Shi, L.-M. Duan, G. Vidal, Classical simulation of quantum many-body systems with a tree tensor network, *Phys. Rev. A* 74 (2) (2006). <https://doi.org/10.1103/physreva.74.022320>

- [95] N. Nakatani, G.K.-L. Chan, Efficient tensor network states (TTNS) for quantum chemistry: generalizations of the density matrix renormalization group algorithm, *J. Chem. Phys.* 138 (13) (2013). <https://doi.org/10.1063/1.4798639>
- [96] G. Eevenly, G. Vidal, Entanglement renormalization in two spatial dimensions, *Phys. Rev. Lett.* 102 (18) (2009). <https://doi.org/10.1103/physrevlett.102.180406>
- [97] Tensor network website, 2025, <https://tensornetwork.org/software/>.
- [98] P. Sehlstedt, DMRG-software, 2025, <https://github.com/PerSehlstedt/DMRG-software>.
- [99] A.F. Albuquerque, F. Alet, P. Corboz, P. Dayal, A. Feiguin, S. Fuchs, L. Gamper, E. Gull, S. Gürtler, A. Honecker, R. Igarashi, M. Körner, A. Kozhevnikov, A. Läuchli, S.R. Manmana, M. Matsumoto, I.P. McCulloch, F. Michel, R.M. Noack, G. Pawłowski, L. Pollet, T. Pruschke, U. Schollwöck, S. Todo, S. Trebst, M. Troyer, P. Werner, S. Wessel, The ALPS project release 1.3: open-source software for strongly correlated systems, *J. Magn. Magn. Mater.* 310 (2) (2007) 1187–1193. <https://doi.org/10.1016/j.jmmm.2006.10.304>
- [100] B. Bauer, L.D. Carr, H.G. Evertz, A. Feiguin, J. Freire, S. Fuchs, L. Gamper, J. Gukelberger, E. Gull, S. Guertler, A. Hehn, R. Igarashi, S.V. Isakov, D. Koop, P.N. Ma, P. Mates, H. Matsuo, O. Parcollet, G. Pawłowski, J.D. Picon, L. Pollet, E. Santos, V.W. Scarola, U. Schollwöck, C. Silva, B. Surer, S. Todo, S. Trebst, M. Troyer, M.L. Wall, P. Werner, S. Wessel, The ALPS project release 2.0: open source software for strongly correlated systems, *J. Stat. Mech. Theory Exp.* 2011 (05) (2011) P05001. <https://doi.org/10.1088/1742-5468/2011/05/p05001>
- [101] ALPS - algorithms and libraries for physics simulations, 2025, <https://alps.comp-phys.org>.
- [102] ALPS, 2025, <https://github.com/ALPSim/ALPS>.
- [103] M. Dolfi, B. Bauer, S. Keller, A. Kosenkov, T. Ewart, A. Kantian, T. Giamarchi, M. Troyer, Matrix product state applications for the ALPS project, *Comput. Phys. Commun.* 185 (12) (2014) 3430–3440. <https://doi.org/10.1016/j.cpc.2014.08.019>
- [104] T. Shiozaki, BAGEL: brilliantly advanced general electronic-structure library, *WIREs Comput. Mol. Sci.* 8 (1) (2017). <https://doi.org/10.1002/wcms.1331>
- [105] BAGEL Brilliantly advanced general electronic-structure library, 2025, <http://www.nubakery.org>.
- [106] BAGEL, 2025, <https://github.com/qsimulate-open/bagel>.
- [107] QSimulate, 2025, <https://qsimulate.com/>.
- [108] QSimulate, sourced QSimulate programs for quantum chemistry, 2025, <https://github.com/qsimulate-open>.
- [109] S.M. Parker, T. Shiozaki, Communication: active space decomposition with multiple sites: density matrix renormalization group algorithm, *J. Chem. Phys.* 141 (21) (2014). <https://doi.org/10.1063/1.4902991>
- [110] H. Zhai, H.R. Larsson, S. Lee, Z.-H. Cui, T. Zhu, C. Sun, L. Peng, R. Peng, K. Liao, J. Tölle, J. Yang, S. Li, G.K.-L. Chan, Block2: a comprehensive open source framework to develop and apply state-of-the-art DMRG algorithms in electronic structure and beyond, *J. Chem. Phys.* 159 (23) (2023). <https://doi.org/10.1063/5.0180424>
- [111] H. Zhai, G.K.-L. Chan, Low communication high performance ab initio density matrix renormalization group algorithms, *J. Chem. Phys.* 154 (22) (2021). <https://doi.org/10.1063/5.005902>
- [112] H.R. Larsson, H. Zhai, K. Gunst, G.K.-L. Chan, Matrix product states with large sites, *J. Chem. Theory Comput.* 18 (2) (2022) 749–762. <https://doi.org/10.1021/acs.jctc.1c00957>
- [113] H. Zhai, G.K.-L. Chan, A comparison between the one- and two-step spin-orbit coupling approaches based on the ab initio density matrix renormalization group, *J. Chem. Phys.* 157 (16) (2022). <https://doi.org/10.1063/5.0107805>
- [114] Block2, 2025, <https://github.com/block-hczhai/block2-preview>.
- [115] Q. Sun, T.C. Berkelbach, N.S. Blunt, G.H. Booth, S. Guo, Z. Li, J. Liu, J.D. McClain, E.R. Sayfutyarova, S. Sharma, S. Wouters, G.K. Chan, PySCF: the python-based simulations of chemistry framework, *WIREs Comput. Mol. Sci.* 8 (1) (2017). <https://doi.org/10.1002/wcms.1340>
- [116] Q. Sun, X. Zhang, S. Banerjee, P. Bao, M. Barbry, N.S. Blunt, N.A. Bogdanov, G.H. Booth, J. Chen, Z.-H. Cui, J.J. Eriksen, Y. Gao, S. Guo, J. Hermann, M.R. Hermes, K. Koh, P. Koval, S. Lehtola, Z. Li, J. Liu, N. Mardirossian, J.D. McClain, M. Motta, B. Mussard, H.Q. Pham, A. Pulkin, W. Purwanto, P.J. Robinson, E. Ronca, E.R. Sayfutyarova, M. Scheurer, H.F. Schurkus, J.E.T. Smith, C. Sun, S.-N. Sun, S. Upadhyay, L.K. Wagner, X. Wang, A. White, J.D. Whitfield, M.J. Williamson, S. Wouters, J. Yang, J.M. Yu, T. Zhu, T.C. Berkelbach, S. Sharma, A.Y. Sokolov, G.K.-L. Chan, Recent developments in the PySCF program package, *J. Chem. Phys.* 153 (2) (2020). <https://doi.org/10.1063/5.0006074>
- [117] F. Aquilante, J. Autschbach, A. Baiardi, S. Battaglia, V.A. Borin, L.F. Chibotaru, I. Conti, L. De Vico, M. Delcey, F.G. Ignacio, N. Ferré, L. Freitag, M. Garavelli, X. Gong, S. Knecht, E.D. Larsson, R. Lindh, M. Lundberg, P.Å. Malmqvist, A. Nenov, J. Norell, M. Odellius, M. Olivucci, T.B. Pedersen, L. Pedraza-González, Q.M. Phung, K. Pierloot, M. Reiher, I. Schapiro, J. Segarra-Martí, F. Segatta, L. Seijo, S. Saunikk, D.-C. Sergentu, C.J. Stein, L. Ungur, M. Vacher, A. Valentini, V. Veryazov, Modern quantum chemistry with [Open]Molcas, *J. Chem. Phys.* 152 (21) (2020). <https://doi.org/10.1063/5.0004835>
- [118] F.A. Evangelista, C. Li, P. Verma, K.P. Hannon, J.B. Schriber, T. Zhang, C. Cai, S. Wang, N. He, N.H. Stair, M. Huang, R. Huang, J.P. Misiewicz, S. Li, K. Marin, Z. Zhao, L.A. Burns, Forte: a suite of advanced multireference quantum chemistry methods, *J. Chem. Phys.* 161 (6) (2024). <https://doi.org/10.1063/5.0216512>
- [119] S. Wouters, W. Poelmans, P.W. Ayers, D. Van Neck, CheMPS2: a free open-source spin-adapted implementation of the density matrix renormalization group for ab initio quantum chemistry, *Comput. Phys. Commun.* 185 (6) (2014) 1501–1514. <https://doi.org/10.1016/j.cpc.2014.01.019>
- [120] S. Wouters, T. Bogaerts, P. Van Der Voort, V. Van Speybroeck, D. Van Neck, Communication: DMRG-SCF study of the singlet, triplet, and quintet states of oxo-Mn(salen), *J. Chem. Phys.* 140 (24) (2014). <https://doi.org/10.1063/1.4885815>
- [121] S. Wouters, W. Poelmans, S. De Baerdemacker, P.W. Ayers, D. Van Neck, CheMPS2: improved DMRG-SCF routine and correlation functions, *Comput. Phys. Commun.* 191 (2015) 235–237. <https://doi.org/10.1016/j.cpc.2015.01.007>
- [122] S. Wouters, V. Van Speybroeck, D. Van Neck, DMRG-CASPT2 study of the longitudinal static second hyperpolarizability of all-trans polyenes, *J. Chem. Phys.* 145 (5) (2016). <https://doi.org/10.1063/1.4959817>
- [123] CheMPS2, 2025, <https://github.com/SebWouters/CheMPS2>.
- [124] D.G.A. Smith, L.A. Burns, A.C. Simmonett, R.M. Parrish, M.C. Schieber, R. Galvelis, P. Kraus, H. Kruse, R. Di Remigio, A. Alenaizan, A.M. James, S. Lehtola, J.P. Misiewicz, M. Scheurer, R.A. Shaw, J.B. Schriber, Y. Xie, Z.L. Glick, D.A. Sirianni, J.S. O'Brien, J.M. Waldrop, A. Kumar, E.G. Hohenstein, B.P. Pritchard, B.R. Brooks, H.F. Schaefer, A.Y. Sokolov, K. Patkowski, A.E. DePrince, U. Bozkaya, R.A. King, F.A. Evangelista, J.M. Turney, T.D. Crawford, C.D. Sherrill, P- scf -Si4, 1.4: open-source software for high-throughput quantum chemistry, *J. Chem. Phys.* 152 (18) (2020). <https://doi.org/10.1063/5.0006002>
- [125] ChemTensor, 2025, <https://github.com/qc-tum/chemtensor>.
- [126] Quantum computing group at TUM computer science dept, 2025, <https://www.cs.cit.tum.de/qc>.
- [127] QC-TUM, 2025, <https://github.com/qc-tum>.
- [128] F.-Z. Chen, C. Cheng, H.-G. Luo, Hybrid parallel optimization of density matrix renormalization group method, *Acta Physica Sinica* 68 (12) (2019) 120202. <https://doi.org/10.7498/aps.68.20190586>
- [129] F.-Z. Chen, C. Cheng, H.-G. Luo, Improved hybrid parallel strategy for density matrix renormalization group method*, *Chin. Phys. B* 29 (7) (2020) 070202. <https://doi.org/10.1088/1674-1056/ab8a42>
- [130] F.-Z. Chen, C. Cheng, H.-G. Luo, Real-space parallel density matrix renormalization group with adaptive boundaries, *Chin. Phys. B* 30 (8) (2021) 080202. <https://doi.org/10.1088/1674-1056/abeb08>
- [131] E.M. Stoudenmire, S.R. White, Real-space parallel density matrix renormalization group, *Phys. Rev. B* 87 (15) (2013). <https://doi.org/10.1103/physrevb.87.155137>
- [132] K.-H. Wu, C.-T. Lin, K. Hsu, H.-T. Hung, M. Schneider, C.-M. Chung, Y.-J. Kao, P. Chen, The cytnx library for tensor networks, *SciPost Phys. Codebases* (2025). <https://doi.org/10.21468/scipostphyscodeb.53>
- [133] Cytnx, 2025, <https://github.com/Cytnx-dev/Cytnx>.
- [134] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J.F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T.E. Oliphant, Array programming with numpy, *Nature* 585 (7825) (2020) 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- [135] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, Í. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: fundamental algorithms for scientific computing in python, *Nat. Methods* 17 (3) (2020) 261–272. <https://doi.org/10.1038/s41592-019-0686-2>
- [136] J. Ansel, E. Yang, H. He, N. Gimelshein, A. Jain, M. Voznesensky, B. Bao, P. Bell, D. Berard, E. Burovski, G. Chauhan, A. Chourdia, W. Constable, A. Desmaison, Z. DeVito, E. Ellison, W. Feng, J. Gong, M. Gschwind, B. Hirsh, S. Huang, K. Kalambarak, L. Kirsch, M. Lazos, M. Lezcano, Y. Liang, Y. Lu, C.K. Luk, B. Maher, Y. Pan, C. Puhersch, M. Reso, M. Saroufim, M.Y. Sirachi, H. Suk, S. Zhang, M. Suo, P. Tillet, X. Zhao, E. Wang, K. Zhou, R. Zou, X. Wang, A. Mathews, W. Wen, G. Chanan, P. Wu, S. Chintala, Pytorch 2: faster machine learning through dynamic python bytecode transformation and graph compilation, in: Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2, ASPLOS '24, ACM, 2024, p. 929–947. <https://doi.org/10.1145/3620665.3640366>
- [137] O. Legeza, L. Veis, T. Mosoni, DMRG-Budapest, a program for condensed matter, quantum chemical, and nuclear shell DMRG calculations, 2022.
- [138] Strongly Correlated Systems “Lendület” Research Group, <https://wigner.hu/en/strongly-correlated-systems>.
- [139] A. Menczer, O. Legeza, Tensor network state algorithms on AI accelerators, *J. Chem. Theory Comput.* 20 (20) (2024) 8897–8910. <https://doi.org/10.1021/acs.jctc.4c00800>
- [140] A. Menczer, K. Kapás, M.A. Werner, O. Legeza, Two-dimensional quantum lattice models via mode optimized hybrid CPU-GPU density matrix renormalization group method, *Phys. Rev. B* 109 (19) (2024). <https://doi.org/10.1103/physrevb.109.195148>
- [141] A. Menczer, O. Legeza, Cost optimized ab initio tensor network state methods: industrial perspectives, 2024, <https://doi.org/10.48550/ARXIV.2412.04676>
- [142] A. Menczer, M. van Damme, A. Rask, L. Huntington, J. Hammond, S.S. Xantheas, M. Ganahl, O. Legeza, Parallel implementation of the density matrix renormalization group method achieving a quarter petaFLOPS performance on a single DGX-H100 GPU node, *J. Chem. Theory Comput.* 20 (19) (2024) 8397–8404. <https://doi.org/10.1021/acs.jctc.4c00903>
- [143] A. Menczer, O. Legeza, Massively parallel tensor network state algorithms on hybrid CPU-GPU based architectures, *J. Chem. Theory Comput.* 21 (4) (2025) 1572–1587. <https://doi.org/10.1021/acs.jctc.4c00661>
- [144] O. Legeza, A. Menczer, A. Ganyecz, M.A. Werner, K. Kapás, J. Hammond, S.S. Xantheas, M. Ganahl, F. Neese, Orbital optimization of large active spaces via AI-Accelerators, *J. Chem. Theory Comput.* 21 (13) (2025) 6545–6558. <https://doi.org/10.1021/acs.jctc.5c00571>

- [145] G. Alvarez, The density matrix renormalization group for strongly correlated electron systems: a generic implementation, *Comput. Phys. Commun.* 180 (9) (2009) 1572–1578. <https://doi.org/10.1016/j.cpc.2009.02.016>
- [146] G. Alvarez, L.G.G.V. Dias da Silva, E. Ponce, E. Dagotto, Time evolution with the density-matrix renormalization-group algorithm: a generic implementation for strongly correlated electronic systems, *Phys. Rev. E* 84 (5) (2011). <https://doi.org/10.1103/physreve.84.056706>
- [147] G. Alvarez, Implementation of the SU(2) Hamiltonian symmetry for the DMRG algorithm, *Comput. Phys. Commun.* 183 (10) (2012) 2226–2232. <https://doi.org/10.1016/j.cpc.2012.04.025>
- [148] G. Alvarez, Production of minimally entangled typical thermal states with the Krylov-space approach, *Phys. Rev. B* 87 (24) (2013). <https://doi.org/10.1103/physrevb.87.245130>
- [149] G. Alvarez, DMRG++ website, 2025, <https://g1257.github.io/dmrgPlusPlus/>.
- [150] DMRG++, 2025, <https://github.com/g1257/dmrgpp>.
- [151] W. Elwasif, E. D’Azevedo, A. Chatterjee, G. Alvarez, O. Hernandez, V. Sarkar, Miniapp for density matrix renormalization group hamiltonian application kernel, in: 2018 IEEE International Conference on Cluster Computing (CLUSTER), IEEE, 2018, p. 590–597. <https://doi.org/10.1109/cluster.2018.00075>
- [152] dmrgppPluginSc, 2025, https://code.ornl.gov/gonzalo_3/dmrgppPluginSc.
- [153] DMRGPy, 2025, <https://github.com/joselado/dmrgpy>.
- [154] M. Fishman, S. White, E. Stoudenmire, The ITensor software library for tensor network calculations, *SciPost Phys. Codebases* (2022). <https://doi.org/10.21468/scipostphyscodeb.4>
- [155] ITensor website, 2025, <https://itensor.org/>.
- [156] ITensor organization, 2025, <https://github.com/ITensor>.
- [157] C. Xiang, W. Jia, W.-H. Fang, Z. Li, Distributed multi-GPU ab initio density matrix renormalization group algorithm with applications to the P-Cluster of nitrogenase, *J. Chem. Theory Comput.* 20 (2) (2024) 775–786. <https://doi.org/10.1021/acs.jctc.3c01228>
- [158] Z. Li, Expressibility of comb tensor network states (CTNS) for the P-cluster and the femo-cofactor of nitrogenase, *Electronic Struct.* 3 (1) (2021) 014001. <https://doi.org/10.1088/2516-1075/abe192>
- [159] Z. Li, Time-reversal symmetry adaptation in relativistic density matrix renormalization group algorithm, *J. Chem. Phys.* 158 (4) (2023). <https://doi.org/10.1063/5.0127621>
- [160] H. Hong, W. Tong, X. Liu, T. Zhang, X.-Y. Liu, High performance single-Site finite DMRG on GPUs, in: CAIBDA 2022; 2nd International Conference on Artificial Intelligence, Big Data and Algorithms, 2022, pp. 1–5.
- [161] ITensor, 2025, <https://github.com/ITensor/ITensor>.
- [162] ITensorMPS.jl, 2025, <https://github.com/ITensor/ITensorMPS.jl>.
- [163] ITensors.jl, 2025, <https://github.com/ITensor/ITensors.jl>.
- [164] Y. Xu, C. Liu, H. Ma, Kylin-V: an open-source package calculating the dynamic and spectroscopic properties of large systems, *J. Chem. Phys.* 161 (5) (2024). <https://doi.org/10.1063/5.0220712>
- [165] Z. Xie, Y. Song, P. Peng, J. Li, Y. Cheng, L. Zhang, Y. Ma, Y. Tian, Z. Luo, H. Ma, Kylin 1.0: an ab-initio density matrix renormalization group quantum chemistry program, *J. Comput. Chem.* 44 (13) (2023) 1316–1328. <https://doi.org/10.1002/jcc.27085>
- [166] Kylin: An Ab-Initio Quantum Chemistry Package, 2025, <https://kylin-qc.com/>.
- [167] The Ma Research Group, 2025, <https://quantum-chemistry-cn.com/>.
- [168] J. Brabec, J. Brandeje, K. Kowalski, S. Xantheas, O. Legeza, L. Veis, Massively parallel quantum chemical density matrix renormalization group method, *J. Comput. Chem.* 42 (8) (2020) 534–544. <https://doi.org/10.1002/jcc.26476>
- [169] Libor Veis Research Group, 2025, <http://libor.veisovi.com/>.
- [170] Heyrovský Institute, 2025, <https://www.jh-inst.cas.cz/>.
- [171] M. Van Damme, L. Devos, J. Haegeman, MPSKit, 2024, <https://doi.org/10.5281/zenodo.14395415>
- [172] MPSKit.jl, 2025, <https://github.com/QuantumKitHub/MPSKit.jl>.
- [173] QuantumKitHub, 2025, <https://github.com/QuantumKitHub>.
- [174] Ghent University Quantum Group, 2025, <https://quantumghent.github.io/>.
- [175] L. Devos, J. Haegeman, TensorKit.jl: a julia package for large-scale tensor computations, with a hint of category theory, 2025. <https://doi.org/10.48550/ARXIV.2508.10076>
- [176] TensorKit.jl, 2025, <https://github.com/Jutho/TensorKit.jl>.
- [177] SUNRepresentations.jl, 2025, <https://github.com/QuantumKitHub/SUNRepresentations.jl>.
- [178] MPSKitModels.jl, 2025, <https://github.com/QuantumKitHub/MPSKitModels.jl>.
- [179] MPToolkit, 2025, <https://github.com/mptoolkit>.
- [180] MPToolkit, 2025, <https://github.com/mptoolkit/mptoolkit>.
- [181] M.L. Wall, L.D. Carr, Out-of-equilibrium dynamics with matrix product states, *New J. Phys.* 14 (12) (2012) 125015. <https://doi.org/10.1088/1367-2630/14/12/125015>
- [182] D. Jaschke, M.L. Wall, L.D. Carr, Open source matrix product states: opening ways to simulate entangled many-body quantum systems in one dimension, *Comput. Phys. Commun.* 225 (2018) 59–91. <https://doi.org/10.1016/j.cpc.2017.12.015>
- [183] D. Jaschke, L.D. Carr, Open source matrix product states: exact diagonalization and other entanglement-accurate methods revisited in quantum systems, *J. Phys. A: Math. Theor.* 51 (46) (2018) 465302. <https://doi.org/10.1088/1751-8121/aae4d1>
- [184] D. Jaschke, S. Montangero, L.D. Carr, One-dimensional many-body entangled open quantum systems with tensor network methods, *Quantum Sci. Technol.* 4 (1) (2018) 013001. <https://doi.org/10.1088/2058-9565/aae724>
- [185] OSMPS, 2025. <https://sourceforge.net/projects/openmps/>.
- [186] Carr Theoretical Physics Research Group, 2025, <https://people.mines.edu/lcarr/team/>.
- [187] C. Mendl, Pytenet: a concise python implementation of quantum tensor network algorithms, *J. Open Source Softw.* 3 (30) (2018) 948. <https://doi.org/10.21105/joss.00948>
- [188] PyTeNet, 2025, <https://github.com/cmendl/pytenet>.
- [189] L.P. Lindoy, D. Rodrigo-Albert, Y. Rath, I. Rungger, pyTTN: an open source toolbox for open and closed system quantum dynamics simulations using tree tensor networks, 2025. <https://doi.org/10.48550/ARXIV.2503.15460>
- [190] PyTTN, 2025, <https://gitlab.npl.co.uk/qsm/pytttn>.
- [191] K. Szeneš, N. Glaser, M. Erakovic, V. Barandun, M. Mörchen, R. Feldmann, S. Battaglia, A. Baiardi, M. Reiher, QCMAquis 4.0: multipurpose electronic, vibrational, and vibronic structure and dynamics calculations with the density matrix renormalization group, *J. Phys. Chem. A* (2025). <https://doi.org/10.1021/acs.jpca.5c02970>
- [192] S. Keller, M. Dolfi, M. Troyer, M. Reiher, An efficient matrix product operator representation of the quantum chemical hamiltonian, *J. Chem. Phys.* 143 (24) (2015). <https://doi.org/10.1063/1.4939000>
- [193] S. Knecht, E.D. Hedegård, S. Keller, A. Kovyrshin, Y. Ma, A. Muolo, C.J. Stein, M. Reiher, New approaches for ab initio calculations of molecules with strong electron correlation, *Chimia* 70 (4) (2016) 244. <https://doi.org/10.2533/chimia.2016.244>
- [194] QCMAquis website, 2025, <https://scine.ethz.ch/download/qcmaquis>.
- [195] QCMAquis, 2025, <https://github.com/qcscine/qcmaquis>.
- [196] SCINE website, 2025, <https://scine.ethz.ch/download/>.
- [197] QCSCINE, Software for chemical interaction networks, developed in the Reiher group at ETH Zurich, 2025, <https://github.com/qcscine>.
- [198] The Reiher Research Group, 2025, <https://reiher.ethz.ch/>.
- [199] A. Baiardi, C.J. Stein, V. Barone, M. Reiher, Vibrational density matrix renormalization group, *J. Chem. Theory Comput.* 13 (8) (2017) 3764–3777. <https://doi.org/10.1021/acs.jctc.7b00329>
- [200] A. Baiardi, C.J. Stein, V. Barone, M. Reiher, Optimization of highly excited matrix product states with an application to vibrational spectroscopy, *J. Chem. Phys.* 150 (9) (2019). <https://doi.org/10.1063/1.5068747>
- [201] A. Baiardi, M. Reiher, Large-scale quantum dynamics with matrix product states, *J. Chem. Theory Comput.* 15 (6) (2019) 3481–3498. <https://doi.org/10.1021/acs.jctc.9b00301>
- [202] A. Baiardi, A.K. Kelemen, M. Reiher, Excited-state DMRG made simple with FEAST, *J. Chem. Theory Comput.* 18 (1) (2021) 415–430. <https://doi.org/10.1021/acs.jctc.1c00984>
- [203] A. Muolo, A. Baiardi, R. Feldmann, M. Reiher, Nuclear-electronic all-particle density matrix renormalization group, *J. Chem. Phys.* 152 (20) (2020). <https://doi.org/10.1063/5.0007166>
- [204] R. Feldmann, A. Muolo, A. Baiardi, M. Reiher, Quantum proton effects from density matrix renormalization group calculations, *J. Chem. Theory Comput.* 18 (1) (2022) 234–250. <https://doi.org/10.1021/acs.jctc.1c00913>
- [205] S. Battaglia, S. Keller, S. Knecht, Efficient relativistic density-Matrix renormalization group implementation in a matrix-Product formulation, *J. Chem. Theory Comput.* 14 (5) (2018) 2353–2369. <https://doi.org/10.1021/acs.jctc.7b01065>
- [206] A. Weichselbaum, Non-abelian symmetries in tensor networks: a quantum symmetry space approach, *Ann. Phys.* 327 (12) (2012) 2972–3047. <https://doi.org/10.1016/j.aop.2012.07.009>
- [207] A. Weichselbaum, X-symbols for non-Abelian symmetries in tensor networks, *Phys. Rev. Res.* 2 (2) (2020). <https://doi.org/10.1103/physrevresearch.2.023385>
- [208] A. Weichselbaum, QSpace - An Open-source tensor library for abelian and non-Abelian symmetries, *SciPost Phys. Codebases* (2024). <https://doi.org/10.21468/scipostphyscodeb.40>
- [209] QSpace, Official public git repository, 2025, <https://bitbucket.org/qspace4u>.
- [210] D. Bacilieri, M. Ballarin, G. Cataldi, A. Costantini, D. Jaschke, G. Magnifico, S. Montangero, S. Notarnicola, A. Pagano, L. Pavesic, D. Rattacaso, M. Rigobello, N. Reinić, S. Scarlatella, P. Silvi, D. Wanisch, Quantum TEA: qtealeaves, 2024, <https://doi.org/10.5281/zenodo.13383350>
- [211] Quantum TEA website, 2025, <https://www.quantumtea.it/>.
- [212] Quantum TEA, 2025, https://baltig.infn.it/quantum_tea.
- [213] Quantum Information and Matter group, 2025, <https://quantum.dfa.unipd.it/>.
- [214] D. Jaschke, M. Ballarin, N. Reinić, L. Pavešić, S. Montangero, Benchmarking quantum red TEA on CPUs, GPUs, and TPUs, 2024. <https://doi.org/10.48550/ARXIV.2409.03818>
- [215] J. Gray, Quimb: a python package for quantum information and many-body calculations, *J. Open Source Softw.* 3 (29) (2018) 819. <https://doi.org/10.21105/joss.00819>
- [216] quimb, 2025, <https://github.com/jcmgray/quimb>.
- [217] J. Gray, S. Kourtis, Hyper-optimized tensor network contraction, *Quantum* 5 (2021) 410. <https://doi.org/10.22331/q-2021-03-15-410>
- [218] J. Gray, cotengra, 2025, <https://github.com/jcmgray/cotengra>.
- [219] J. Gray, autoray, 2025, <https://github.com/jcmgray/autoray/>.
- [220] J. Gray, symmray, 2025, <https://github.com/jcmgray/symmray>.
- [221] J. Ren, Z. Shuai, G. Kin-Lic Chan, Time-Dependent density matrix renormalization group algorithms for nearly exact absorption and fluorescence spectra of molecular aggregates at both zero and finite temperature, *J. Chem. Theory Comput.* 14 (10) (2018) 5027–5039. <https://doi.org/10.1021/acs.jctc.8b00628>
- [222] T. Jiang, W. Li, J. Ren, Z. Shuai, Finite temperature dynamical density matrix renormalization group for spectroscopy in frequency domain, *J. Phys. Chem Lett.* 11 (10) (2020) 3761–3768. <https://doi.org/10.1021/acs.jpcclett.0c00905>
- [223] J. Ren, W. Li, T. Jiang, Y. Wang, Z. Shuai, Time-dependent density matrix renormalization group method for quantum dynamics in complex systems, *WIREs Comput. Mol. Sci.* 12 (6) (2022). <https://doi.org/10.1002/wcms.1614>
- [224] Renormalizer, 2025, <https://github.com/shuaigrup/Renormalizer>.

- [225] Zhiqiang Shuai Group, 2025, <http://www.shuaigroup.net/>.
- [226] shuaigroup, 2025, <https://github.com/shuaigroup>.
- [227] J. Ren, W. Li, T. Jiang, Z. Shuai, A general automatic method for optimal construction of matrix product operators using bipartite graph theory, *J. Chem. Phys.* 153 (8) (2020). <https://doi.org/10.1063/5.0018149>
- [228] W. Li, J. Ren, H. Yang, H. Wang, Z. Shuai, Optimal tree tensor network operators for tensor network simulations: applications to open quantum systems, *J. Chem. Phys.* 161 (5) (2024). <https://doi.org/10.1063/5.0218773>
- [229] J.J. García-Ripoll, Quantum-inspired algorithms for multivariate analysis: from interpolation to partial differential equations, *Quantum* 5 (2021) 431. <https://doi.org/10.22331/q-2021-04-15-431>
- [230] P. García-Molina, L. Tagliacozzo, J.J. García-Ripoll, Global optimization of MPS in quantum-inspired numerical analysis, 2023. <https://doi.org/10.48550/ARXIV.2303.09430>
- [231] SeeMPS2, 2025, <https://github.com/juanjosegarciaripoll/seemps2>.
- [232] SeeMPS, 2025, <https://github.com/juanjosegarciaripoll/seemps>.
- [233] Quantum Information and Foundations Group – CSIC, 2025, <https://quinfog.hbar.es/>.
- [234] SUNDMRG.jl, 2025, <https://github.com/MGYamada/SUNDMRG.jl>.
- [235] S. Paeckel, T. Köhler, SymMPS website, 2025, <https://www.symmps.eu/>.
- [236] SymMPS, 2025, <https://www.symmps.eu/>.
- [237] S.C. Kramer, J. Hagemann, SciPAL: expression templates and composition closure objects for high performance computational physics with CUDA and openMP, *ACM Trans. Parallel Comput.* 1 (2) (2015) 1–31. <https://doi.org/10.1145/2686886>
- [238] S. Paeckel, T. Köhler, S.R. Manmana, Automated construction of $U(1)$ -invariant matrix-product operators from graph representations, *SciPost Phys.* 3 (5) (2017). <https://doi.org/10.21468/scipostphys.3.5.035>
- [239] C. Hubig, Symmetry-protected tensor networks, Ph.D. thesis, LMU München, 2017. <https://doi.org/10.5282/EDOC.21348>
- [240] C. Hubig, F. Lachenmaier, N.-O. Linden, T. Reinhard, L. Stenzel, A. Swoboda, M. Grundner, S. Mardazad, S. Paeckel, The SyTen Toolkit, 2025, <https://syten.eu>.
- [241] Group of Ulf Schollwöck, 2025, <https://www.theorie.physik.uni-muenchen.de/lsschollwoeck/>.
- [242] J. Hauschild, F. Pollmann, Efficient numerical simulations with tensor networks: tensor network python (tenPy), *SciPost Phys. Lecture Notes* (2018). <https://doi.org/10.21468/scipostphyslectnotes.5>
- [243] J. Hauschild, J. Unfried, S. Anand, B. Andrews, M. Bintz, U. Borla, S. Divic, M. Drescher, J. Geiger, M. Hefel, K. Hémerly, W. Kadow, J. Kemp, N. Kirchner, V.S. Liu, G. Moller, D. Parker, M. Rader, A. Romen, S. Scalet, L. Schoonderwoerd, M. Schulz, T. Soejima, P. Thoma, Y. Wu, P. Zechmann, L. Zweng, R. Mong, M. Zaletel, F. Pollmann, Tensor network python (tenPy) version 1, *SciPost Phys. Codebases* (2024). <https://doi.org/10.21468/scipostphyscodeb.41>
- [244] tenPy, 2025, <https://github.com/tenpy/tenpy>.
- [245] R. Levy, E. Solomonik, B.K. Clark, Distributed-Memory DMRG via sparse and dense parallel tensor contractions, in: *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis, IEEE*, 2020, p. 1–14. <https://doi.org/10.1109/sc41405.2020.00028>
- [246] tensor-tools, 2025, <https://github.com/ClarkResearchGroup/tensor-tools>.
- [247] E. Solomonik, D. Matthews, J.R. Hammond, J.F. Stanton, J. Demmel, A massively parallel tensor contraction framework for coupled-cluster computations, *J. Parallel Distrib. Comput.* 74 (12) (2014) 3176–3190. <https://doi.org/10.1016/j.jpdc.2014.06.002>
- [248] Clark research group, 2025, <https://clark.physics.illinois.edu/>.
- [249] L. Devos, L. Burgelman, B. Vanhecke, J. Haegeman, F. Verstraete, L. Vanderstraeten, *TensorTrack*, 2024, <https://doi.org/10.5281/zenodo.10641042>
- [250] *TensorTrack*, 2025, <https://github.com/quantumghent/TensorTrack>.
- [251] *UltraDMRG*, 2025, <https://github.com/QuantumLiquids/UltraDMRG>.
- [252] H.-X. Wang, *TensorToolkit*, 2025, <https://github.com/QuantumLiquids/TensorToolkit>.
- [253] xDMRG + +, 2025, <https://github.com/DavidAce/xDMRGpp>.
- [254] M. Rams, G. Wojtowicz, A. Sinha, J. Hasik, YASTN: Yet another symmetric tensor networks; a python library for abelian symmetric tensor network calculations, *SciPost Phys. Codebases* (2025). <https://doi.org/10.21468/scipostphyscodeb.52>
- [255] YASTN, 2025, <https://github.com/yastn/yastn>.
- [256] J. Vance, Large-Scale Implementation of the Density Matrix Renormalization Group Algorithm, Master's thesis, SISSA, 2017. <https://doi.org/20.500.11767/68070>
- [257] J. Vance, *DMRG.x*, 2025, <https://github.com/jvance/DMRG.x>.
- [258] J.J. García-Ripoll, *mPS*, 2025, <https://github.com/juanjosegarciaripoll/mPS>.
- [259] N. Nakatani, *MPS + +*, 2025, <https://github.com/naokin/mpsxx>.
- [260] M. Ganahl, *MPSTools*, 2024, <https://github.com/mganahl/MPSTools>.
- [261] Y. Kurashige, G.K.-L. Chan, T. Yanai, Entangled quantum electronic wavefunctions of the mn4cao5 cluster in photosystem II, *Nat. Chem.* 5 (8) (2013) 660–666. <https://doi.org/10.1038/nchem.1677>
- [262] *paralleldMRG*, 2025, <https://github.com/emstoudenmire/paralleldMRG>.
- [263] P. Helms, *PyDMRG*, 2025, <https://github.com/philliphelms/PyDMRG>.
- [264] M. Ganahl, *PyTeN*, 2025, <https://github.com/mganahl/PyTeN>.
- [265] Z. Li, QC-MPO-DMRG, 2016, <https://github.com/zhendongli2008/qcmopdmrg>.
- [266] Y. Kurashige, T. Yanai, High-performance ab initio density matrix renormalization group method: applicability to large-scale multireference problems for metal compounds, *J. Chem. Phys.* 130 (23) (2009). <https://doi.org/10.1063/1.3152576>
- [267] C. Guo, A. Weichselbaum, S. Kehrein, T. Xiang, J. von Delft, Density matrix renormalization group study of a quantum impurity model with Landau-Zener time-dependent Hamiltonian, *Phys. Rev. B* 79 (11) (2009). <https://doi.org/10.1103/physrevb.79.115137>
- [268] Snake DMRG, 2025, <https://github.com/entron/snake-dmrg>.
- [269] C. Roberts, A. Milsted, M. Ganahl, A. Zalcman, B. Fontaine, Y. Zou, J. Hidary, G. Vidal, S. Leichenauer, *TensorNetwork*: a library for physics and machine learning, 2019, <https://doi.org/10.48550/ARXIV.1905.01330>
- [270] *TensorNetwork*, 2025, <https://github.com/google/TensorNetwork>.
- [271] S. Al-Assam, S.R. Clark, D. Jaksch, The tensor network theory library, *J. Stat. Mech: Theory Exp.* 2017 (9) (2017) 093102. <https://doi.org/10.1088/1742-5468/aa7df3>
- [272] Z. Li, ZMPO-DMRG, 2025, https://github.com/zhendongli2008/zmpo_dmrg.
- [273] *BLOCK*, 2025, <https://github.com/sanshar/Block>.
- [274] *StackBlock*, 2025, <https://github.com/sanshar/StackBlock>.
- [275] *GraceQ*, *MPS2*, 2025, <https://github.com/gracequantum/MPS2>.
- [276] Y.-J. Kao, Y.-D. Hsieh, P. Chen, Uni10: an open-source library for tensor network algorithms, *J. Phys. Conf. Ser.* 640 (2015) 012040. <https://doi.org/10.1088/1742-6596/640/1/012040>
- [277] uni10, 2025, <https://gitlab.com/uni10/uni10/>.
- [278] J.R. Garrison, R.V. Mishmash, simple-dmrg/simple-dmrg: Simple DMRG 1.0, 2017, <https://doi.org/10.5281/zenodo.1068359>
- [279] J.R. Garrison, R.V. Mishmash, Sophisticated DMRG, 2021. <https://doi.org/10.5281/zenodo.4651419>
- [280] G. Evenbly, *Tensors.net*, 2025, <https://www.tensors.net/>.
- [281] *ITensorNetworks.jl*, 2025, <https://github.com/ITensor/ITensorNetworks.jl>.
- [282] G.K.-L. Chan, An algorithm for large scale density matrix renormalization group calculations, *J. Chem. Phys.* 120 (7) (2004) 3172–3178. <https://doi.org/10.1063/1.1638734>
- [283] Y. Tian, Z. Xie, Z. Luo, H. Ma, Mixed-Precision implementation of the density matrix renormalization group, *J. Chem. Theory Comput.* 18 (12) (2022) 7260–7271. <https://doi.org/10.1021/acs.jctc.2c00632>
- [284] G. Hager, E. Jeckelmann, H. Fehske, G. Wellein, Parallelization strategies for density matrix renormalization group algorithms on shared-memory systems, *J. Comput. Phys.* 194 (2) (2004) 795–808. <https://doi.org/10.1016/j.jcp.2003.09.018>
- [285] N. Bultinck, D.J. Williamson, J. Haegeman, F. Verstraete, Fermionic matrix product states and one-dimensional topological phases, *Phys. Rev. B* 95 (7) (2017). <https://doi.org/10.1103/physrevb.95.075108>
- [286] Q. Mortier, L. Devos, L. Burgelman, B. Vanhecke, N. Bultinck, F. Verstraete, J. Haegeman, L. Vanderstraeten, Fermionic tensor network methods, *SciPost Phys.* 18 (1) (2025). <https://doi.org/10.21468/scipostphys.18.1.012>
- [287] Y. Gao, H. Zhai, J. Gray, R. Peng, G. Park, W.-Y. Liu, E.F. Kjonstad, G.K.-L. Chan, Fermionic tensor network contraction for arbitrary geometries, 2024, <https://doi.org/10.48550/ARXIV.2410.02215>
- [288] G.M. Crosswhite, D. Bacon, Finite automata for caching in matrix product algorithms, *Phys. Rev. A* 78 (1) (2008). <https://doi.org/10.1103/physreva.78.012356>
- [289] S. Szalay, M. Pfeiffer, V. Murg, G. Barcza, F. Verstraete, R. Schneider, O. Legeza, Tensor product methods and entanglement optimization for ab initio quantum chemistry, *Int. J. Quantum Chem.* 115 (19) (2015) 1342–1391. <https://doi.org/10.1002/qua.24898>
- [290] B.-B. Chen, L. Chen, Z. Chen, W. Li, A. Weichselbaum, Exponential thermal tensor network approach for quantum lattice models, *Phys. Rev. X* 8 (3) (2018). <https://doi.org/10.1103/physrevx.8.031082>
- [291] D.E. Parker, X. Cao, M.P. Zaletel, Local matrix product operators: canonical form, compression, and control theory, *Phys. Rev. B* 102 (3) (2020). <https://doi.org/10.1103/physrevb.102.035147>
- [292] T. Felser, Tree tensor networks for high-dimensional quantum systems and beyond, Ph.D. thesis, University of Saarland, 2021. <https://doi.org/10.22028/D291-35211>
- [293] H. Çakır, R.M. Milbradt, C.B. Mendl, Optimal symbolic construction of matrix product operators and tree tensor network operators, *Phys. Rev. B* 112 (3) (2025). <https://doi.org/10.1103/physrevb.112.034102>
- [294] O. Legeza, J. Sólyom, Optimizing the density-matrix renormalization group method using quantum information entropy, *Phys. Rev. B* 68 (19) (2003). <https://doi.org/10.1103/physrevb.68.195116>
- [295] J. Brandejs, N. Hörnblad, E.F. Valeev, A. Heinecke, J. Hammond, D. Matthews, P. Bientinesi, Tensor Algebra Processing Primitives (TAPP): towards a standard for tensor operations, 2026, <https://doi.org/10.48550/ARXIV.2601.07827>
- [296] Cyten, 2025, <https://github.com/tenpy/cyten>.
- [297] Reusable libraries in quantum chemistry 2025, 2025, <https://www.helsinki.fi/en/conferences/reusable-libraries-quantum-chemistry-2025>.
- [298] S. Lehtola, A call to arms: making the case for more reusable libraries, *J. Chem. Phys.* 159 (18) (2023). <https://doi.org/10.1063/5.0175165>
- [299] J. Brandejs, T. Saue, A.S.P. Gomes, L. Visscher, P. Bientinesi, Report on the second toulouse tensor workshop, 2026, <https://doi.org/10.48550/ARXIV.2602.05490>