Subspace Iteration Methods in terms of Matrix Product States

Thomas Huckle[1] and **Konrad Waldherr**[1,*]

[1] Boltzmannstraße 3 85748 Garching

When dealing with quantum many-body systems one is faced with problems growing exponentially in the number of particles to be considered. To overcome this curse of dimensionality one has to consider representation formats which scale only polynomially. Physicists developed concepts like *matrix product states* (MPS) to represent states of interest and formulated algorithms such as the *density matrix renormalization group* (DMRG) to find such states. We consider the standard Lanczos algorithm and formulate it for vectors given in the MPS format. It turns out that a restarted version which includes a projection onto the MPS manifold gives the same approximation quality as the well-established DMRG method. Moreover, this variant is more flexible and provides more information about the spectrum.

Copyright line will be provided by the publisher

1 Introduction

In the simulation of an N-particle quantum system, the computation of the *ground state*, i.e. the lowest-energy state of the system under consideration, is an important task, because such a state typically appears. From a linear algebraic point of view, the physical system can be described by a Hermitian matrix, the so-called *Hamiltonian* $\boldsymbol{H}$. Its spectrum describes the set of possible energy levels and its eigenvectors represent the corresponding states. Therefore, the ground state is the eigenvector related to the smallest eigenvalue of $\boldsymbol{H}$. As an example (compare [1]), the Ising-ZZ model for spin-1/2 particles reads

$$\boldsymbol{H}^{\text{Is}} = \sum_{k=1}^{N-1} \boldsymbol{I}^{\otimes(k-1)} \otimes \boldsymbol{\sigma_z} \otimes \boldsymbol{\sigma_z} \otimes \boldsymbol{I}^{\otimes(N-k-1)} + \lambda \sum_{k=1}^{N} \boldsymbol{I}^{\otimes(k-1)} \otimes \boldsymbol{\sigma_x} \otimes \boldsymbol{I}^{\otimes(N-k)} \tag{1}$$

with the 2 × 2 identity $\boldsymbol{I}$ and the *Pauli matrices* $\boldsymbol{\sigma_x} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ and $\boldsymbol{\sigma_z} = \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$. As the dimension of the underlying Hilbert space grows exponentially in N, we have to consider data-sparse representation formats to express the vectors of interest and to formulate algorithms acting on such formats. To compute low-lying eigenstates, physicists invented the DMRG method [2], a variational ansatz in the MPS [3] formalism. In the following, we present the MPS format from a mathematical perspective and modify the Lanczos algorithm for vectors given in the MPS format.

2 Matrix Product States: Formalism and Computations

In the MPS format, a 2^N-dimensional vector $\boldsymbol{x}$ is considered as binary tensor of order N and represented by

$$\boldsymbol{x} = \sum_{i_1,\ldots,i_N} \operatorname{tr}\left(\boldsymbol{A}_{\boldsymbol{1}}^{(i_1)} \boldsymbol{A}_{\boldsymbol{2}}^{(i_2)} \cdots \boldsymbol{A}_{\boldsymbol{N}}^{(i_N)}\right) \left(\boldsymbol{e}_{i_1} \otimes \boldsymbol{e}_{i_2} \otimes \cdots \otimes \boldsymbol{e}_{i_N}\right) \tag{2}$$

with $D_j \times D_{j+1}$ matrices $\boldsymbol{A}_{\boldsymbol{j}}^{(0)}$ and $\boldsymbol{A}_{\boldsymbol{j}}^{(1)}$. For *open boundary conditions* (OBC) there is no connection between particles 1 and N and therefore the matrices at both ends simplify to vectors (i.e. $D_1 = D_{N+1} = 1$). Such MPS correspond to the *Tensor Train* format [4]. The numbers D_j define the ranks of the representation and are usually referred to as *bond dimensions*.

For the computation of **inner products** of two MPS vectors, one has to find a convenient ordering of all summations that have to be executed. Following [5], an efficient contraction scheme requires $\mathcal{O}(ND^3)$ for OBC. For expressing the **sum** of two MPS vectors, we have to build block diagonal matrices of the original MPS matrices for each $j = 2, \ldots, N-1$ and at both ends we have to concatenate the original vectors. Hence, adding MPS vectors leads to an augmentation of the ranks. Finally we consider the **application of an operator** to an MPS. To this end, it will be useful to introduce the concept of *matrix product operators* (MPO) [6], which is a generalization of MPS to operators $\boldsymbol{H}$. The general format of an MPO is

$$\boldsymbol{H} = \sum_{\substack{i_1,\ldots,i_N \\ k_1,\ldots,k_N}} \operatorname{tr}\left(\boldsymbol{W}_{\boldsymbol{1}}^{(i_1,k_1)} \boldsymbol{W}_{\boldsymbol{2}}^{(i_2,k_2)} \cdots \boldsymbol{W}_{\boldsymbol{N}}^{(i_N,k_N)}\right) \left(\left(\boldsymbol{e}_{i_1}\boldsymbol{e}_{\boldsymbol{k_1}}^{\mathbf{T}}\right) \otimes \left(\boldsymbol{e}_{i_2}\boldsymbol{e}_{\boldsymbol{k_2}}^{\mathbf{T}}\right) \otimes \cdots \otimes \left(\boldsymbol{e}_{i_N}\boldsymbol{e}_{\boldsymbol{k_N}}^{\mathbf{T}}\right)\right) \tag{3}$$

with $D_j^{(\boldsymbol{H})} \times D_{j+1}^{(\boldsymbol{H})}$ matrices $\boldsymbol{W}_{\boldsymbol{j}}^{(i_j,k_j)}$. Now, the product of the matrix $\boldsymbol{H}$ (3) with the vector $\boldsymbol{x}$ (2) is

$$(\boldsymbol{Hx})_{i_1,\ldots,i_N} = \operatorname{tr}\left(\left(\sum_{k_1=0}^{1} \boldsymbol{W}_{\boldsymbol{1}}^{(i_1,k_1)} \otimes \boldsymbol{A}_{\boldsymbol{1}}^{(k_1)}\right) \cdots \left(\sum_{k_N=0}^{1} \boldsymbol{W}_{\boldsymbol{N}}^{(i_N,k_N)} \otimes \boldsymbol{A}_{\boldsymbol{N}}^{(k_N)}\right)\right) ,$$

[*] Corresponding author: email waldherr@in.tum.de, phone +00 49 089 28918613, fax +00 49 089 28918607

Copyright line will be provided by the publisher

which is again an MPS, but with augmented ranks $D_j^{(\boldsymbol{H})} D_j$.

3 Lanczos Algorithm for MPS Vectors

Both adding MPS vectors and applying an MPO to an MPS lead to an augmentation of the ranks. Hence, if we want to apply the Lanczos algorithm to MPS vectors we have to introduce a **compression** $\mathcal{P}_{D_{\max}}$ of MPS vectors in order to keep their ranks limited by some prescribed $D_{\max}$. To achieve this we use successive SVD-based truncations. We start from the left, rearrange the MPS matrices at site r into a rectangular matrix, carry out an SVD and do the truncation:

$$\begin{pmatrix} \boldsymbol{A}_r^{(0)} \\ \boldsymbol{A}_r^{(1)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{U}_r^{(0)} \\ \boldsymbol{U}_r^{(1)} \end{pmatrix} \boldsymbol{\Sigma}_r \boldsymbol{V}_r^{\mathrm{H}} \overset{trunc}{\approx} \begin{pmatrix} \tilde{\boldsymbol{U}}_r^{(0)} \\ \tilde{\boldsymbol{U}}_r^{(1)} \end{pmatrix} \tilde{\boldsymbol{\Sigma}}_r \tilde{\boldsymbol{V}}_r^{\mathrm{H}} .$$

We replace $\boldsymbol{A}_r^{(i_r)}$ by $\tilde{\boldsymbol{U}}_r^{(i_r)}$ and proceed recursively with $\tilde{\boldsymbol{\Sigma}}_r \tilde{\boldsymbol{V}}_r^{\mathrm{H}} \boldsymbol{A}_{r+1}^{(i_{r+1})}$.

As we want to avoid solving linear systems in the Hamiltonian, which is required by algorithms such as Jacobi-Davidson, we propose to consider the Lanczos algorithm, but to modify it to a restarted version. Then, there are different variants where the compression can be performed. In order to keep the ranks as small as possible without loss of orthogonality we propose to introduce the compression $\mathcal{P}_D$ for the matrix-vector product in each Lanczos iteration and for the Ritz vector obtained at the end of each outer iteration. Algorithm 1 sketches the proposed procedure.

Algorithm 1 Restarted Lanczos with compression

```
Choose initial guess $\boldsymbol{r_0} \neq \boldsymbol{0}$
for $iter = 1, 2, \dots$ do
    Define $\beta_0 = \|\boldsymbol{r_0}\|$ and $\boldsymbol{q_0} = \boldsymbol{0}$
    for $j = 1, 2, \dots, m$ do
        $\boldsymbol{q_j} = \boldsymbol{r_{j-1}} / \beta_{j-1}$
        $\boldsymbol{u_j} = \mathcal{P}_D\left(\boldsymbol{H} \boldsymbol{q_j}\right)$
        $\alpha_j = \boldsymbol{q_j}^{\mathrm{H}} \boldsymbol{u_j}$
        $\boldsymbol{r_j} = \boldsymbol{u_j} - \alpha_j \boldsymbol{q_j} - \beta_{j-1} \boldsymbol{q_{j-1}}$
        $\beta_j = \|\boldsymbol{r_j}\|$
    end for
    $\boldsymbol{T_m} = \mathrm{tridiag}(\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\beta})$
    $(\lambda, \boldsymbol{y}) = \mathrm{eig}(\boldsymbol{T_m})$
    $\boldsymbol{v} = \boldsymbol{Q}\boldsymbol{y} = (\boldsymbol{q_1}, \dots, \boldsymbol{q_m})\boldsymbol{y}$
    $\boldsymbol{r_0} = \mathcal{P}_D(\boldsymbol{v})$
end for
```

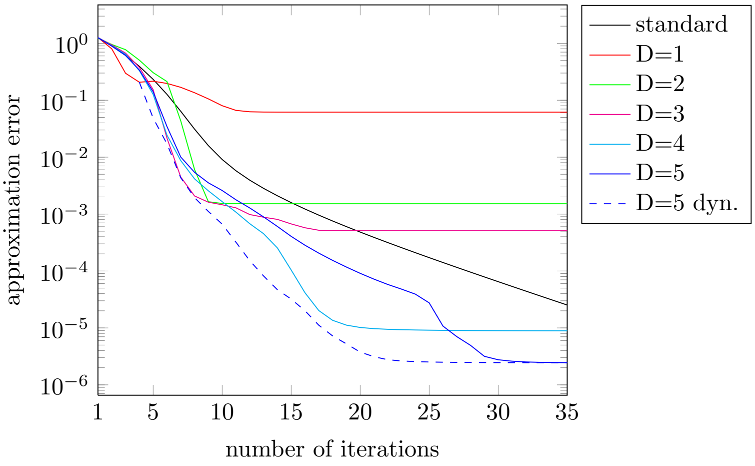

Fig. 1 Application of Alg. 1 ($m = 3$) to the 20 spin Ising model. The introduction of the compression even leads to an improvement of the eigenvalue approximation.

We applied Algorithm 1 (subspace dimension $m = 3$) to compute the lowest eigenvalue of the Ising model (1) with $N = 20$ particles and compared the convergence behavior for different choices of bond dimensions D with an exact Lanczos version, where no compression is carried out, see Fig. 1. We obtain the same approximation quality as achieved by the state-of-the-art DMRG method, but, however, DMRG requires less iteration steps. Our results show that the compression to smaller ranks even leads to an improvement compared to the classical standard Lanczos. This gives reason to use a dynamic version, where we start with comparably small rank D in order to obtain proper initial values for larger D.

Altogether, the restarted Lanczos algorithm introduces more flexibility (choice of D and m) and allows modifications to obtain more information on the spectrum such as the gap between the two smallest eigenvalues.

Acknowledgements This work was supported in part by the Bavarian excellence network ENB via the International Doctorate Program of Excellence Quantum Computing, Control, and Communication (QCCC).

References

[1] T. Huckle, K. Waldherr, and T. Schulte-Herbrüggen, Linear Multilinear Algebra (2012), in press: 10.1080/03081087.2012.663371.
[2] S. R. White, Phys. Rev. Lett. **69**(Nov), 2863–2866 (1992).
[3] S. Rommer and S. Ostlund, PHYS.REV.B **55**, 2164 (1997).
[4] I. Oseledets, SIAM J. Scientific Computing **33**, 2295–2317 (2011).
[5] T. Huckle, K. Waldherr, and T. Schulte-Herbrüggen, Linear Algebra Appl. (2012), in press: 10.1016/j.laa.2011.12.019.
[6] F. Verstraete, J. J. García-Ripoll, and J. I. Cirac, Phys. Rev. Lett. **93**(Nov), 207204 (2004).

Copyright line will be provided by the publisher