Computation of extreme eigenvalues in higher dimensions using block tensor train format[*]

S. V. Dolgov[†], B. N. Khoromskij[†], I. V. Oseledets[‡], D. V. Savostyanov[‡§]

June 12, 2013

Abstract

We consider an approximate computation of several minimal eigenpairs of large Hermitian matrices which come from high–dimensional problems. We use the tensor train format (TT) for vectors and matrices to overcome the curse of dimensionality and make storage and computational cost feasible. Applying a block version of the TT format to several vectors simultaneously, we compute the low–lying eigenstates of a system by minimization of a block Rayleigh quotient performed in an alternating fashion for all dimensions. For several numerical examples, we compare the proposed method with the deflation approach when the low–lying eigenstates are computed one-by-one, and also with the variational algorithms used in quantum physics.

Keywords: high–dimensional problems, DMRG, MPS, tensor train format, low–lying eigenstates.

MSC: 15A18, 15A69, 65F15, 82B28, 82B20,

PACS: 02.10.Xm, 02.60.Dc, 75.10.Pq, 05.10.Cc

1 Introduction

High-dimensional problems are notoriously difficult to solve by standard numerical techniques due to the *curse of dimensionality* — the complexity grows exponentially with the number of degrees of freedom. The problems of such kind arise in many different applications in physics, chemistry, biology and engineering, but their study in numerical linear algebra has begun quite recently.

There are not many techniques capable of solving high–dimensional problems efficiently. The most prominent among them are Monte Carlo and quasi Monte

[*] Partially supported by RFBR grants 11-01-00549-a, 12-01-33013, 12-01-00546-a, 12-01-91333-nnio-a, Rus. Fed. Gov. project 16.740.12.0727 at the Institute of Numerical Mathematics RAS and EPSRC grant EP/H003789/1 at the University of Southampton.

[†] Max-Planck Institute for Mathematics in the Sciences, Inselstrasse 22, Leipzig 04103, Germany (bokh@mis.mpg.de, dolgov@mis.mpg.de)

[‡] Institute of Numerical Mathematics of Russian Academy of Sciences, Gubkina 8, Moscow 119333, Russia (ivan.oseledets@gmail.com)

[§] University of Southampton, School of Chemistry, Highfield Campus, Southampton SO17 1BJ, United Kingdom (dmitry.savostyanov@gmail.com)

Carlo methods, best N-term approximations, and advanced discretization methods such as sparse grids and radial basis functions. However, all of these methods have their own disadvantages. For example, it is difficult to achieve high accuracy using the Monte Carlo approach, and sparse grid techniques require sophisticated analytical and algebraic manipulations and still suffer (in a milder way though) from the curse of dimensionality, which make them inapplicable for $d \gtrsim 10$.

One of the most fruitful ideas for solving high-dimensional problems is the idea of *separation of variables*. For two variables it boils down to the celebrated Schmidt decomposition, which is known on a discrete level as the singular value decomposition (SVD), a particular low–rank decomposition of a matrix. Different generalizations of this idea to higher dimensions, most notable are the canonical (CP) and Tucker formats, have been studied motivated by the applications in data analysis, particularly chemometrics, see the review [27]. These classical formats have their drawbacks as well, e.g. the canonical format is in general not stable to perturbations, and the Tucker format suffers from the curse of dimensionality. Nevertheless, in many applications the canonical representation can be computed efficiently using, e.g. *greedy algorithms* [1, 28, 43] or by a multigrid accelerated reduced higher order SVD combined with the Tucker format [23], and for the Tucker format a quasioptimal approximation can be computed using the SVD algorithm [3].

Efficient methods for quantum many-body systems are based on low–parametric tensor product formats. One of the most successful approaches, the *density matrix renormalization group* (DMRG) [45, 46] is an optimization technique that uses the matrix product state (MPS) representation, see the review [41]. The MPS and DMRG are described in a problem–specific language, and despite they became the methods of choice for many applications in the solid state physics and quantum chemistry, they were unknown in numerical analysis.

Looking for more efficient dimensionality reduction schemes, two groups in the numerical linear algebra community have re-discovered independently successful tensor formats under different names, namely the Tree-Tucker [35], tensor train (TT) [33] and hierarchical Tucker (HT) [15, 12] formats. The equivalence of the TT and MPS format has been shortly discovered and reported in [16]. This connection is very beneficial and fruitful: the idea of the DMRG algorithm has been applied to different kinds of problem in numerical analysis: approximate solution of linear systems [16, 8], solution of eigenvalue problems [25], dynamics [6], cross interpolation [40]. At the same time, new tensor formats have been proposed, e.g. the *quantized* tensor train (QTT) [21, 31], and the QTT-Tucker [4], and several new results have been obtained for eigenproblems [29], solution of linear systems [9, 10], multidimensional convolution [20, 13, 17], multidimensional Fourier transform [7], interpolation [39].

The use of different tensor product formats for quantum chemistry problems has been considered in [22, 11, 23, 14, 24, 19]. This paper can be considered as a natural extension of the work [25], where the computation of a single extreme eigenvalue using the (quantized) tensor train format was revisited. Instead of computing one eigenvalue we want to compute approximations to several extreme eigenpairs simultaneously. Such problems appear, for example, in the computation of excited states in physics and chemistry. This generalization is not straightforward, taking into account the special structure of the manifold where the solution is sought. Our approach consists in two key components. First, all eigenstates are represented in the TT-format in a simultaneous manner (so-called *block TT-format*), and the minimization problem is formulated in a rigorous way as a minimization of the Block Rayleigh quotient. The optimization problem is then solved by means of alternating optimization with a small but very important trick: instead of using one-and-the same representation for all steps, the auxiliary index corresponding to the eigenstate number is always associated with the currently optimized TT core. The local problem at each iteration step becomes a linear block eigenvalue problem which can be treated by standard iterative techniques.

At the moment the draft of this paper was written, Frank Verstraete and Iztok Pizorn kindly informed us about the work [36] in the MPS community, which also addresses solution of the block Rayleigh minimization problem in the alternating framework, but the methods described in [36] differ from the algorithm proposed in this paper.

Our algorithm has a better asymptotic complexity with respect to the mode size, but adapts the TT ranks during the iterations like in the DMRG scheme. This can be particularly important for solving high–dimensional problems with large mode sizes.

The paper is organized as follows. In Section 2 definitions of the tensor train format are introduced. In Section 3 the algorithm is presented, and the complexity is analyzed and compared to the algorithms used in quantum physics. Sections 4, 5, 6 contain numerical experiments for the particle in a box, the Henon–Heiles potential, and the Heisenberg model, including the comparison of the computational speed with the publicly available DMRG implementations.

2 Notation, definitions and preliminaries

We consider the eigenproblem $AX = X\Lambda$, with the Hermitian matrix $A = A^*$, and we are interested in B extreme eigenvalues λ_b and their eigenvectors x_b, for $b = 1, \dots, B$. This problem is equivalent to the minimization of the block Rayleigh quotient

$$\operatorname{trace}(X^*AX) \to \min, \qquad \text{s.t.} \quad X^*X = I, \tag{1}$$

where $X = [x_b]_{b=1}^{B}$ contains the orthogonal eigenvectors.

We assume that the problem has a tensor-product structure, i.e. all eigenvectors involved can be associated with d-dimensional tensors. Specifically, the elements of a vector $x = [x(i)]_{i=1}^{N}$ can be enumerated with d *mode indices* $i_1, \dots, i_d$ by a linear map $i = \overline{i_1 \dots i_d}$. The mode indices run through $i_k = 1, \dots, n_k$, where n_k are referred to as the *mode sizes* for $k = 1, \dots, d$. Naturally, $N = n_1 \dots n_d$, and if all mode sizes are of the same order $n_k \sim n$, the number of unknowns grows exponentially with the dimension, $N \sim n^d$. To make the problem tractable, we use the *tensor train* (TT) format [33], defined as follows,

$$\begin{aligned} x(i) &= x(\overline{i_1 \dots i_d}) = X^{(1)}(i_1) \dots X^{(d)}(i_d) \\ &= \sum_{\boldsymbol{\alpha}} X^{(1)}_{\alpha_1}(i_1) \dots X^{(k-1)}_{\alpha_{k-2},\alpha_{k-1}}(i_{k-1}) X^{(k)}_{\alpha_{k-1},\alpha_k}(i_k) X^{(k+1)}_{\alpha_k,\alpha_{k+1}}(i_{k+1}) \dots X^{(d)}_{\alpha_{d-1}}(i_d). \end{aligned} \tag{2}$$

Here and later we write equations in the *elementwise notation*, i.e. assume they hold for all possible values of all free indices. The summation over $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_{d-1})$

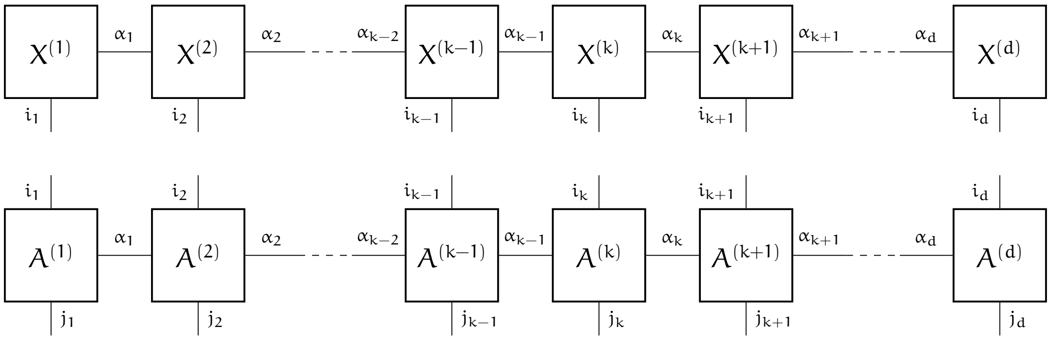

Figure 1: Above: tensor train format (2) shown as a linear tensor network. Below: tensor train format (9) for a matrix in higher dimensions. The boxes are tensors with lines (legs) denoting indices. Each bond between two tensors assumes a summation over the joint index.

assumes the summation over all possible values of all *auxiliary* (or bond) indices $\alpha_k = 1, \dots, r_k$, where numbers $r_1, \dots, r_{d-1}$ are referred to as the *tensor train ranks* (TT–ranks). Each $X^{(k)}(i_k)$ is an $r_{k-1} \times r_k$ matrix, i.e. each entry of a vector $x = [x(i)]$ is represented by a product of d matrices in the right–hand side. The three–dimensional arrays $X^{(k)} = [X^{(k)}_{\alpha_{k-1},\alpha_k}(i_k)]$ of size $r_{k-1} \times n_k \times r_k$ are referred to as the *TT–cores*. In the representation (2), the vector x is seen as a *vectorization* of a d–tensor, which explains the name of the format. The tensor train format is a *linear tensor network*, and can be illustrated as a graph, see Fig. 1.

In this paper we represent all eigenvectors of interest *simultaneously* by the *block* tensor train format.

Definition 1 (block TT-format)**.** The vectors $x_b(i)$ are said to be in the block–TT format, if

$$x_b(i) = x_b(\overline{i_1 \dots i_d}) = X^{(1)}(i_1) \dots X^{(p-1)}(i_{p-1})\hat{X}^{(p)}(i_p, b)X^{(p+1)}(i_{p+1}) \dots X^{(d)}(i_d). \tag{3}$$

The choice of the mode p which *carries* the index b is not fixed — we will move it back and forth during the optimization. When the position p is chosen, it means that the matrix $\hat{X}^{(p)}(i_p, b)$ is additionally parametrized by the index b. The 'block' TT–core $\hat{X}^{(p)} = [\hat{X}^{(p)}_{\alpha_{p-1},\alpha_p}(i_p, b)]$ is now a tensor with four indices.

Following [37], we define the *interfaces* $X^{<k}$ of size $n_1 \dots n_{k-1} \times r_{k-1}$ and $X^{>k}$ of size $r_k \times n_{k+1} \dots n_d$ as follows

$$\begin{aligned} X^{<k}(\overline{i_1 i_2 \dots i_{k-1}}, \beta_{k-1}) &= \sum_{\alpha_1 \dots \alpha_{k-2}} X^{(1)}_{\alpha_1}(i_1) X^{(2)}_{\alpha_1 \alpha_2}(i_2) \dots X^{(k-1)}_{\alpha_{k-2},\beta_{k-1}}(i_{k-1}), \\ X^{>k}(\beta_k, \overline{i_{k+1} \dots i_{d-1} i_d}) &= \sum_{\alpha_{k+1} \dots \alpha_{d-1}} X^{(k+1)}_{\beta_k,\alpha_{k+1}}(i_{k+1}) \dots X^{(d-1)}_{\alpha_{d-2},\alpha_{d-1}}(i_{d-1}) X^{(d)}_{\alpha_{d-1}}(i_d). \end{aligned} \tag{4}$$

Using the interfaces we introduce the *frame matrix* (see Fig. 2) as follows

$$\begin{aligned} &\mathcal{X}_{\neq k} = X^{<k} \otimes I_{n_k} \otimes \left(X^{>k}\right)^\top, \\ &\mathcal{X}_{\neq k}(\overline{i_1 \dots i_d}, \overline{\beta_{k-1} j_k \beta_k}) \\ &\quad = \sum_{\substack{\alpha_1,\dots,\alpha_{k-2} \\ \alpha_{k+1} \dots \alpha_d}} X^{(1)}_{\alpha_1}(i_1) \dots X^{(k-1)}_{\alpha_{k-2},\beta_{k-1}}(i_{k-1}) \delta(i_k, j_k) X^{(k+1)}_{\beta_k,\alpha_{k+1}}(i_{k+1}) \dots X^{(d)}_{\alpha_{d-1}}(i_d). \end{aligned} \tag{5}$$

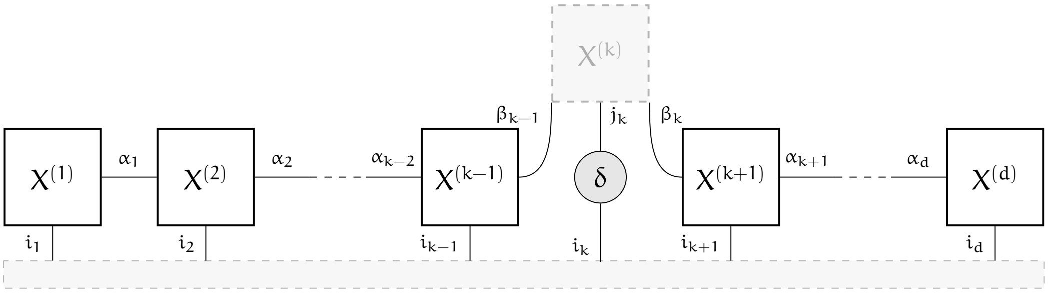

Figure 2: The frame matrix (5) maps a TT core (above) into a large vector (below).

The tensor train format (2) is polylinear with respect to the TT–cores, and in particular the block-TT format (3) is linear w.r.t. the block core $\hat{X}^{(p)}$. We can write it using the frame matrix as follows,

$$x = \mathcal{X}_{\neq k}\hat{x}^{(k)}, \qquad X = \mathcal{X}_{\neq p}\hat{X}^{(p)}, \tag{6}$$

where the vector $x^{(k)}$ of size $r_{k-1}n_k r_k$ is a vectorization of the TT–core $X^{(k)}$,

$$x^{(k)} = \operatorname{vec} X^{(k)}, \qquad x^{(k)}(\overline{\beta_{k-1}j_k\beta_k}) = X^{(k)}_{\beta_{k-1},\beta_k}(j_k),$$

and the block TT–core $\hat{X}^{(p)} = [\hat{x}^{(p)}_b]$ is seen as a $r_{p-1}n_p r_p \times B$ matrix, which encapsulates vectorizations $\hat{x}^{(p)}_b = \operatorname{vec}\hat{X}^{(p)}(b)$ as columns.

The TT–core $\hat{X}^{(p)}$ can be also reshaped into a $r_{p-1}n_p \times Br_p$ matrix by arranging the indices in the form $\hat{X}^{(p)} = [\hat{X}^{(p)}(\overline{\alpha_{p-1}i_p}, \overline{b\alpha_p})]$, where the overline shows that the indices α_{p-1} and i_p are combined in a single multi-index, as well as b and α_p. We can write the low–rank decomposition of this matrix

$$\hat{X}^{(p)}_{\alpha_{p-1},\alpha_p}(i_p, b) = \hat{X}^{(p)}(\overline{\alpha_{p-1}i_p}, \overline{b\alpha_p}) = \sum_{\alpha'_p=1}^{r'_p} X^{(p)}_{\alpha_{p-1},\alpha'_p}(i_p)G_{\alpha'_p,\alpha_p}(b), \tag{7}$$

where the new matrix $G(b)$ of size $r'_p \times r_p$ appears, which carries the index b. Substituting (7) into (3) as follows,

$$x_b(i) = x_b(\overline{i_1 \dots i_d}) = X^{(1)}(i_1)\dots X^{(p)}(i_p)G(b)X^{(p+1)}(i_{p+1})\dots X^{(d)}(i_d), \tag{8}$$

we obtain the regular TT/MPS format for $X = [x_b(i)]$, where the index i is defined by d mode indices $i_1, \dots, i_d$ and b is considered as the additional index which is placed between i_p and i_{p+1}. This format with $G(b)$ in the rightmost position has been applied for the eigenproblem solution in [29], and is also considered in [41].

Using the decomposition (7), we can move the index b back and forth the tensor train. Indeed, if we multiply $G(b)$ by $X^{(p+1)}$ and reshape the result, we obtain $\hat{X}^{(p+1)}_b$ and recover (3) with the 'block' TT–core at the position $p+1$. We see that after this multiplication the TT–rank r_p is replaced by r'_p and in general $r'_p \neq r_p$. This means that the TT–ranks depend on the position p of the TT–core which carries the index b. The natural bound for the considered example is $r'_p \leqslant r_p B$, which technically allows the ranks to grow exponentially during the iterations. For the optimization algorithm to be efficient, special measures should be taken to keep the TT–ranks moderate.

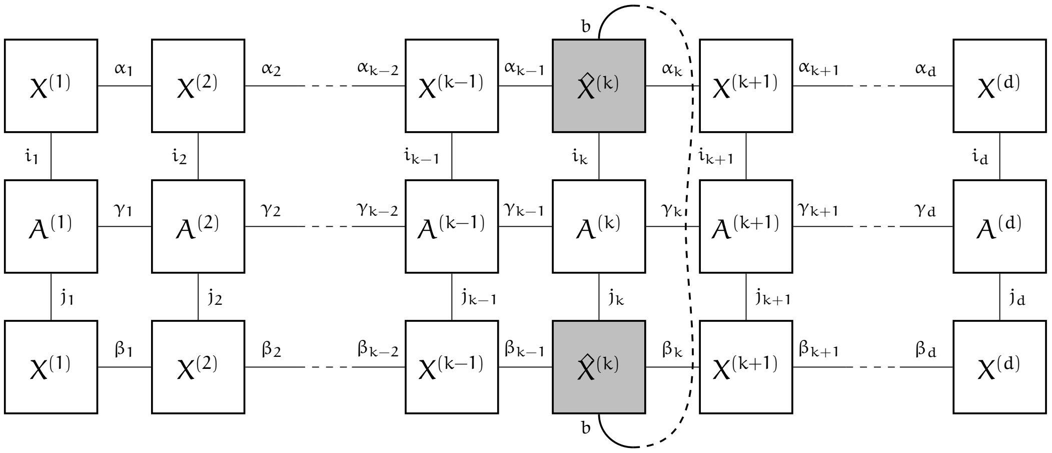

Figure 3: Tensor network corresponding to the block Rayleigh quotient (1) with the matrix A and vectors $X = [x_b]$ given in the tensor train format.

To reduce (truncate) the TT–rank, we can compute the low–rank decomposition (7) approximately. The approximation of the lowest rank within the prescribed accuracy level can be computed by the *singular value decomposition* (SVD). The perturbation introduced to $\hat{X}^{(p)}$ results in the perturbation to the whole matrix of eigenstates X, which can be amplified by the norm of the frame matrix $\mathcal{X}_{\neq p}$ due to the linearity (6). If $\mathcal{X}_{\neq p}^{*}\mathcal{X}_{\neq p} = I$, the Frobenius norms of the local and the global perturbations are the same. For a given tensor the orthogonality of the frame matrix can be achieved using the non–uniqueness of the TT format, by implying the *left–orthogonality* (left–normalization) constrains to $X^{(1)}, \dots, X^{(p-1)}$ and the *right–orthogonality* (right–normalization) to $X^{(p+1)}, \dots, X^{(d)}$.

Definition 2 (TT–orthogonality)**.** The TT–core $X^{(k)}$ is called *left–* or *right–orthogonal*, if, respectively,

$$\sum_{i_k} (X^{(k)}(i_k))^{*} X^{(k)}(i_k) = I, \qquad \sum_{i_k} X^{(k)}(i_k)(X^{(k)}(i_k))^{*} = I.$$

It can be shown [33, 41] that if $X^{(1)}, \dots, X^{(p-1)}$ are left-orthogonal, then the interface matrix $X^{<p}$ defined by (4) has orthonormal columns. Analogously, right-orthogonality of the TT-cores $X^{(p+1)}, \dots, X^{(d)}$ implies that the rows of $X^{>p}$ are orthonormal. Together these conditions provide the orthogonality of the frame matrix $\mathcal{X}_{\neq p}$ defined by (5), which allows the full accuracy control in the rank truncation step.

For a given TT format the TT–orthogonality can be implied constructively by the subsequent orthogonalization of the cores, see [33, 41]. The procedure is very cheap and requires only $\mathcal{O}(nr^3)$ operations for each TT–core to be normalized. This operation is never a bottleneck in our algorithms, so we always assume that for a chosen TT–core $\hat{X}^{(p)}$ the frame matrix is orthogonal, without going into the minor detail.

The TT format for a matrix is written as follows,

$$A(i,j) = A(\overline{i_1 \dots i_d}, \overline{j_1 \dots j_d}) = A^{(1)}(i_1, j_1) \dots A^{(d)}(i_d, j_d). \tag{9}$$

The multiplication of a matrix by a vector in the TT format has been discussed in detail in [42, 32]. The *tensor network* which represents the quadratic function in (1) is shown in Fig. 3.

3 Minimization of the block Rayleigh quotient in the block–TT format

We formulate the eigenproblem via the Rayleigh quotient optimization (1), which is restrictively large when d increases. To make the problem tractable, the minimization over the whole space is substituted by the optimization over the manifold of tensors in the TT-format. The most natural approach is the alternating least squares (ALS)-type algorithm. Let all the TT-cores except $X^{(p)}(i_p)$ be "frozen". Since the frame matrix is assumed to be orthogonal, the minimization problem (1) is reduced to a smaller problem

$$\hat{X}^{(p)}_{\star} = \arg\min_{\hat{X}^{(p)}} \operatorname{trace}((\hat{X}^{(p)})^{*}\mathcal{X}^{*}_{\neq p} A \mathcal{X}_{\neq p}\hat{X}^{(p)}), \qquad \text{s.t.} \quad (\hat{X}^{(p)})^{*}\hat{X}^{(p)} = I, \tag{10}$$

which is in fact equivalent to the standard block eigenproblem for a small matrix. This *local* problem is often called *one–dimensional* (one–site), since it corresponds to the particular mode p, and the number of unknowns is linear in the mode size n_p. It is natural to organize the optimization process into *sweeps*, see Alg. 1. In this way, the orthogonalization of the TT-cores is very cheap.

The matrix-by-vector product for the local problem scales as $\mathcal{O}(r^2 r_A^2)$ matvec(n) and $\mathcal{O}(r^3 n r_A)$ additional operations[1], where matvec(n_p) is a cost of each multiplication of a $n_p \times n_p$ matrix by a vector,

$$y(i_p) = \sum_{j_p=1}^{n} A^{(p)}_{\gamma_{p-1},\gamma_p}(i_p, j_p)x(j_p).$$

For applications in quantum chemistry and physics the mode size is usually not large, and $\mathcal{O}(n^2)$ complexity is acceptable. In applications of numerical linear algebra and scientific computing n can grow up to thousands, and the use of sparsity, whether possible, is essential to reduce the complexity to linear in n. The B extreme eigenvectors can be found using the block Krylov techniques (e.g. the notable LOBPCG method [26]) using $\mathcal{O}(B)$ local matvecs and $\mathcal{O}(B^2)$ orthogonalizations of vectors x_b. The low–rank approximation (7) can be done by the SVD in $\mathcal{O}(Br^3 n \min(B, n))$ operations, which can be the slowest part of the algorithm if n and B are both large. To speed up this step, we can use the cross interpolation of matrices [44] (for more robust implementation see [34, Alg. 3]), or incomplete Cholesky method applied to the Gram matrix, see e.g. [38]. Using this methods, we reduce the complexity to $\mathcal{O}(Br^3 n)$, that is to the level of other steps of the algorithm.

Summarizing the above, the overall storage and complexity of Alg. 1 are

$$\begin{aligned} \text{mem} &= \mathcal{O}((d+B)nr^2), \\ \text{work} &= \underbrace{\mathcal{O}(dBr^3 r_A n) + \mathcal{O}(dBr^2 r_A^2)\,\text{matvec}(n)}_{\text{local matvecs}} + \underbrace{\mathcal{O}(dB^2 r^2 n)}_{\text{orthogonalization}} + \underbrace{\mathcal{O}(dBr^3 n)}_{\text{truncation}}. \end{aligned} \tag{11}$$

[1] In complexity estimates we assume that all mode sizes are $\mathcal{O}(n)$, TT–ranks of vector X are $\mathcal{O}(r)$ and TT–ranks of A are $\mathcal{O}(r_A)$

Algorithm 1 ALS optimization for the block Rayleigh quotient in the TT format

```
Require: Matrix $A$ and initial guess $X$ in the TT–format (9) and (3), resp.
Ensure: Improved eigenvectors $X = [x_b]$ in the TT–format (3)
while stopping criterion is not satisfied do
    Move index $b$ to $\hat{X}^{(1)}(i_1, b)$ and make $X^{(2)}, \dots, X^{(d)}$ right–orthogonal
    for $p = 1, \dots, d-1$ do {Left–to–right half–sweep}
        Solve one–dimensional problem (10) and replace $\hat{X}^{(p)} := \hat{X}_\star^{(p)}$
        Decompose $\hat{X}^{(p)}$ by (7) with accuracy $\varepsilon$ and replace $r_p := r'_p$
        Ensure the left–orthogonality of $X^{(p)}$
        In (8), merge blocks $G(b)X^{(p+1)}(i_{p+1}) = \hat{X}^{(p+1)}(i_{p+1}, b)$
    end for
    Perform right–to–left half–sweep in the same way
end while
```

Typically the TT–rank of X grows with B at least as $r = \mathcal{O}(B)$, since the TT–format (3) has to represent B vectors with possibly different structures simultaneously. In this case the first term is more likely to dominate asymptotically for large B and r and we have work $= \mathcal{O}(dBr^3 r_A n)$.

The complexity of the proposed method can be compared with the one of the algorithms used in quantum physics community, e.g. the DMRG algorithm with *targeting* [47] and the *variational* numerical renormalization group (NRG) [36]. In the notation of this paper (see Table 1), the complexity of the DMRG reported in [36] is $\mathcal{O}(dBr^3n^3)$ and the complexity of the variational NRG is $\mathcal{O}(Br^3 + dr^3n^3)$. Note that the complexity of the DMRG method is cubic in n, since the optimization over two modes is used in each step. This is a usual way to couple the subspaces corresponding to different modes (sites) and increase (or decrease) the TT–rank (bond dimension) between the blocks. In our method the adaptation of the TT–rank r_k is done by the approximate decomposition (7) of the TT–block $\hat{X}^{(p)}$ which is seen as a $r_{p-1}n_p \times Br_{p+1}$ matrix.

The truncation step introduces the perturbation to the local vector $\hat{X}^{(p)}$. The orthogonality of the interface $\mathcal{X}_{\neq p}$ is essential in this step to control the accuracy of the vectors in X. The orthogonality between x_b may be also perturbed, but can be easily recovered by the reorthogonalization of $G(b) = [G_{\alpha'_p,\alpha_p}(b)]$ as a collection of n vectors of size $r'_p r_p$. In practice this step is not necessary, since the non–orthogonal $G(p)$ is accumulated into $\hat{X}^{(p+1)}$, which is replaced by the orthogonal set of the extreme eigenvectors of the local problem in the next step of Alg. 1.

Table 1: Correspondence between notation of our paper and notation used in quantum physics, e.g. [36]

TT notation		DMRG literature	
mode size	n	local dimension	d
dimension	d	number of sites	n
TT–ranks	r	bond dimension	D
# eigenvectors	B	# target states	M

4 Laplace operator (particle in a box)

We consider the eigenstates of a particle in a d–dimensional box and discretize this problem on a uniform tensor product grid with n elements in each direction as follows

$$-\Delta X = X\Lambda, \qquad \Delta = D \otimes I \otimes \cdots \otimes I + \cdots + I \otimes \cdots \otimes I \otimes D, \tag{12}$$

where I is the identity $n \times n$ matrix, and $D = \mathrm{tridiag}(1, -2, 1)$ is the standard finite difference discretization of the one-dimensional Laplace operator. This problem is a nice sanity test for eigensolvers, because the analytical solution of the eigenproblem is available. The eigenpairs $\{\mu_b, u_b\}$ of the matrix $(-D)$ read

$$\mu_b = 4\sin^2\left(\frac{\pi(b+1)}{2(n+1)}\right), \quad u_b(j) = \sin\left(\frac{\pi(b+1)(j+1)}{n+1}\right), \qquad b, j = 0, \ldots, n-1,$$

and for the high–dimensional problem (12) we obtain

$$\lambda_{\sigma(b_1,\ldots,b_d)} = \mu_{b_1} + \cdots + \mu_{b_d}, \quad x_{\sigma(b_1,\ldots,b_d)} = u_{b_1} \otimes \cdots \otimes u_{b_d}, \tag{13}$$

where the order function σ sorts the eigenvalues from low to high. Alternatively, we can enumerate the *energy levels* by multiindices as follows:

$$E_0 = \lambda_{\sigma(0,0,\ldots,0)}, \qquad E_1 = \lambda_{\sigma(1,0,\ldots,0,0)} = \ldots = \lambda_{\sigma(0,0,\ldots,0,1)},$$

and so on, and define the invariant eigenspaces X_k, which consist of the eigenvectors corresponding to the same eigenvalues. The excited eigenstates possess a strong multiplicity, which we show in Table 2, where $\lambda_0 = d\mu_0$ denotes the ground state energy. The multiplicity (or *degeneracy*) of the energy levels is a known issue for the convergence of eigensolvers, which makes this example particularly instructive to consider.

The matrix Δ is represented in the TT format with TT–ranks not larger than two, see [18]. Each eigenvector $x_{\sigma(b_1,\ldots,b_d)}$ has a rank-one decomposition by (13), and therefore B eigenstates are represented simultaneously in the block–TT format (3) with TT–ranks not larger than B. Since $\{u_b\}$ are orthogonal, the maximal TT–rank grows linearly with B (for small B it is easy to check this directly, but in general one would obtain a complicated combinatorial formula). Summarizing the above, we see that Alg. 1 can be applied to compute lower–lying eigenstates of (12) with $B \lesssim 100$.

We compare the proposed method with the deflation technique, which computes the eigenstates one–by–one using the standard DMRG algorithm. When

Table 2: Low–lying eigenvalues and dimensions of the corresponding invariant subspaces, Laplace example (12)

level k	$E_k - E_0$	multiplicity of X_k	example of eigenvector from X_k
0	0	1	$u_0 \otimes u_0 \otimes u_0 \otimes u_0 \otimes \ldots \otimes u_0$
1	$\mu_1 - \mu_0$	d	$u_1 \otimes u_0 \otimes u_0 \otimes u_0 \otimes \ldots \otimes u_0$
2	$2(\mu_1 - \mu_0)$	$d(d-1)/2$	$u_1 \otimes u_1 \otimes u_0 \otimes u_0 \otimes \ldots \otimes u_0$
3	$\mu_2 - \mu_0$	d	$u_2 \otimes u_0 \otimes u_0 \otimes u_0 \otimes \ldots \otimes u_0$
4	$3(\mu_1 - \mu_0)$	$d(d-1)(d-2)/6$	$u_1 \otimes u_1 \otimes u_1 \otimes u_0 \otimes \ldots \otimes u_0$

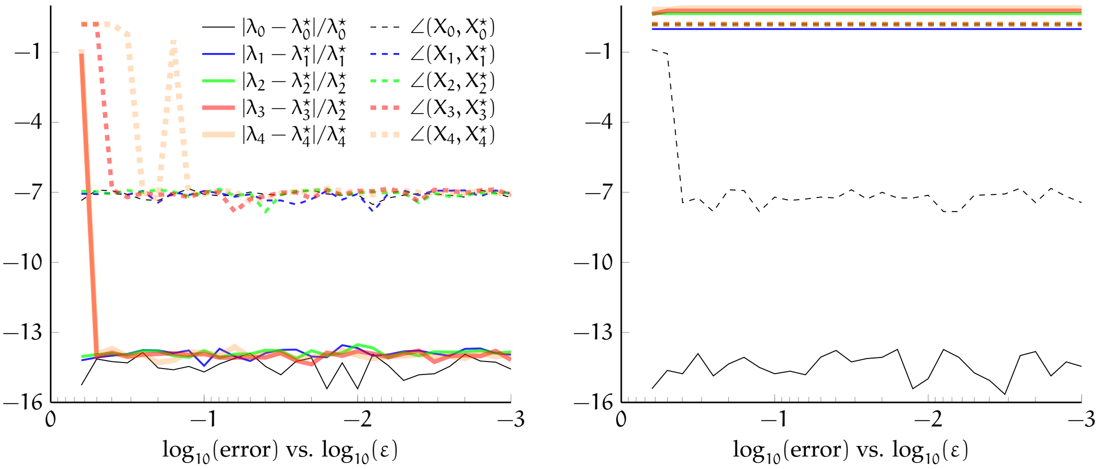

Figure 4: Errors in eigenvalues and vectors vs. the thruncation threshold for the Laplace example (12). Left: block-TT Alg. 1. Right: deflation method. The reference values $\lambda_k^\star$ and $X_k^\star$ are computed by (13). Parameters: dimension $d = 5$, mode size $n = 16$.

$B - 1$ eigenvectors $X = [x_b]$ are already computed, the Rayleigh quotient $x^* A x$ is minimized over the normalized vectors orthogonal to X. All vectors are represented simultaneously in the block–TT format (3), and the local problem is written as a *deflated* eigenvalue problem with $B - 1$ orthogonality constrains. We could expect the deflation method to be more efficient since the size of the variational problem is smaller. However, in contrary with the ordinary case, the DMRG combined with the deflation may converge to a wrong eigenpair, corresponding to a larger energy level.

We apply both methods to find $B = 30$ low–lying eigenstates of (12) in dimension $d = 5$. This should be enough for the subspaces $X_0, \ldots, X_3$, and find several vectors from X_4. We vary the truncation parameter ε and measure the accuracy of the computed eigenvalues comparing them with the analytical values (13). For X_k, we track the angle between the computed and analytical eigenspace,

$$\angle(X, Y) = \max \angle(x, y), \qquad x \in \operatorname{span} X, \quad y \in \operatorname{span} Y,$$

using the formula $\cos \angle(X, Y) = \sigma_{\min}(X^* Y)$, where $\sigma_{\min}$ is the smallest singular value.

The results are shown in Fig. 4. We see that all errors in the block method drop down to the machine precision rapidly[2]. The eigenvectors are represented exactly in the TT format as soon as the rank is large enough, even for very rough truncation threshold ε. We see that the block DMRG method computes all $B = 30$ eigenstates correctly. The deflation method determines accurately the ground state X_0 and two vectors from the first excited subspace X_1. All further eigenvectors belong to the subspaces higher than X_4, and information on the multiplicity and other intermediate states is not computed correctly. This shows that Alg. 1 is reliable in the case of large multiplicities, and therefore can be also applied to compute the eigenstates with small spectral gaps.

[2] Note that the minimal possible error for the angle is the square root of the machine precision due to the acos function

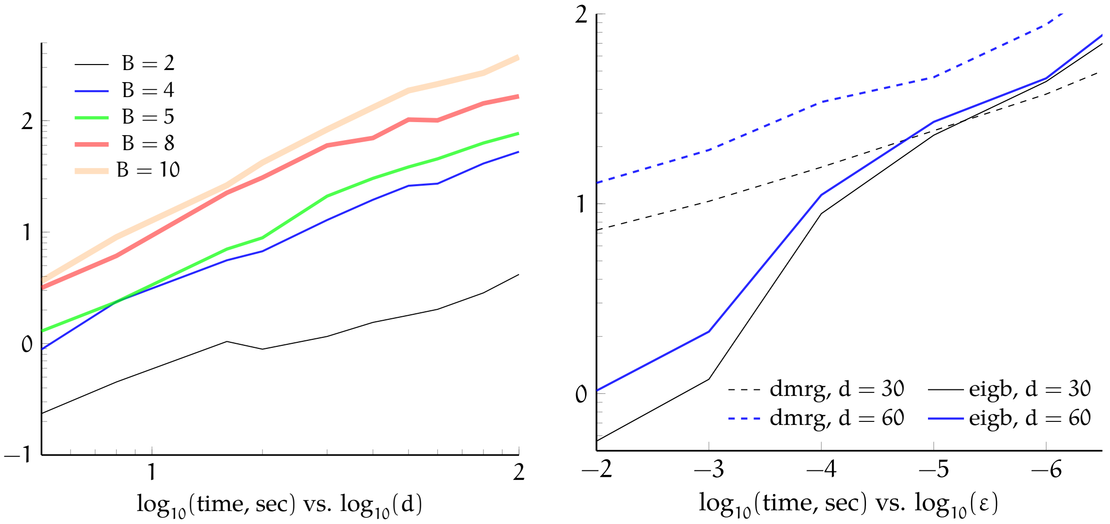

Figure 5: CPU time vs. dimension d (left) and truncation threshold ε (right) for the Henon-Heiles example (14). Parameters: $\varepsilon = 10^{-3}$ (left), $B = 2$ (right), $n = 28$.

5 Henon–Heiles potential

We consider a particle in a d–dimensional Henon–Heiles potential which is used as a benchmark for high-dimensional quantum molecular dynamics computations [30]. The wavefunction $\psi = \psi(q_1, \ldots, q_d)$ satisfies a stationary Schrödinger equation $H\psi = E\psi$, where the Hamiltonian is defined as follows

$$H = \overbrace{-\frac{1}{2}\Delta + \underbrace{\frac{1}{2}\sum_{k=1}^{d} q_k^2}}^{\text{harmonic part}} \underbrace{+ \overbrace{\lambda \sum_{k=1}^{d-1} \left(q_k^2 q_{k+1} - \frac{1}{3} q_{k+1}^3 \right)}^{\text{anharmonic part}}}_{\text{Henon-Heiles potential } V(q_1,\ldots,q_d)}, \tag{14}$$

where parameter $\lambda = 0.111803$ controls the anharmonic contribution to the potential.

The principal part of the Henon-Heiles operator describes a harmonic oscillator, whose eigenstates are products of a Gaussian by Hermite polynomials and have tensor rank 1. Therefore, for moderate anharmonicity λ the rank-1 basis of eigenfunctions of the harmonic oscillator is a good choice for the discretization of (14). The Galerkin discretization scheme results not only in dense stiffness and mass matrices, but also in a dense matrix that describes the action of the potential V. The DVR(discrete variable representation) scheme uses a collocation of V and ψ on the nodes of the Hermite polynomials and is known to provide the same order of accuracy for the eigenproblem as the Galerkin method.

The one-dimensional Laplace operator $D = -\frac{d^2}{dq^2}$ in the Hermite-DVR approach is discretized (see [2]) as follows

$$D_{ij} = \begin{cases} \frac{1}{6}(4n - 1 - 2t_i^2), & i = j, \\ (-1)^{(i-j)}(2(t_i - t_j)^{-2} - \frac{1}{2}), & i \neq j, \end{cases}$$

where $t_1, \ldots, t_n$ are the roots of the n-th Hermite polynomial, i.e. the *Hermite mesh*. The discretization of the d-dimensional Laplace operator is written in the same

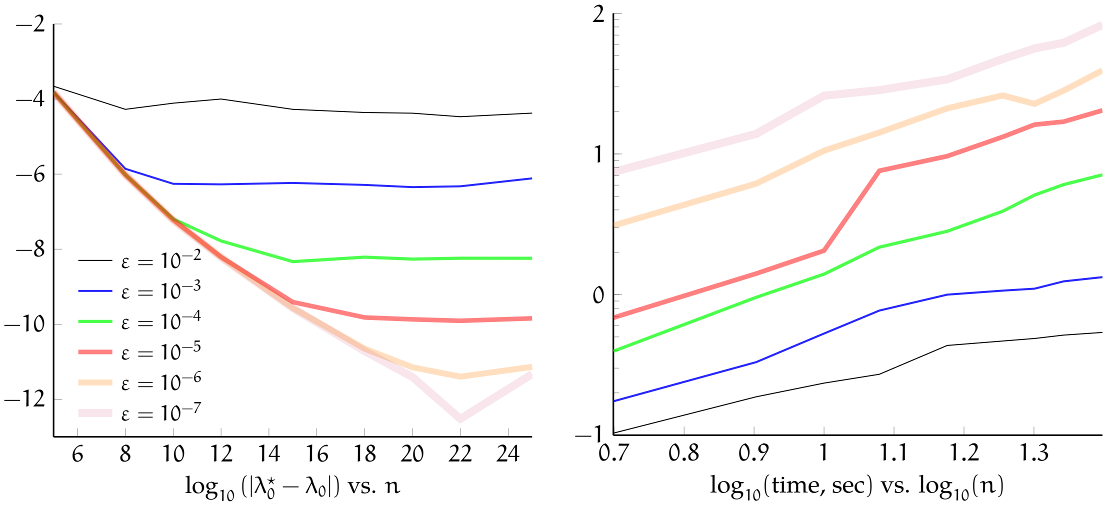

Figure 6: Eigenvalue errors (left) and CPU times (right) vs. mode size n and truncation threshold ε for the Henon-Heiles example (14). The reference value $\lambda^\star$ is computed by the same algorithm with $\varepsilon = 10^{-7}, n = 28$. Parameters: $B = 2$, $d = 30$

way as in (12) and has TT–ranks not larger than two. The potential V is constructed as a sum of rank-1 monomials and can be represented by the TT format with TT–ranks not larger than 7, see [25]. Therefore, Alg. 1 can be applied to find the low–lying eigenstates of (14).

As an initial guess, we take a random TT tensor of rank B in the block form (3). We test the block TT algorithm for different dimensions d, Frobenius-norm truncation tolerances for eigenvalues ε and number of eigenstates B. Numerical experiments show that the internal convergence of the eigenvalues is quadratic w.r.t. the truncation threshold ε, as expected from the perturbation theory. The errors in the eigenvalues do not grow with the dimension, so the method can be exploited for higher–dimensional systems. From Fig. 5 (left), we observe that the cost grows mildly with the dimension and B, which allows to compute energy levels with high accuracy.

In Fig. 5 (right) we compare our block-TT method (Alg. 1, referenced as 'eigb') with the DMRG technique from [25], intended for searching one lowest eigenpair only. Alg. 1 computes at least $B = 2$ eigenstates, and the TT-ranks of the block–TT format which represents two eigenstates are larger (sometimes significantly) than the TT-ranks of the ground state returned by the DMRG method. We see how large ranks of the excited state manifest themselves for high accuracies in Fig. 5. Despite the lower cost w.r.t. the mode size n, the 'eigb' method requires the same CPU time as the DMRG due to larger TT-ranks, which leads to a higher complexity of the local eigenproblem (10).

Finally we investigate the performance of Alg. 1 w.r.t. the mode size (number of Hermite polynomials) n, see Fig. 6. The error decreases exponentially with n due to the Hermite-DVR scheme, but only until the level $\mathcal{O}(\varepsilon^2)$ governed by the tensor truncation threshold. The CPU time growth rate is balanced between $\mathcal{O}(n)$ (truncation step) and $\mathcal{O}(n^2)$ (dense matrix-by-vector multiplication).

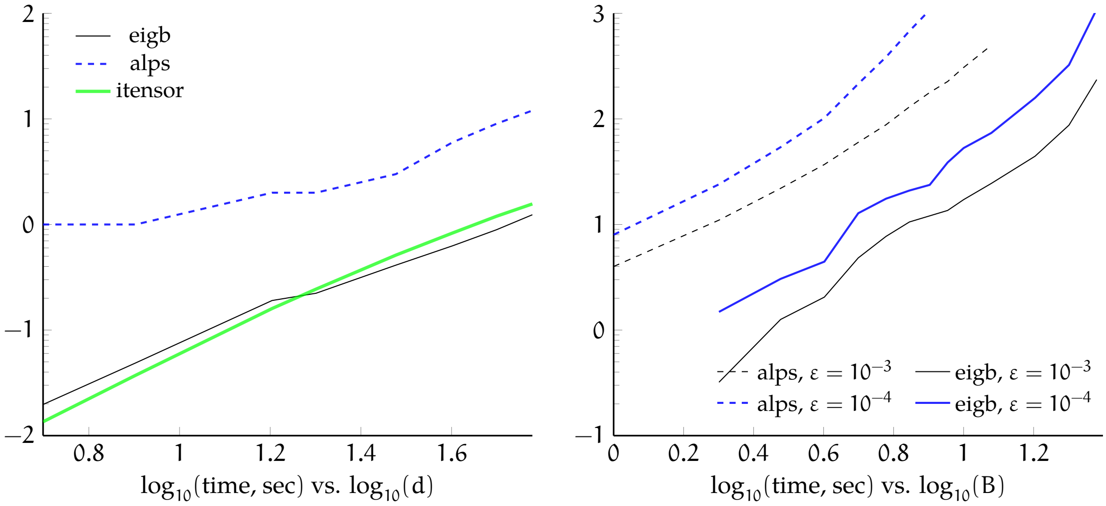

Figure 7: CPU times of different methods vs. d (left), B (right), Heisenberg example. Parameters: $\varepsilon = 10^{-3}$, $B = 1$ (left), $d = 30$ (right).

6 Heisenberg model

The one-dimensional Heisenberg model is one of the classical applications of the MPS/DMRG algorithms, so it is worth to test our approximate block eigenvalue solver on it, and compare it with the established software from the MPS community. This model describes the interaction of spins on a one-dimensional lattice. The Hamiltonian for the antiferromagnetic case is written in the following form

$$H = \sum_{i=1}^{d-1} \mathbf{S}_i \mathbf{S}_{i+1}, \tag{15}$$

where $\mathbf{S}_i$ is the spin operator. The product $\mathbf{S}_i \mathbf{S}_{i+1}$ can be written in terms of the spin components as follows

$$\mathbf{S}_i \mathbf{S}_{i+1} = (S_x)_i (S_x)_{i+1} + (S_y)_i (S_y)_{i+1} + (S_z)_i (S_z)_{i+1}. \tag{16}$$

Since the operator $(S_x)_i$ affects only the i-th spin, it has the form

$$(S_x)_i = I \otimes \ldots \otimes S_x \otimes \ldots \otimes I, \tag{17}$$

and similarly for $(S_y)_i$, $(S_z)_i$. The "elementary" spin operators S_x, S_y and S_z are the Pauli matrices, defined as follows

$$S_x = \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad S_y = \frac{1}{2}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad S_z = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{18}$$

Using the equations (15),(16),(17) and (18), it is very easy to construct the TT representation of the Heisenberg Hamiltonian. Moreover, it is also not difficult to show that its TT ranks are not larger than 5, see [5].

For different lengths of the spin chain we compute the lowest eigenvalues with the help of our block eigenvalue solver, and compare it with two packages containing the DMRG: ALPS (Algorithms and Libraries for Physics Simulations)[3] and

[3] http://alps.comp-phys.org/, Release 2.1.1

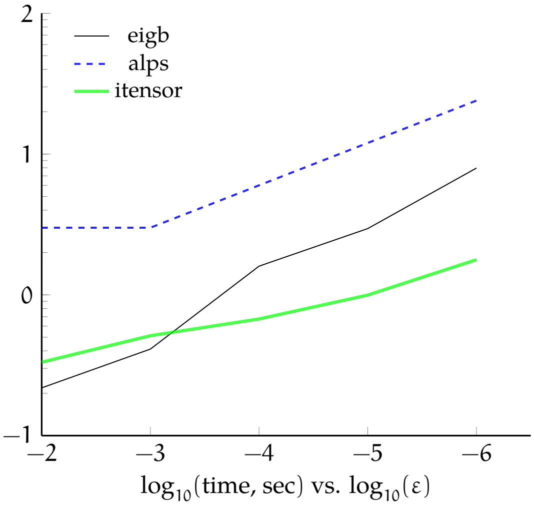

Figure 8: CPU times of different methods vs. ε, Heisenberg example. Parameters: $B = 1$, $d = 30$.

ITensor[4]. The ALPS allows to compute several excited states, whereas the ITensor is devised for targeting the ground state only. Using Alg. 1 we can compute $B \geqslant 2$ eigenstates, so the comparison with ITensor and ALPS with $B = 1$ assumes that our algorithm is applied with $B = 2$ and the excited state is computed but not used. To introduce a fair amount of optimization, all software was compiled using Intel C/Fortran compiler 2013 and linked with the optimized LAPACK/BLAS packages provided in the MKL library. The experiments have been computed using one Intel Xeon processor with 4 cores at 2GHz.

The computational times of Alg. 1 ('eigb'), ALPS and ITensor are presented in Figs 7 and 8. We observe that all methods manifest the polynomial complexity scaling in the number of spins d and targeting eigenstates B. The dependence on B is of particular interest. Recalling the complexity estimate (11), we may expect work $= \mathcal{O}(B^4)$, if the TT–ranks behave as $r \sim B$. In Fig. 7 we see even milder growth of the CPU time, time $= \mathcal{O}(B^\beta)$, $\beta \approx 2.5$, and the same phenomenon is observed for the DMRG method from the ALPS. It can be explained by the non-uniform distribution of the TT–ranks, which makes the complexity estimate (11) larger than the actual computation time.

Considering the memory usage, we observe that the ALPS package failed to compute more than 12 targets due to the Out-of-Memory exception, while our implementation of Alg. 1 goes readily beyond 20 eigenstates.

The accuracy parameter ε requires an additional comment, since it is used differently in different methods. The block TT method performs the Frobenius-norm ε-approximation of the eigenvectors, whereas the other two packages are designed to keep the accuracy of the eigenvalue, and hence are parametrized with ε^2 as a threshold.

We see that the performances of our block TT technique and the method from ITensor are comparable, and the algorithm from ALPS is significantly slower. Taking into account that our approach allows to compute several eigenstates (which seems to be not the case for ITensor), we may conclude that it overcomes the cur-

[4] `http://itensor.org/`, downloaded on May 20, 2013 (no particular release)

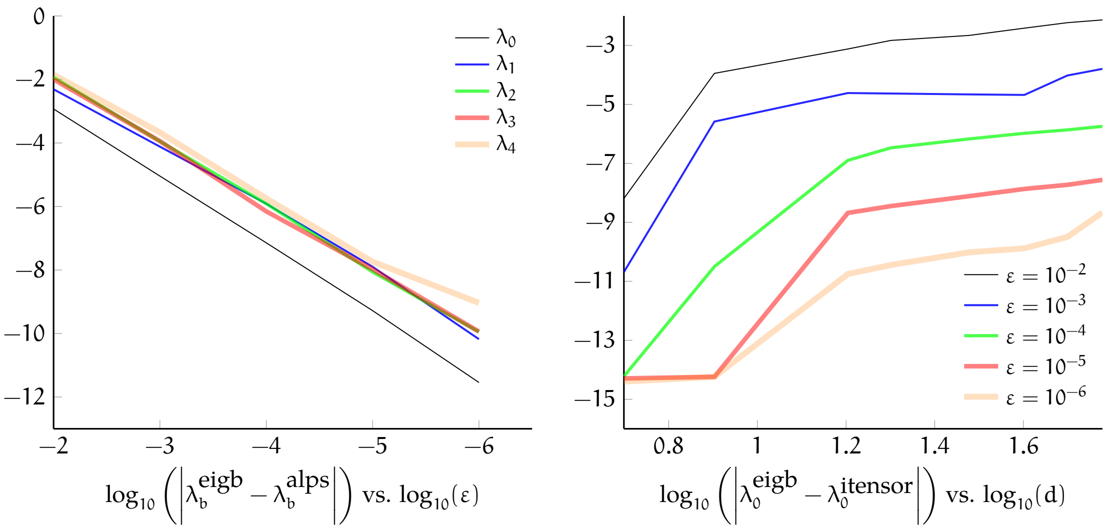

Figure 9: Discrepancies between eigenvalues in different methods vs. eigenvalue number (left) and d (right), Heisenberg example. Parameters: $d = 30$, $B = 5$ (left), $B = 1$ (right).

rent state-of-the-art software. To be sure that the correctness is maintained, we show the difference between the eigenvalues computed by the eigb and the other two methods in Fig. 9, which demonstrates that the $\mathcal{O}(\varepsilon^2)$ accuracy of the eigenvalues is achieved.

7 Conclusions

We propose the efficient one–site rank–adaptive algorithm for the computation of several extreme eigenvalues of a high–dimensional Hermitian matrix. The eigenstates are represented simultaneously by the block tensor train format, which is flexible and has a particular form in each variational step of the algorithm. The complexity of the proposed method is linear with respect to the mode size for sparse matrices and quadratic for dense matrices, which is better than the complexity of the DMRG algorithm. The proposed method has the same asymptotic in the number of eigenstates, the TT–ranks and the dimension, as the DMRG and NRG algorithms used in quantum physics.

The algorithm is verified on a number of representative problems. First, it is tested on the high–dimensional Laplace matrix, which describes the eigenstates of the particle in a box, and it is demonstrated that the algorithm is capable of finding the eigenstates with high multiplicity. Second, it is applied to the stationary Schrödinger equation with the Henon-Heiles potential, and it is shown that the proposed method can compute the ground state faster than the DMRG algorithm. Finally, we apply the algorithm to the Heisenberg spin chain and show that it is competitive with the state of the art publicly available DMRG implementations.

The framework developed in this paper can be applied to a wider class of problems, which are formulated via the subspace optimization.

Software implementation

We have implemented the main Algorithm 1 in Fortran with interfaces to Python and MATLAB. The codes are available online at

- `http://github.com/oseledets/tt-fort`
- `http://github.com/oseledets/ttpy`
- `http://github.com/oseledets/TT-Toolbox`

References

[1] A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings, A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids, *Journal of Non-Newtonian Fluid Mechanics*, 139(3):153 – 176, 2006. doi: 10.1016/j.jnnfm.2006.07.007.

[2] D. Baye and P.-H. Heenen, Generalised meshes for quantum mechanical problems, *J. Phys. A: Math. Gen.*, 19(11):2041–2059, 1986. doi: 10.1088/0305-4470/19/11/013.

[3] L. de Lathauwer, B. de Moor, and J. Vandewalle, A multilinear singular value decomposition, *SIAM J. Matrix Anal. Appl.*, 21:1253–1278, 2000. doi: 10.1137/s0895479896305696.

[4] S. Dolgov and B. Khoromskij, Two-level QTT-Tucker format for optimized tensor calculus, *SIAM J. on Matrix An. Appl.*, 34(2):593–623, 2013. doi: 10.1137/120882597.

[5] S. V. Dolgov and B. N. Khoromskij, Tensor-product approach to global time-space-parametric discretization of chemical master equation, Preprint 68, MPI MIS, 2012. `http://www.mis.mpg.de/preprints/2012/preprint2012_68.pdf`.

[6] S. V. Dolgov, B. N. Khoromskij, and I. V. Oseledets, Fast solution of multi-dimensional parabolic problems in the tensor train/quantized tensor train–format with initial application to the Fokker-Planck equation, *SIAM J. Sci. Comput.*, 34(6):A3016–A3038, 2012. doi: 10.1137/120864210.

[7] S. V. Dolgov, B. N. Khoromskij, and D. V. Savostyanov, Superfast Fourier transform using QTT approximation, *J. Fourier Anal. Appl.*, 18(5):915–953, 2012. doi: 10.1007/s00041-012-9227-4.

[8] S. V. Dolgov and I. V. Oseledets, Solution of linear systems and matrix inversion in the TT-format, *SIAM J. Sci. Comput.*, 34(5):A2718–A2739, 2012. doi: 10.1137/110833142.

[9] S. V. Dolgov and D. V. Savostyanov, Alternating minimal energy methods for linear systems in higher dimensions. Part I: SPD systems, arXiv preprint 1301.6068, 2013. `http://arxiv.org/abs/1301.6068`.

[10] ——, Alternating minimal energy methods for linear systems in higher dimensions. Part II: Faster algorithm and application to nonsymmetric systems, arXiv preprint 1304.1222, 2013. `http://arxiv.org/abs/1304.1222`.

[11] H.-J. Flad, B. N. Khoromskij, D. V. Savostyanov, and E. E. Tyrtyshnikov, Verification of the cross 3D algorithm on quantum chemistry data, *Rus. J. Numer. Anal. Math. Model.*, 23(4):329–344, 2008. doi: 10.1515/RJNAMM.2008.020.

[12] L. Grasedyck, Hierarchical singular value decomposition of tensors, *SIAM J. Matrix Anal. Appl.*, 31(4):2029–2054, 2010.

[13] W. Hackbusch, Tensorisation of vectors and their efficient convolution, *Numer. Math.*, 119(3):465–488, 2011. doi: 10.1007/s00211-011-0393-0.

[14] W. Hackbusch, B. N. Khoromskij, S. A. Sauter, and E. E. Tyrtyshnikov, Use of tensor formats in elliptic eigenvalue problems, *Numer. Linear Algebra Appl.*, 19(1):133–151, 2012. doi: 10.1002/nla.793.

[15] W. Hackbusch and S. Kühn, A new scheme for the tensor representation, *J. Fourier Anal. Appl.*, 15(5):706–722, 2009.

[16] S. Holtz, T. Rohwedder, and R. Schneider, The alternating linear scheme for tensor optimization in the tensor train format, *SIAM J. Sci. Comput.*, 34(2):A683–A713, 2012. doi: 10.1137/100818893.

[17] V. Kazeev, B. N. Khoromskij, and E. E. Tyrtyshnikov, Multilevel Toeplitz matrices generated by tensor-structured vectors and convolution with logarithmic complexity, Tech. Rep. 36, MPI MIS, Leipzig, 2011. `http://www.mis.mpg.de/publications/preprints/2011/prepr2011-36.html`.

[18] V. A. Kazeev and B. N. Khoromskij, Low-rank explicit QTT representation of the Laplace operator and its inverse, *SIAM J. Matrix Anal. Appl.*, 33(3):742–758, 2012. doi: 10.1137/100820479.

[19] V. Khoromskaia, B. N. Khoromskij, and R. Schneider, Tensor-structured factorized calculation of two-electron integrals in a general basis, *SIAM J Sci Comp*, 35(2):A987–A1010, 2013. doi: 10.1137/120884067.

[20] B. N. Khoromskij, Fast and accurate tensor approximation of multivariate convolution with linear scaling in dimension, *J. Comp. Appl. Math.*, 234(11):3122–3139, 2010. doi: 10.1016/j.cam.2010.02.004.

[21] ——, $\mathcal{O}(d \log n)$–Quantics approximation of N–d tensors in high-dimensional numerical modeling, *Constr. Appr.*, 34(2):257–280, 2011. doi: 10.1007/s00365-011-9131-1.

[22] B. N. Khoromskij and V. Khoromskaia, Low rank Tucker-type tensor approximation to classical potentials, *Central European journal of mathematics*, 5(3):523–550, 2007. doi: 10.2478/s11533-007-0018-0.

[23] ——, Multigrid accelerated tensor approximation of function related multidimensional arrays, *SIAM J. Sci. Comput.*, 31(4):3002–3026, 2009. doi: 10.1137/080730408.

[24] B. N. Khoromskij, V. Khoromskaia, and H.-J. Flad., Numerical solution of the Hartree–Fock equation in multilevel tensor-structured format, *SIAM J. Sci. Comput.*, 33(1):45–65, 2011. doi: 10.1137/090777372.

[25] B. N. Khoromskij and I. V. Oseledets, DMRG+QTT approach to computation of the ground state for the molecular Schrödinger operator, Preprint 69, MPI MIS, Leipzig, 2010. `www.mis.mpg.de/preprints/2010/preprint2010_69.pdf`.

[26] A. V. Knyazev, Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method, *SIAM J. Sci. Comput.*, 23(2):517–541, 2001. doi: 10.1137/S1064827500366124.

[27] T. G. Kolda and B. W. Bader, Tensor decompositions and applications, *SIAM Review*, 51(3):455–500, 2009. doi: 10.1137/07070111X.

[28] C. Le Bris, T. Leliévre, and Y. Maday, Results and questions on a nonlinear approximation approach for solving high-dimensional partial differential equations, *Constr. Approx.*, 30:621–651, 2009. doi: 10.1007/s00365-009-9071-1.

[29] O. S. Lebedeva, Tensor conjugate-gradient-type method for Rayleigh quotient minimization in block QTT-format, *Russ. J. Numer. Anal. Math. Modelling*, 26(5):465–489, 2011. doi: 10.1515/rjnamm.2011.026.

[30] M. Nest and H.-D. Meyer, Benchmark calculations on high-dimensional Henon-Heiles potentials with the multi-configuration time dependent Hartree (MCTDH) method, *J. Chem. Phys.*, 117(23):10499, 2002. doi: 10.1063/1.1521129.

[31] I. V. Oseledets, Approximation of $2^d \times 2^d$ matrices using tensor decomposition, *SIAM J. Matrix Anal. Appl.*, 31(4):2130–2145, 2010. doi: 10.1137/090757861.

[32] ——, DMRG approach to fast linear algebra in the TT–format, *Comput. Meth. Appl. Math*, 11(3):382–393, 2011.

[33] ——, Tensor-train decomposition, *SIAM J. Sci. Comput.*, 33(5):2295–2317, 2011. doi: 10.1137/090752286.

[34] I. V. Oseledets, D. V. Savostyanov, and E. E. Tyrtyshnikov, Cross approximation in tensor electron density computations, *Numer. Linear Algebra Appl.*, 17(6):935–952, 2010. doi: 10.1002/nla.682.

[35] I. V. Oseledets and E. E. Tyrtyshnikov, Breaking the curse of dimensionality, or how to use SVD in many dimensions, *SIAM J. Sci. Comput.*, 31(5):3744–3759, 2009. doi: 10.1137/090748330.

[36] I. Pižorn and F. Verstraete, Variational Numerical Renormalization Group: Bridging the gap between NRG and Density Matrix Renormalization Group, *Phys. Rev. Lett.*, 108(067202), 2012. doi: 10.1103/PhysRevLett.108067202.

[37] T. Rohwedder and A. Uschmajew, Local convergence of alternating schemes for optimization of convex problems in the TT format, *SIAM J Num. Anal.*, 51(2):1134–1162, 2013. doi: 10.1137/110857520.

[38] D. V. Savostyanov, Fast revealing of mode ranks of tensor in canonical format, *Numer. Math. Theor. Meth. Appl.*, 2(4):439–444, 2009. doi: 10.4208/nmtma.2009.m9006s.

[39] ——, Quasioptimality of maximum–volume cross interpolation of tensors, arXiv preprint 1305.1818, 2013. `http://arxiv.org/abs/1305.1818`.

[40] D. V. Savostyanov and I. V. Oseledets, Fast adaptive interpolation of multidimensional arrays in tensor train format, in Proceedings of 7th International Workshop on Multidimensional Systems (nDS), IEEE, 2011. doi: 10.1109/nDS.2011.6076873.

[41] U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, *Annals of Physics*, 326(1):96–192, 2011. doi: 10.1016/j.aop.2010.09.012.

[42] U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, *Ann. Phys*, 326(1):96–192, 2011.

[43] V. Temlyakov, Greedy Approximation, Cambridge University Press, 2011.

[44] E. E. Tyrtyshnikov, Incomplete cross approximation in the mosaic–skeleton method, *Computing*, 64(4):367–380, 2000. doi: 10.1007/s006070070031.

[45] S. R. White, Density matrix formulation for quantum renormalization groups, *Phys. Rev. Lett.*, 69(19):2863–2866, 1992. doi: 10.1103/PhysRevLett.69.2863.

[46] ——, Density-matrix algorithms for quantum renormalization groups, *Phys. Rev. B*, 48(14):10345–10356, 1993. doi: 10.1103/PhysRevB.48.10345.

[47] S. R. White and D. Huse, Numerical renormalization-group study of low-lying eigenstates of the antiferromagnetic $S = 1$ Heisenberg chain, *Phys. Rev. B*, 48(6):3844–3852, 1993. doi: 10.1103/PhysRevB.48.3844.