

LOW-RANK TENSOR METHODS WITH SUBSPACE CORRECTION FOR SYMMETRIC EIGENVALUE PROBLEMS*

DANIEL KRESSNER[†], MICHAEL STEINLECHNER[†], AND ANDRÉ USCHMAJEW[†]

Abstract. We consider the solution of large-scale symmetric eigenvalue problems for which it is known that the eigenvectors admit a low-rank tensor approximation. Such problems arise, for example, from the discretization of high-dimensional elliptic PDE eigenvalue problems or in strongly correlated spin systems. Our methods are built on imposing low-rank (block) tensor train (TT) structure on the trace minimization characterization of the eigenvalues. The common approach of alternating optimization is combined with an enrichment of the TT cores by (preconditioned) gradients, as recently proposed by Dolgov and Savostyanov for linear systems. This can equivalently be viewed as a subspace correction technique. Several numerical experiments demonstrate the performance gains from using this technique.

Key words. ALS, DMRG, high-dimensional eigenvalue problems, LOBPCG, low-rank tensor methods, trace minimization, tensor train format

AMS subject classifications. 15A18, 15A69, 65F15, 65K10

DOI. 10.1137/130949919

1. Introduction. The computation of the p smallest eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_p$ for a symmetric matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ can be posed as the minimization problem

$$(1.1) \quad \min \{ \text{trace}(\mathbf{U}^T \mathbf{A} \mathbf{U}) : \mathbf{U} \in \mathbb{R}^{N \times p}, \mathbf{U}^T \mathbf{U} = \mathbf{I}_p \}.$$

The minimum is given by $\lambda_1 + \lambda_2 + \dots + \lambda_p$, with a minimizer \mathbf{U}_* composed of the corresponding eigenvectors. This formulation has been the basis of several established eigenvalue solvers; see [2, 11, 38, 39] for examples.

The required computational cost of traditional methods for solving an $N \times N$ eigenvalue problem scales at least linearly with N , which makes them unfeasible for treating very large values of N , say $N = 10^{42}$. During the last years, techniques based on low-rank tensor approximations have been developed to overcome this limitation for several classes of problems; see [15] for an overview. The common approach is to exploit an underlying tensor product structure

$$\mathbb{R}^N \cong \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d},$$

which may arise from, e.g., the structured discretization of a high-dimensional PDE on a tensor product domain. This allows us to regard a vector $\mathbf{u} \in \mathbb{R}^N$ as an $n_1 \times \dots \times n_d$ tensor with each size n_1, \dots, n_d much smaller than N . Under certain conditions, this tensor can be very well approximated in a low-rank tensor format, i.e., by a tensor allowing for a low-dimensional parametrization. Several such formats have been proposed, in particular the tensor train (TT) [33, 34] and the hierarchical Tucker [14, 19]

*Submitted to the journal's Methods and Algorithms for Scientific Computing section December 20, 2013; accepted for publication (in revised form) June 30, 2014; published electronically October 14, 2014.

<http://www.siam.org/journals/sisc/36-5/94991.html>

[†]MATHICSE-ANCHP, Section de Mathématiques, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland (daniel.kressner@epfl.ch, michael.steinlechner@epfl.ch, andre.uschmajew@epfl.ch). The work of the second author has been supported by the SNSF research module *Riemannian optimization for solving high-dimensional problems with low-rank tensor techniques* within the SNSF ProDoc *Efficient Numerical Methods for Partial Differential Equations*.

formats. These techniques can be employed for eigenvalue computation by, e.g., restricting the admissible set of the optimization problem (1.1) to such a low-rank format. The resulting (highly nonlinear) optimization problem can be addressed by alternating block optimization techniques. This approach has been successfully used in the context of the density matrix renormalization group (DMRG) for simulating strongly correlated spin systems; see, e.g., the survey [40] and the references therein. In the numerical analysis community, related ideas have been considered in [6, 26, 29]. We mention in passing that another tensor-based approach to eigenvalue computation consists of combining a standard iterative solver (such as the Lanczos or the locally optimal block preconditioned conjugate gradient (LOBPCG) methods) with repeated low-rank compression [18, 22, 27, 29, 30]; see [15, section 3.1] for further references. Low-rank matrix and tensor techniques have also been used successfully for performing large-scale electronic structure calculations in computational quantum chemistry; see [5, 20, 25, 24] for examples.

In this work, we propose a novel low-rank tensor method that allows for the computation of several smallest eigenvalues, offers rank adaptivity, and the possibility of incorporating preconditioners. For this purpose, we combine and extend several existing techniques. First, the block TT format proposed by Dolgov et al. [6], which has previously been considered in the context of Wilson's numerical renormalization group [35, 41], is used for representing an orthonormal basis of p vectors. The approach for solving (1.1) considered in [6, 35] consists of cyclically optimizing each core of the block TT format, after shifting and merging the index enumerating the eigenvalues to this core. This makes the method different from alternating block optimization approaches, such as alternating linear scheme (ALS) [21, 29], and allows for rank adaptivity if $p > 1$. Second, we combine this idea with locally enriching each core based on projected gradient information. Such an enrichment was initially proposed by White [43] for eigenvalue problems, and extended to symmetric as well as nonsymmetric linear systems by Dolgov and Savostyanov [8, 9]. Enrichment allows for rank adaptivity even in the case $p = 1$. Preconditioners, which arise naturally in the context of PDE eigenvalue problems, can be easily incorporated by using the preconditioned gradient instead of the gradient.

2. The TT format. We recall the tensor train (TT) format for a single vector $\mathbf{u} \in \mathbb{R}^{n_1 n_2 \cdots n_d}$, when regarded as an $n_1 \times n_2 \times \cdots \times n_d$ tensor, the block TT format for a matrix $\mathbf{U} \in \mathbb{R}^{n_1 n_2 \cdots n_d \times p}$, when regarded as a collection of p such tensors, and the operator TT format for a linear operator acting on $\mathbb{R}^{n_1 n_2 \cdots n_d}$. The description will be rather brief; see [6, 33, 37] for further details.

2.1. The TT format for a single vector. Given a column vector $\mathbf{u} \in \mathbb{R}^{n_1 n_2 \cdots n_d}$, we can access its entries using multi-indices

$$u_{i_1, i_2, \dots, i_d}, \quad 1 \leq i_\mu \leq n_\mu, \quad \mu = 1, 2, \dots, d.$$

Here and in the following, it is assumed that multi-indices of the form (i_1, i_2, \dots, i_d) are traversed in *reverse lexicographical order*.

In the *TT format*, the vector \mathbf{u} is represented entrywise as

$$(2.1) \quad u_{i_1, i_2, \dots, i_d} = U_1(i_1)U_2(i_2) \cdots U_d(i_d),$$

where each $U_\mu(i_\mu)$ is a matrix of size $r_{\mu-1} \times r_\mu$ for $i_\mu = 1, 2, \dots, n_\mu$. By definition, $r_0 = r_d = 1$. The third-order tensors \mathbf{U}_μ of size $r_{\mu-1} \times n_\mu \times r_\mu$ whose slices are the matrices $U_\mu(i_\mu)$, $i_\mu = 1, 2, \dots, n_\mu$, form the building blocks of the TT format and

are called the *TT cores*. In principle, any explicitly given vector \mathbf{u} can be written or approximated in the form (2.1) for sufficiently large ranks r_μ using the TT-SVD algorithm [33]. However, to be able to cope with large d , it is important to completely avoid the explicit representation of \mathbf{u} and perform all operations within the format.

If all ranks r_μ remain modest, the complexity for storing and performing operations with the vector \mathbf{u} in the TT format drops significantly. Unfortunately, a priori statements about the size of the numerical ranks are generally difficult, and we do not dwell upon this problem in this paper. There are several applications for which the numerical ranks have been observed to remain reasonably small, while maintaining an acceptable approximation error at the same time.

2.1.1. Extraction of one core. For practical and theoretical purposes, we require various reformulations of the decomposition (2.1). Given a TT core $\mathbf{U}_\mu \in \mathbb{R}^{r_{\mu-1} \times n_\mu \times r_\mu}$, we define the *left* and the *right unfoldings*,

$$\mathbf{U}_\mu^L \in \mathbb{R}^{r_{\mu-1} n_\mu \times r_\mu} \quad \text{and} \quad \mathbf{U}_\mu^R \in \mathbb{R}^{r_{\mu-1} \times n_\mu r_\mu}$$

by reshaping \mathbf{U}_μ into $r_{\mu-1} n_\mu \times r_\mu$ and $r_{\mu-1} \times n_\mu r_\mu$ matrices, respectively, according to our choice of reverse lexicographical ordering. Also, we need the *interface matrices*

$$\begin{aligned} \mathbf{U}_{\leq \mu} &= [U_1(i_1) U_2(i_2) \cdots U_\mu(i_\mu)] \in \mathbb{R}^{n_1 n_2 \cdots n_\mu \times r_\mu}, \\ \mathbf{U}_{\geq \mu} &= [U_\mu(i_\mu) U_{\mu+1}(i_{\mu+1}) \cdots U_d(i_d)]^T \in \mathbb{R}^{n_\mu n_{\mu+1} \cdots n_d \times r_{\mu-1}}. \end{aligned}$$

These interface matrices can be seen as the left and right parts of a chain of TT cores represented by (2.1), see Figure 1. The following recursive relations hold:

$$(2.2) \quad \mathbf{U}_{\leq \mu} = (I_{n_\mu} \otimes \mathbf{U}_{\leq \mu-1}) \mathbf{U}_\mu^L, \quad \text{and} \quad \mathbf{U}_{\geq \mu}^T = \mathbf{U}_\mu^R (\mathbf{U}_{\geq \mu+1}^T \otimes I_{n_\mu}),$$

where \otimes denotes the standard Kronecker product. Using these formulas, we obtain for every $\mu = 1, 2, \dots, d$ that

$$(2.3) \quad \begin{aligned} \mathbf{u} &= \text{vec}(\mathbf{U}_{\leq \mu-1} \mathbf{U}_{\geq \mu}^T) = \text{vec}(\mathbf{U}_{\leq \mu-1} \mathbf{U}_\mu^R (\mathbf{U}_{\geq \mu+1}^T \otimes I_{n_\mu})) \\ &= (\mathbf{U}_{\geq \mu+1} \otimes I_{n_\mu} \otimes \mathbf{U}_{\leq \mu-1}) \text{vec}(\mathbf{U}_\mu^R), \end{aligned}$$

where we assume the convention $\mathbf{U}_{\leq 0} = \mathbf{U}_{\geq d+1} = 1$. This relation will allow us to extract or insert individual cores. Introducing

$$(2.4) \quad \mathcal{U}_{\neq \mu} = \mathbf{U}_{\geq \mu+1} \otimes I_{n_\mu} \otimes \mathbf{U}_{\leq \mu-1},$$

we have the memorable formula

$$(2.5) \quad \mathbf{u} = \mathcal{U}_{\neq \mu} \cdot \text{vec}(\mathbf{U}_\mu^R).$$

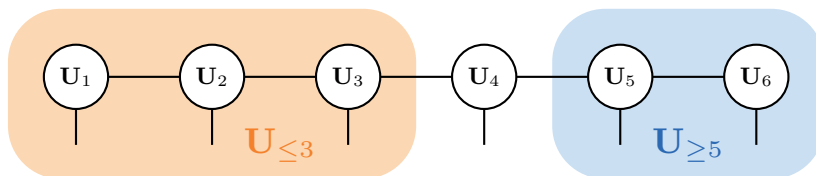


FIG. 1. Graphical representation of a TT tensor of order 6 and two of its interface matrices.

In terms of tensor spaces, we can interpret (2.3)–(2.5) as follows: For every $\mu = 1, 2, \dots, d$ it holds

$$\mathbf{u} \in \text{ran}(\mathbf{U}_{\geq \mu}) \otimes \text{ran}(\mathbf{U}_{\leq \mu-1}) \subseteq \text{ran}(\mathbf{U}_{\geq \mu+1}) \otimes \mathbb{R}^{n_\mu} \otimes \text{ran}(\mathbf{U}_{\leq \mu-1}).$$

Moreover, the subspace of all vectors in TT format having the same cores \mathbf{U}_ν at $\nu \neq \mu$ but an arbitrary core (of compatible size) at μ is given by

$$\text{ran}(\mathcal{U}_{\neq \mu}) = \text{ran}(\mathbf{U}_{\geq \mu+1}) \otimes \mathbb{R}^{n_\mu} \otimes \text{ran}(\mathbf{U}_{\leq \mu-1}).$$

2.1.2. Extraction of two neighboring cores. In our algorithms, it will also be important to extract two neighboring cores. Using the recursive relation also for $\mathbf{U}_{\geq \mu+1}$ in the middle term of (2.3) we get for every $\mu = 1, \dots, d-1$ that

$$(2.6) \quad \mathbf{u} = \mathcal{U}_{\neq \mu, \mu+1} \cdot \text{vec}(\mathbf{U}_\mu^L \mathbf{U}_{\mu+1}^R)$$

with

$$(2.7) \quad \mathcal{U}_{\neq \mu, \mu+1} = \mathbf{U}_{\geq \mu+2} \otimes I_{n_{\mu+1}} \otimes I_{n_\mu} \otimes \mathbf{U}_{\leq \mu-1}.$$

In terms of tensor spaces, this means

$$(2.8) \quad \mathbf{u} \in \text{ran}(\mathcal{U}_{\neq \mu, \mu+1}) = \text{ran}(\mathbf{U}_{\geq \mu+2}) \otimes \mathbb{R}^{n_{\mu+1}} \otimes \mathbb{R}^{n_\mu} \otimes \text{ran}(\mathbf{U}_{\leq \mu-1}).$$

For later reference we note that combining (2.7), (2.2), and (2.4) gives

$$(2.9) \quad \mathcal{U}_{\neq \mu, \mu+1} (I_{r_{\mu+1} n_{\mu+1}} \otimes \mathbf{U}_\mu^L) = \mathcal{U}_{\neq \mu+1}.$$

2.1.3. Orthonormalized TT formats. The representation of a vector \mathbf{u} in the TT format (2.1) is not unique. Given a decomposition $\mathbf{U}_\mu^L = QR$, the substitutions

$$(2.10) \quad \mathbf{U}_\mu^L \leftarrow Q, \quad \mathbf{U}_{\mu+1}^R \leftarrow R \mathbf{U}_{\mu+1}^R$$

do not change \mathbf{u} . In the case that Q has orthonormal rows, this is called the *left orthonormalization* of the μ th core. Similarly, the *right orthonormalization* of a core is obtained by decomposing $\mathbf{U}_\mu^R = (QR)^T = R^T Q^T$, and substituting

$$(2.11) \quad \mathbf{U}_\mu^R \leftarrow Q^T, \quad \mathbf{U}_{\mu-1}^L \leftarrow \mathbf{U}_{\mu-1}^L R^T.$$

Orthonormalization is an essential ingredient of algorithms for the TT format. When focusing on the μ th core, we usually assume that all cores to the left are left orthonormal, and all cores to the right are right orthonormal. This then implies that $\mathbf{U}_{\leq \mu-1}$ has orthonormal columns and that $\mathbf{U}_{\geq \mu+1}$ has orthonormal rows. Consequently, the matrices $\mathcal{U}_{\neq \mu}$, $\mathcal{U}_{\neq \mu, \mu+1}$ from (2.4) and (2.7) have orthonormal columns:

$$(2.12) \quad \mathcal{U}_{\neq \mu}^T \mathcal{U}_{\neq \mu} = I_{r_{\mu-1} n_\mu r_\mu}, \quad \mathcal{U}_{\neq \mu, \mu+1}^T \mathcal{U}_{\neq \mu, \mu+1} = I_{r_{\mu-1} n_\mu n_{\mu+1} r_{\mu+1}}.$$

2.2. The block TT format for a collection of vectors. When computing $p > 1$ eigenvectors, we need to work with p vectors $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p \in \mathbb{R}^{n_1 n_2 \dots n_d}$ simultaneously. Instead of representing each vector individually in the TT format, we will represent them jointly in a block TT format, following [6]. For this purpose, we fix an arbitrary position μ and use the representation

$$(2.13) \quad (u_\alpha)_{i_1, i_2, \dots, i_d} = U_1(i_1) \cdots U_{\mu-1}(i_{\mu-1}) U_{\mu, \alpha}(i_\mu) U_{\mu+1}(i_{\mu+1}) \cdots U_d(i_d), \quad 1 \leq \alpha \leq p.$$

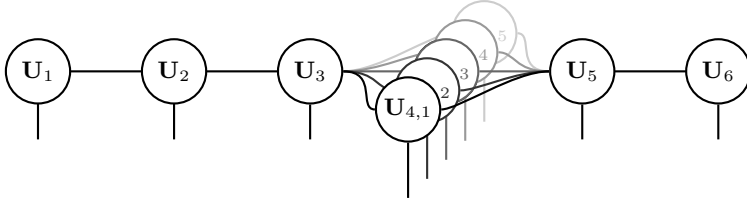


FIG. 2. Graphical representation of a block-4 TT tensor of order 6.

To highlight the position of the index α , we call this format the *block- μ TT format*; see Figure 2 for an illustration. It coincides with the usual TT format (2.1) except for the core $\mathbf{U}_{\mu,\alpha}$, which is different for every α . None of the other cores depend on α , which is the crucial difference between the block- μ TT format and a collection of p TT formats.

Writing $U_\mu(\alpha, i_\mu)$ instead of $U_{\mu,\alpha}(i_\mu)$ in (2.13), it becomes clear that the block- μ TT format can be obtained by applying the TT format to the vectorization $\text{vec}(\mathbf{U})$ of the matrix

$$\mathbf{U} = [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \dots \quad \mathbf{u}_p] \in \mathbb{R}^{n_1 n_2 \dots n_d \times p}$$

into a vector of length $n_1 \dots n_{\mu-1} (pn_\mu) n_{\mu+1} \dots n_d$, where we treat the double index (α, i_μ) at position μ as a single index j_μ running from 1 to $n_\mu p$ in appropriate order. In terms of tensor spaces, (2.13) states that (see (2.5))

$$(2.14) \quad \begin{aligned} \text{ran}(\mathbf{U}) &= \text{span}\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p\} \\ &\subseteq \text{ran}(\mathcal{U}_{\neq \mu}) = \text{ran}(\mathbf{U}_{\leq \mu-1}) \otimes \mathbb{R}^{n_\mu} \otimes \text{ran}(\mathbf{U}_{\geq \mu+1}). \end{aligned}$$

The position of the index α can be moved by separating it from the core \mathbf{U}_μ and attaching it to the $(\mu+1)$ th core, similarly to (2.10); see Algorithm 1 for details. The minimal-rank decomposition (2.15) needed in Algorithm 1 can be performed by, e.g., a QR decomposition with column pivoting or an SVD. In either case, the updated left unfolding $\mathbf{U}_\mu^L \leftarrow Q$ has orthonormal columns, i.e., \mathbf{U}_μ is left orthonormal.

ALGORITHM 1. CONVERSION FROM BLOCK- μ TT FORMAT TO BLOCK- $(\mu+1)$ TT FORMAT.

Input: \mathbf{U} in block- μ TT format, $\mu < d$.

Output: New cores \mathbf{U}_μ and $\mathbf{U}_{\mu+1,\alpha}$ representing \mathbf{U} in block- $(\mu+1)$ TT format.

1: Perform a minimal-rank decomposition

$$(2.15) \quad \begin{aligned} &[\mathbf{U}_{\mu,1}^L \quad \mathbf{U}_{\mu,2}^L \quad \dots \quad \mathbf{U}_{\mu,p}^L] \\ &= Q [P_1 \quad P_2 \quad \dots \quad P_p], \quad Q \in \mathbb{R}^{r_{\mu-1} n_\mu \times s}, P_\alpha \in \mathbb{R}^{s \times r_\mu}. \end{aligned}$$

2: Update cores:

$$\mathbf{U}_\mu^L \leftarrow Q, \quad \mathbf{U}_{\mu+1,\alpha}^R \leftarrow P_\alpha \mathbf{U}_{\mu+1}^R, \quad \alpha = 1, 2, \dots, p.$$

3: Update rank: $r_\mu \leftarrow s$.

ALGORITHM 2. CONVERSION FROM BLOCK- μ FORMAT TO BLOCK- $(\mu - 1)$ FORMAT.

Input: \mathbf{U} in block- μ TT format, $\mu > 1$.**Output:** New cores \mathbf{U}_μ and $\mathbf{U}_{\mu-1,\alpha}$ representing \mathbf{U} in block- $(\mu - 1)$ TT format.

1: Perform a minimal-rank decomposition

$$\begin{bmatrix} \mathbf{U}_{\mu,1}^R \\ \mathbf{U}_{\mu,2}^R \\ \vdots \\ \mathbf{U}_{\mu,p}^R \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_p \end{bmatrix} P, \quad Q_\alpha \in \mathbb{R}^{r_{\mu-1} \times s}, \quad P \in \mathbb{R}^{s \times n_\mu r_\mu}.$$

2: Update cores:

$$\mathbf{U}_\mu^R = P, \quad \mathbf{U}_{\mu-1,\alpha}^L = \mathbf{U}_{\mu-1}^L Q_\alpha, \quad \alpha = 1, 2, \dots, p.$$

3: Update rank: $r_{\mu-1} \leftarrow s$.

2.3. The TT format for linear operators. To efficiently apply a linear operator \mathbf{A} to a vector in TT format, the operator \mathbf{A} itself needs to be represented in a suitable format. More specifically, we represent a matrix $\mathbf{A} \in \mathbb{R}^{m_1 m_2 \cdots m_d \times n_1 n_2 \cdots n_d}$ in the *operator TT format*

$$\mathbf{A}_{(i_1, \dots, i_d), (j_1, \dots, j_d)} = A_1(i_1, j_1) A_2(i_2, j_2) \cdots A_d(i_d, j_d),$$

which is equivalent to applying the vector TT format (2.1) to $\text{vec}(\mathbf{A})$ but treating each pair (i_μ, j_μ) as a single index running from 1 to $m_\mu n_\mu$ in appropriate order.

In principle, any sum of Kronecker products

$$\mathbf{A} = \sum_{k=1}^R L_d^{(k)} \otimes \cdots \otimes L_2^{(k)} \otimes L_1^{(k)}$$

can be represented in the operator TT format with ranks at most $r_1 = \cdots = r_{d-1} = R$ by setting

$$A_1(i_1, j_1) = \begin{bmatrix} L_1^{(1)}(i_1, j_1) & L_1^{(2)}(i_1, j_1) & \cdots & L_1^{(R)}(i_1, j_1) \end{bmatrix},$$

$$A_d(i_d, j_d) = \begin{bmatrix} L_d^{(1)}(i_d, j_d) \\ L_d^{(2)}(i_d, j_d) \\ \vdots \\ L_d^{(R)}(i_d, j_d) \end{bmatrix},$$

and

$$A_\mu(i_\mu, j_\mu) = \begin{bmatrix} L_\mu^{(1)}(i_\mu, j_\mu) & & & \\ & L_\mu^{(2)}(i_\mu, j_\mu) & & \\ & & \ddots & \\ & & & L_\mu^{(R)}(i_\mu, j_\mu) \end{bmatrix}, \quad \mu = 2, \dots, d-1.$$

In particular, if $\mathbf{A} = L_d \otimes \cdots \otimes L_2 \otimes L_1$, the cores are given by the coefficients L_μ themselves and the operator TT ranks are all equal to one.

The vector $\tilde{\mathbf{u}}$ resulting from a matrix-vector product $\tilde{\mathbf{u}} = \mathbf{A}\mathbf{u}$, with \mathbf{u} in TT format and \mathbf{A} in operator TT format, is again in TT format. Its cores $\tilde{\mathbf{U}}_\mu$ are given by the formula [33]

$$\tilde{U}_\mu(i_\mu) = \sum_{j_\mu=1}^{n_\mu} A_\mu(i_\mu, j_\mu) \otimes U_\mu(j_\mu),$$

which can be implemented as the matrix-matrix multiplication

$$(2.16) \quad \tilde{\mathbf{U}}'_\mu = \mathbf{A}'_\mu \mathbf{U}'_\mu,$$

where the prime symbol denotes suitable reshapings. The TT ranks of $\tilde{\mathbf{u}}$ are thus bounded by the products of the TT ranks of \mathbf{A} and \mathbf{u} .

2.3.1. Laplace-like operators. Matrices of the form

$$(2.17) \quad \mathbf{A} = \sum_{\mu=1}^d M_d \otimes \cdots \otimes M_{\mu+1} \otimes L_\mu \otimes M_{\mu-1} \otimes \cdots \otimes M_1$$

arise when discretizing the d -dimensional Laplace operator. It turns out that this particular class admits a rank-two operator TT representation [23] with cores

$$A_1(i_1, j_1) = \begin{bmatrix} L_1(i_1, j_1) & M_1(i_1, j_1) \end{bmatrix}, \quad A_d(i_d, j_d) = \begin{bmatrix} M_d(i_d, j_d) \\ L_d(i_d, j_d) \end{bmatrix},$$

and

$$\begin{bmatrix} M_\mu(i_\mu, j_\mu) & 0 \\ L_\mu(i_\mu, j_\mu) & M_\mu(i_\mu, j_\mu) \end{bmatrix}, \quad \mu = 2, \dots, d-1.$$

For symmetric positive definite coefficient matrices L_μ, M_μ , the inverse of (2.17) can be well approximated by a sum of Kronecker products of exponentials [13]:

$$\mathbf{A}^{-1} \approx \sum_{k=1}^R \frac{\omega_k}{\lambda_{\min}} E_{d,k} \otimes \cdots \otimes E_{2,k} \otimes E_{1,k} \quad \text{with} \quad E_{\mu,k} = \exp\left(-\frac{\alpha_k}{\lambda_{\min}} M_\mu^{-1} L_\mu\right) M_\mu^{-1},$$

that is, in operator TT format with all ranks bounded by R . The approximation error in the spectral norm is governed by the maximal error of the approximation

$$\frac{1}{x} \approx \sum_{k=1}^R \omega_k e^{-\alpha_k x}$$

on the interval $[1, \lambda_{\max}/\lambda_{\min}]$, where λ_{\min} and λ_{\max} denote the smallest and largest eigenvalue of \mathbf{A} ; see [17, section 9.7.2]. Usually, rather small values for R are sufficient to attain an acceptable error. Almost optimal values for R , ω_k , and α_k to achieve a specific accuracy have been reported in [16].

2.3.2. Potentials with nearest neighbor interaction. In the numerical experiments, we will consider operators that arise from the discretization of the Laplace operator plus a potential $V(\mathbf{x})$. In general, when the potential is discretized to a grid function \mathbf{v} , its application to another grid function \mathbf{u} takes the form of a Hadamard product

$$u_{i_1, i_2, \dots, i_d} \mapsto v_{i_1, i_2, \dots, i_d} \cdot u_{i_1, i_2, \dots, i_d}.$$

When both \mathbf{u} and \mathbf{v} are in the TT format with small ranks, the Hadamard product can be efficiently implemented [33] using only core-by-core operations. The Hadamard product will typically multiply the TT ranks, so that rank truncation might be necessary afterwards.

In some cases, for instance in the case of the Henon–Heiles potential (see section 4), the potential describes nearest neighbor interactions, leading to operators of the form

$$(2.18) \quad \mathbf{A} = \sum_{\mu=1}^d M_d \otimes \cdots \otimes M_{\mu+1} \otimes L_\mu \otimes M_{\mu-1} \otimes \cdots \otimes M_1 \\ + \sum_{\mu=1}^{d-1} M_d \otimes \cdots \otimes C_{\mu+1} \otimes B_\mu \otimes \cdots \otimes M_1.$$

Remarkably, operators of this form allow for a rank-three operator TT representation [26] using the cores

$$A_1(i_1, j_1) = [L_1(i_1, j_1) \quad B_1(i_1, j_1) \quad M_1(i_1, j_1)], \quad A_d(i_d, j_d) = \begin{bmatrix} M_d(i_d, j_d) \\ C_d(i_d, j_d) \\ L_d(i_d, j_d) \end{bmatrix},$$

and

$$A_\mu(i_\mu, j_\mu) = \begin{bmatrix} M_\mu(i_\mu, j_\mu) & 0 & 0 \\ C_\mu(i_\mu, j_\mu) & 0 & 0 \\ L_\mu(i_\mu, j_\mu) & B_\mu(i_\mu, j_\mu) & M_\mu(i_\mu, j_\mu) \end{bmatrix}, \quad \mu = 2, \dots, d-1.$$

The suitably reshaped cores \mathbf{A}'_μ needed in (2.16) when performing a matrix-vector multiplication are given by

$$\mathbf{A}'_1 = L_1 \otimes \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + B_1 \otimes \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + M_1 \otimes \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \\ \mathbf{A}'_d = M_d \otimes \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + C_d \otimes \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + L_d \otimes \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

and

$$\mathbf{A}'_\mu = M_\mu \otimes (\mathbf{e}_1 + \mathbf{e}_9) + L_\mu \otimes \mathbf{e}_3 + B_\mu \otimes \mathbf{e}_6 + C_\mu \otimes \mathbf{e}_2, \quad \mu = 2, \dots, d-1,$$

where \mathbf{e}_i denotes the i th unit vector of length 9.

3. Trace minimization in TT format. We return to the trace minimization problem (1.1), with the additional constraint that $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p]$ is represented in the block-1 TT format.

3.1. Alternating optimization. Our starting point is the recently proposed alternating algorithm of Pizorn and Verstraete [35], called the *variational numerical renormalization group*, for excited states calculations in quantum many-body systems. In the numerical analysis community, a variant of this algorithm has been proposed by Dolgov et al. [6].

The algorithm proceeds by alternatingly optimizing each core of the block TT format, similar to block coordinate descent methods in optimization. Given a matrix \mathbf{U} in the block- μ TT format, let us consider the optimization of its μ th core under the assumption that all other cores are fixed and orthonormalized, such that (2.12) holds. Using (2.3), we have

$$\mathbf{U} = \mathcal{U}_{\neq\mu} \mathbf{V}$$

with the matrix

$$(3.1) \quad \mathbf{V} = [\text{vec}(\mathbf{U}_{\mu,1}^R) \quad \text{vec}(\mathbf{U}_{\mu,2}^R) \quad \dots \quad \text{vec}(\mathbf{U}_{\mu,p}^R)] \in \mathbb{R}^{r_{\mu-1}n_{\mu}r_{\mu} \times p}.$$

Since $\mathcal{U}_{\neq\mu}$ has orthonormal columns, the matrix \mathbf{U} has orthonormal columns if and only if $\mathbf{V}^T \mathbf{V} = I_p$. Therefore, optimizing (1.1) with respect to the μ th core becomes the trace minimization problem

$$(3.2) \quad \min\{\text{trace}(\mathbf{V}^T \mathcal{A}_{\mu} \mathbf{V}) : \mathbf{V} \in \mathbb{R}^{r_{\mu-1}n_{\mu}r_{\mu} \times p}, \mathbf{V}^T \mathbf{V} = I_p\},$$

with the *reduced matrix*

$$(3.3) \quad \mathcal{A}_{\mu} = \mathcal{U}_{\neq\mu}^T \mathbf{A} \mathcal{U}_{\neq\mu}.$$

In terms of tensor spaces, it follows from (2.5) and (2.14) that this problem is equivalent to

$$(3.4) \quad \min\{\text{trace}(\mathbf{U}^T \mathbf{A} \mathbf{U}) : \text{ran}(\mathbf{U}) \subseteq \text{ran}(\mathbf{U}_{\geq\mu+1}) \otimes \mathbb{R}^{n_{\mu}} \otimes \text{ran}(\mathbf{U}_{\leq\mu-1}) \\ = \text{ran}(\mathcal{U}_{\neq\mu}), \mathbf{U}^T \mathbf{U} = I_p\}.$$

After the core $\mathbf{U}_{\mu,\alpha}$ has been replaced by the solution \mathbf{V} of (3.2), the index α is moved to the right using Algorithm 1. In this manner one proceeds from left to right until $\mu = d$. The described procedure constitutes one *half sweep* of ALS [21]. When the d th core is reached, one continues by a half sweep from right to left in an analogous manner. Two subsequent half sweeps are called a *full sweep*.

Algorithm 3 describes one sweep of ALS. The solution of the reduced problem (3.2) can be done using LOBPCG [28], which benefits from the use of a preconditioner for the reduced matrix \mathcal{A}_{μ} . This will be discussed in section 3.3.

3.2. Core enrichment based on gradient information. The convergence of ALS can be improved by enriching the cores with information that aims to improve the quality of the current approximation. Such a core enrichment has been proposed by White [43] in the context of one-site DMRG algorithms, to overcome the unfavorable scaling of *exact* two-site DMRG algorithms. Dolgov and Savostyanov [8, 9] have significantly extended this idea to ALS-type algorithms for solving symmetric and

ALGORITHM 3. ONE SWEEP OF ALS FOR SOLVING THE TRACE MINIMIZATION PROBLEM (1.1).

Input: Starting guess \mathbf{U} in block-1 TT format with right-orthonormal cores $\mathbf{U}_2, \dots, \mathbf{U}_d$.

- 1: **for** $\mu = 1, 2, \dots, d-1$ **do**
 - 2: Replace core $\mathbf{U}_{\mu,\alpha}$ by the solution \mathbf{V} of (3.2).
 - 3: Apply Algorithm 1 to move position of index α to $\mu+1$, such that the updated core \mathbf{U}_μ^L is left orthonormal.
 - 4: **end for**
 - 5: **for** $\mu = d, d-1, \dots, 2$ **do**
 - 6: Replace core $\mathbf{U}_{\mu,\alpha}$ by the solution \mathbf{V} of (3.2).
 - 7: Apply Algorithm 2 to move position of index α to $\mu-1$, such that the updated core \mathbf{U}_μ^R is right orthonormal.
 - 8: **end for**
-

nonsymmetric linear systems (see also [10] for a discussion of the similarities and differences of both approaches). In this section, we show how core enrichment can be combined with Algorithm 3. For this purpose, it is convenient to first discuss a variant that enriches all cores simultaneously. Although such a *global correction* is conceptually simple, it bears the disadvantage of enlarging all TT ranks at the same time, which makes all subsequent operations of ALS significantly more expensive. We therefore develop a second variant, which only enriches the cores that are next to the core currently optimized in a step of ALS. Such a *local correction* not only decreases the computational effort, but it also allows us to adjust the correction during an ALS sweep.

3.2.1. Global correction. This approach was termed $ALS(t+z)$ in [8]. Assume that \mathbf{U} is our current approximation in block-1 TT format and suppose we add an error correction \mathbf{R} , also in block-1 TT format. Then $\tilde{\mathbf{U}} = \mathbf{U} + \mathbf{R}$ can again be represented in block-1 TT format using the cores

$$(3.5) \quad \tilde{\mathbf{U}}_{\alpha,1}(i_1) = [U_{1,\alpha}(i_1) \quad R_{1,\alpha}(i_1)], \quad \tilde{\mathbf{U}}_d(d_1) = \begin{bmatrix} U_d(i_d) \\ R_d(i_d) \end{bmatrix},$$

and

$$(3.6) \quad \tilde{\mathbf{U}}_\mu(i_\mu) = \begin{bmatrix} U_\mu(i_\mu) & \mathbf{0} \\ \mathbf{0} & R_\mu(i_\mu) \end{bmatrix}, \quad \mu = 2, \dots, d-1.$$

In particular, addition sums up the TT ranks of \mathbf{U} and \mathbf{R} .

The $ALS(t+z)$ algorithm now proceeds by applying a sweep of ALS (Algorithm 3) to (3.5)–(3.6). According to (3.4), the first step of ALS determines the best p -dimensional subspace for the trace minimization problem constrained to the range of $\tilde{\mathcal{U}}_{\neq 1}$. The following proposition implies that this constraint includes all linear combinations from the ranges of \mathbf{U} and \mathbf{R} .

PROPOSITION 3.1. *Consider a matrix $\tilde{\mathbf{U}} = \mathbf{U} + \mathbf{R}$ in block-1 TT format with the cores given by (3.5)–(3.6). Then the matrix $\tilde{\mathcal{U}}_{\neq 1} = \tilde{\mathbf{U}}_{\geq 2} \otimes I_{n_1}$ satisfies $\text{ran}(\tilde{\mathcal{U}}_{\neq 1}) \supseteq \text{ran}(\mathbf{U}) + \text{ran}(\mathbf{R})$, i.e.,*

$$\text{ran}(\tilde{\mathcal{U}}_{\neq 1}) \supseteq \{\mathbf{U}\mathbf{S} + \mathbf{R}\mathbf{T} : \mathbf{S}, \mathbf{T} \in \mathbb{R}^{p \times p}\}.$$

Proof. Because of the structure of the TT cores of $\tilde{\mathbf{U}}$, we have $\tilde{\mathbf{U}}_{\geq 2} = [\mathbf{U}_{\geq 2} \quad \mathbf{R}_{\geq 2}]$. Hence,

$$\begin{aligned} \text{ran}(\mathbf{U}) + \text{ran}(\mathbf{R}) &\subseteq \text{ran}(\mathcal{U}_{\neq 1}) + \text{ran}(\mathbf{R}_{\neq 1}) \\ &= (\text{ran}(\mathbf{U}_{\geq 2}) \otimes \mathbb{R}^{n_1}) + (\text{ran}(\mathbf{R}_{\geq 2}) \otimes \mathbb{R}^{n_1}) \\ &= (\text{ran}(\mathbf{U}_{\geq 2}) + \text{ran}(\mathbf{R}_{\geq 2})) \otimes \mathbb{R}^{n_1} \\ &= \text{ran}(\tilde{\mathbf{U}}_{\geq 2}) \otimes \mathbb{R}^{n_1} = \text{ran}(\tilde{\mathcal{U}}_{\neq 1}), \end{aligned}$$

where the first inclusion is due to (2.14). \square

Proposition 3.1 implies that the result from the first step of ALS applied to $\tilde{\mathcal{U}}$ is at least as good as selecting the best p linear combinations from $\text{ran}(\mathbf{U})$ and $\text{ran}(\mathbf{R})$. This point of view allows us to interpret \mathbf{R} as a *subspace correction*. In particular, it is important to emphasize that the particular linear combination $\mathbf{U} + \mathbf{R}$ is only used to setup the cores; it does *not* correspond to the correction of \mathbf{U} that is actually used.

Similar statements can be made when \mathbf{U} and \mathbf{R} are represented in the block- μ TT format for $\mu \neq 1$. In particular, this gives the possibility of injecting an additional subspace acceleration before every step of Algorithm 3. However, the incurred growth of the TT ranks makes such an approach little attractive. In the ALS(t+z) algorithm proposed in [7] a subspace correction is therefore only added before every left-to-right and before every right-to-left half-sweep.

It remains to discuss the choice of \mathbf{R} . A natural candidate is the negative block residual $-(\mathbf{A}\mathbf{U} - \mathbf{U}\Lambda)$ with $\Lambda = \mathbf{U}^\top \mathbf{A}\mathbf{U}$. This choice can be motivated as follows. As \mathbf{A} is symmetric, the gradient of $\frac{1}{2} \text{trace}(\mathbf{U}^\top \mathbf{A}\mathbf{U})$ as a function of \mathbf{U} is given by $\mathbf{A}\mathbf{U}$. The optimization task (1.1) is posed on the Stiefel manifold $\{\mathbf{U} \in \mathbb{R}^{N \times p} : \mathbf{U}^\top \mathbf{U} = \mathbf{I}_p\}$. Therefore, the direction of steepest descent is given by the orthogonal projection of the negative gradient onto the tangent space, which—according to [1, Example 3.6.2]—is calculated as

$$(\mathbf{I} - \mathbf{U}\mathbf{U}^\top)(-\mathbf{A}\mathbf{U}) + \mathbf{U} \text{skew}(\mathbf{U}^\top(-\mathbf{A}\mathbf{U})) = -(\mathbf{A}\mathbf{U} - \mathbf{U}\Lambda),$$

where we have used that $\text{skew}(-\Lambda)$, the skew-symmetric part of $-\Lambda$, is zero.

The convergence of gradient methods for solving eigenvalue problems critically depends on certain eigenvalue gaps *relative* to the width of the spectrum of \mathbf{A} and it deteriorates when these gaps approach zero. The method of steepest descent applied to (1.1) is no exception [32]. For discretized PDE eigenvalue problems, the unboundedness of the underlying operator implies that the width of the spectrum becomes wider (therefore, the relative eigenvalue gaps become smaller) as the discretization gets refined. This effect hampers convergence, but it can be avoided by using preconditioned gradients instead. In the case of (1.1), the negative preconditioned gradient is given by the negative¹ *preconditioned block residual*

$$(3.7) \quad \mathbf{R} = -\mathbf{B}^{-1}(\mathbf{A}\mathbf{U} - \mathbf{U}\Lambda)$$

for a preconditioner \mathbf{B} of \mathbf{A} ; see also [4]. If \mathbf{B} is spectrally equivalent to \mathbf{A} then the convergence rate of the resulting preconditioned gradient method does not deteriorate as the discretization gets refined [32]. To make use of (3.7) in the context of low-rank tensor methods, \mathbf{B}^{-1} needs to be represented in the operator TT format; see

¹The sign and norm of the \mathbf{R} are irrelevant for the subsequent optimization steps. We use negative residuals in the presentation, as it allows for the interpretation of the enrichment as a correction step in the direction of steepest descent.

section 2.3.1 for an example. As usual, both the computation of \mathbf{R} and its addition to \mathbf{U} will only be performed approximately, that is, low-rank truncation is applied to limit the rank growth.

3.2.2. Local corrections. As discussed above, the simultaneous increase of all TT ranks renders a global correction too expensive to be applied before *every* step of ALS. To address this issue, an approach based on local corrections has been proposed for linear systems in [9]. In the following, we will extend this approach to eigenvalue problems.

Let us consider the μ th step of a left-to-right ALS sweep and suppose that the μ th core has been optimized, but the position of the index α has not yet been moved to $\mu + 1$. We then augment the neighboring cores with the corresponding cores of a correction \mathbf{R} :

$$(3.8) \quad \tilde{\mathbf{U}}_{\mu,\alpha}^{\mathbf{L}} = [\mathbf{U}_{\mu,\alpha}^{\mathbf{L}} \quad \mathbf{R}_{\mu,\alpha}^{\mathbf{L}}], \quad \tilde{\mathbf{U}}_{\mu+1}^{\mathbf{R}} = \begin{bmatrix} \mathbf{U}_{\mu+1}^{\mathbf{R}} \\ \mathbf{R}_{\mu+1}^{\mathbf{R}} \end{bmatrix}.$$

In particular, only the rank r_μ is changed. Similarly we proceed in a right-to-left sweep; see Algorithm 4 for a complete description and Figure 3 for an illustration. Since the analogous algorithm for linear systems was coined AMEn (Alternating Minimal Energy method) [9], we will call our algorithm *EVAMEn* (EigenValue AMEn).

ALGORITHM 4. ONE FULL SWEEP OF EVAMEn FOR SOLVING THE TRACE MINIMIZATION PROBLEM (1.1).

Input: Starting guess \mathbf{U} in block-1 TT format and right-orthonormal cores $\mathbf{U}_2, \dots, \mathbf{U}_d$.

- 1: **for** $\mu = 1, 2, \dots, d-1$ **do**
 - 2: Replace core $\mathbf{U}_{\mu,\alpha}$ by the solution \mathbf{V} of (3.2).
 - 3: Augment cores $\mathbf{U}_{\mu,\alpha}$ and $\mathbf{U}_{\mu+1}$ with cores $\mathbf{R}_{\mu,\alpha}$ and $\mathbf{R}_{\mu+1}$ of compatible size, according to (3.8).
 - 4: Apply Algorithm 1 to move position of index α to $\mu + 1$, such that the updated core $\mathbf{U}_\mu^{\mathbf{L}}$ is left orthonormal.
 - 5: **end for**
 - 6: **for** $\mu = d, d-1, \dots, 2$ **do**
 - 7: Replace core $\mathbf{U}_{\mu,\alpha}$ by the solution \mathbf{V} of (3.2).
 - 8: Augment cores $\mathbf{U}_{\mu-1}$ and $\mathbf{U}_{\mu,\alpha}$ with cores $\mathbf{R}_{\mu-1}$ and $\mathbf{R}_{\mu,\alpha}$ of compatible size, analogously to (3.8).
 - 9: Apply Algorithm 1 to move position of index α to $\mu - 1$, such that the updated core $\mathbf{U}_\mu^{\mathbf{R}}$ is right orthonormal.
 - 10: **end for**
-

We now turn to an important interpretation of Algorithm 4 as an alternating local subspace correction method for the famous *two-site DMRG* algorithm [42], which we therefore briefly recall.

DMRG for trace minimization. Assume that \mathbf{U} is either in block- μ or in block- $(\mu + 1)$ TT format. One step of the DMRG algorithm applied to (1.1) begins with merging two neighboring cores, at positions μ and $\mu + 1$, and then solves the resulting optimization problem for the merged *supercore* of size $r_{\mu-1} \times n_\mu \times n_{\mu+1} \times r_{\mu+1}$, denoted by \mathbf{W} in the following:

$$(3.9) \quad \min\{\text{trace}(\mathbf{W}^{\mathbf{T}} \mathcal{A}_{\mu,\mu+1} \mathbf{W}) : \mathbf{W} \in \mathbb{R}^{r_{\mu-1} n_\mu n_{\mu+1} r_{\mu+1} \times p}, \mathbf{W}^{\mathbf{T}} \mathbf{W} = I_p\},$$

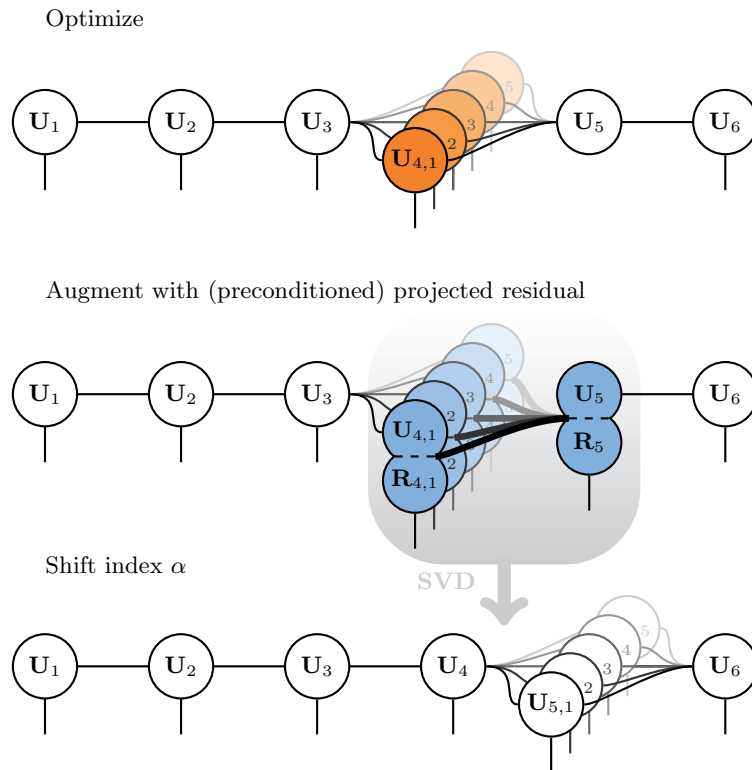


FIG. 3. One microiteration in a left-to-right sweep of the EVAMEn algorithm (lines 2–4 of Algorithm 4).

where

$$\mathcal{A}_{\mu,\mu+1} = \mathcal{U}_{\neq\mu,\mu+1}^T \mathbf{A} \mathcal{U}_{\neq\mu,\mu+1},$$

and the matrix $\mathcal{U}_{\neq\mu,\mu+1}$ defined in (2.7) is assumed to have orthonormal columns. By (2.8), the DMRG subproblem (3.9) is equivalent to minimizing the trace of $\mathbf{U}^T \mathbf{A} \mathbf{U}$ under the constraints $\mathbf{U}^T \mathbf{U} = I_p$ and

$$\text{ran}(\mathbf{U}) \subseteq \text{ran}(\mathbf{U}_{\geq\mu+2}) \otimes \mathbb{R}^{n_{\mu+1}} \otimes \mathbb{R}^{n_{\mu}} \otimes \text{ran}(\mathbf{U}_{\leq\mu-1}) = \text{ran}(\mathcal{U}_{\neq\mu,\mu+1}).$$

Once (3.9) is solved, a *minimal* subspace $\text{ran}(\mathbf{U}_{\leq\mu}) \subseteq \mathbb{R}^{n_{\mu}} \otimes \text{ran}(\mathbf{U}_{\leq\mu-1})$ is determined such that $\text{ran}(\mathbf{U}) \subseteq \text{ran}(\mathbf{U}_{\geq\mu+2}) \otimes \mathbb{R}^{n_{\mu+1}} \otimes \text{ran}(\mathbf{U}_{\leq\mu})$ holds. Computationally, this means that the solution \mathbf{W} of (3.9) is reshaped into an $(r_{\mu-1}n_{\mu}) \times (n_{\mu+1}r_{\mu+1}p)$ matrix and a minimal-rank decomposition is applied to yield new cores \mathbf{U}_{μ} , $\mathbf{U}_{\mu+1,\alpha}$ such that

$$\mathbf{W} = [\text{vec}(\mathbf{U}_{\mu}^L \mathbf{U}_{\mu+1,1}^R) \quad \text{vec}(\mathbf{U}_{\mu}^L \mathbf{U}_{\mu+1,2}^R) \quad \dots \quad \text{vec}(\mathbf{U}_{\mu}^L \mathbf{U}_{\mu+1,p}^R)].$$

Since this decomposition automatically chooses a new, currently optimal rank r_{μ} , the DMRG is rank adaptive, even for $p = 1$. Note that the above decomposition of \mathbf{W} also moves the index α to the next position. In practice, the decomposition is only performed approximately, using a truncated SVD, to avoid that the updated rank r_{μ} becomes too large.

One full sweep of DMRG consists of processing all possible supercores in the manner described above, first from left to right and second from right to left. The global and local convergence properties of the DMRG algorithm are not well understood. However, there are cases for which DMRG needs very few sweeps to converge to high accuracy; see, e.g., [29]. It is, however, quite costly, since the local two-core problem matrix $\mathcal{A}_{\mu,\mu+1}$ is of size $r_{\mu-1}n_{\mu}n_{\mu+1}r_{\mu} \times r_{\mu-1}n_{\mu}n_{\mu+1}r_{\mu}$.

Relation of Algorithm 4 to DMRG. Following [8], we now explain how Algorithm 4, which only operates on individual cores, can be viewed as producing approximate solutions to the DMRG subproblem (3.9). For this purpose, we consider the μ th step of Algorithm 4 and interpret the supercore

$$(3.10) \quad \mathbf{W}^0 = [\text{vec}(\mathbf{U}_{\mu,1}^{\text{L}} \mathbf{U}_{\mu+1}^{\text{R}}) \quad \text{vec}(\mathbf{U}_{\mu,2}^{\text{L}} \mathbf{U}_{\mu+1}^{\text{R}}) \quad \dots \quad \text{vec}(\mathbf{U}_{\mu,p}^{\text{L}} \mathbf{U}_{\mu+1}^{\text{R}})],$$

where $\mathbf{U}_{\mu,\alpha}$ and $\mathbf{U}_{\mu+1}$ are the cores just before the augmentation in line 3, as an initial guess for (3.9). Given the cores $\mathbf{R}_{\mu,\alpha}$ and $\mathbf{R}_{\mu+1}$ of the correction, we define²

$$(3.11) \quad \mathbf{R}_{\mu,\mu+1} = [\text{vec}(\mathbf{R}_{\mu,1}^{\text{L}} \mathbf{R}_{\mu+1}^{\text{R}}) \quad \text{vec}(\mathbf{R}_{\mu,2}^{\text{L}} \mathbf{R}_{\mu+1}^{\text{R}}) \quad \dots \quad \text{vec}(\mathbf{R}_{\mu,p}^{\text{L}} \mathbf{R}_{\mu+1}^{\text{R}})].$$

Then the augmented cores $\tilde{\mathbf{U}}_{\alpha,\mu}$ and $\tilde{\mathbf{U}}_{\mu+1}$, which have been formed according to (3.8), satisfy

$$\tilde{\mathbf{W}}^1 := [\text{vec}(\tilde{\mathbf{U}}_{\mu,1}^{\text{L}} \tilde{\mathbf{U}}_{\mu+1}^{\text{R}}) \quad \text{vec}(\tilde{\mathbf{U}}_{\mu,2}^{\text{L}} \tilde{\mathbf{U}}_{\mu+1}^{\text{R}}) \quad \dots \quad \text{vec}(\tilde{\mathbf{U}}_{\mu,p}^{\text{L}} \tilde{\mathbf{U}}_{\mu+1}^{\text{R}})] = \mathbf{W}^0 + \mathbf{R}_{\mu,\mu+1}.$$

Moving the index α in line 4 of Algorithm 4 does not affect this equality, so that

$$(3.12) \quad \tilde{\mathbf{W}}^1 = [\text{vec}(\mathbf{U}_{\mu}^{\text{L}} \tilde{\mathbf{U}}_{\mu+1,1}^{\text{R}}) \quad \text{vec}(\mathbf{U}_{\mu}^{\text{L}} \tilde{\mathbf{U}}_{\mu+1,2}^{\text{R}}) \quad \dots \quad \text{vec}(\mathbf{U}_{\mu}^{\text{L}} \tilde{\mathbf{U}}_{\mu+1,p}^{\text{R}})] = \mathbf{W}^0 + \mathbf{R}_{\mu,\mu+1}$$

also holds for the updated cores \mathbf{U}_{μ} and $\tilde{\mathbf{U}}_{\mu+1,\alpha}$. The next step of Algorithm 4 begins by replacing $\tilde{\mathbf{U}}_{\mu+1,\alpha}$ with the optimized cores $\mathbf{U}_{\mu+1,\alpha}$. After this optimization, we can regard

$$(3.13) \quad \mathbf{W}^1 = [\text{vec}(\mathbf{U}_{\mu}^{\text{L}} \mathbf{U}_{\mu+1,1}^{\text{R}}) \quad \text{vec}(\mathbf{U}_{\mu}^{\text{L}} \mathbf{U}_{\mu+1,2}^{\text{R}}) \quad \dots \quad \text{vec}(\mathbf{U}_{\mu}^{\text{L}} \mathbf{U}_{\mu+1,p}^{\text{R}})]$$

as an improved approximation to the true DMRG solution.

To get a better understanding of the improvement attained by \mathbf{W}^1 , we now relate this matrix to the trace minimization (3.9) solved in DMRG. By (2.9), the problem

$$\min\{\text{trace}(\mathbf{V}^{\text{T}} \mathcal{A}_{\neq\mu+1} \mathbf{V}) : \mathbf{V}^{\text{T}} \mathbf{V} = I_p\},$$

which is solved for determining \mathbf{W}^1 , is equivalent to the problem

$$(3.14) \quad \min\{\text{trace}(\mathbf{W}^{\text{T}} \mathcal{A}_{\neq\mu,\mu+1} \mathbf{W}) : \mathbf{W} \in \text{ran}(I_{r_{\mu+1}n_{\mu+1}} \otimes \mathbf{U}_{\mu}^{\text{L}}), \mathbf{W}^{\text{T}} \mathbf{W} = I_p\}.$$

The following proposition shows that the constraint includes all linear combinations from $\text{ran}(\mathbf{W}^0)$ and $\text{ran}(\mathbf{R}_{\mu,\mu+1})$.

PROPOSITION 3.2. *Consider the matrices \mathbf{W}^0 and $\mathbf{R}_{\mu,\mu+1}$ defined above. Then the core tensor \mathbf{U}_{μ} , obtained after moving the index α , satisfies*

$$(3.15) \quad \text{ran}(I_{r_{\mu+1}n_{\mu+1}} \otimes \mathbf{U}_{\mu}^{\text{L}}) \supseteq \text{ran}(\mathbf{W}^0) + \text{ran}(\mathbf{R}_{\mu,\mu+1}) = \{\mathbf{W}^0 S + \mathbf{R}_{\mu,\mu+1} T : S, T \in \mathbb{R}^{p \times p}\}.$$

²In practice, we first choose $\mathbf{R}_{\mu,\mu+1} \in \mathbb{R}^{r_{\mu-1}n_{\mu}n_{\mu+1}r_{\mu+1} \times p}$ and then decompose it into the form (3.11) by an (approximate) minimal-rank decomposition.

Proof. By (2.15), moving the index α leads to

$$[\mathbf{U}_{\mu,\alpha}^L \quad \mathbf{R}_{\mu,\alpha}^L] = \tilde{\mathbf{U}}_{\mu,\alpha}^L = \mathbf{U}_\mu^L P_\alpha$$

for some rank matrix P_α , where $\mathbf{U}_{\mu,\alpha}$ and \mathbf{U}_μ refer to the cores before and after moving the index α . This shows that all columns of \mathbf{W}^0 and $\mathbf{R}_{\mu,\mu+1}$ are in $\mathbb{R}^{r_{\mu+1}n_{\mu+1}} \otimes \text{ran}(\mathbf{U}_\mu^L)$, implying the assertion. \square

Combined with (3.14), Proposition 3.2 allows us to conclude that the improved approximation \mathbf{W}^1 is at least as good as the one obtained by $\mathbf{W}^0 S + \mathbf{R}_{\mu,\mu+1} T$ with optimal choices for $S, T \in \mathbb{R}^{p \times p}$. While this implies choosing an optimal p -dimensional subspace in the (generically) $2p$ -dimensional subspace $\text{ran}(\mathbf{W}^0) + \text{ran}(\mathbf{R}_{\mu,\mu+1})$, we should emphasize however, that the actual optimization (3.14) (or, equivalently, the optimization of the $\mathbf{U}_{\mu+1,\alpha}$) is over a much larger space of (generic) dimension $r_\mu n_{\mu+1} r_{\mu+1}$. Similar considerations can be made for the right-to-left sweep. Thus, Algorithm 4 can be viewed as an *enhanced* local subspace correction method for solving the DMRG subproblems (3.9) approximately.

Choice of $\mathbf{R}_{\mu,\mu+1}$. The insights gained from the relation of Algorithm 4 to DMRG can be used to guide the choice of the local corrections $\mathbf{R}_{\mu,\mu+1}$. A natural choice, that can also be efficiently implemented, is given by

$$(3.16) \quad \mathbf{R}_{\mu,\mu+1} = -(\mathcal{A}_{\mu,\mu+1} \mathbf{W}^0 - \mathbf{W}^0 \Lambda)$$

with $\Lambda = (\mathbf{W}^0)^\top \mathcal{A}_{\mu,\mu+1} \mathbf{W}^0$. This is the orthogonal projection of the negative gradient of the local DMRG problem (3.9) on the Stiefel manifold. Thus the addition of $\mathbf{R}_{\mu,\mu+1}$ to \mathbf{W}^0 as implicitly performed in the augmentation step 3 of Algorithm 4 can be seen as a steepest descent step for the DMRG problem.

Furthermore, by (2.12) and (2.6) it holds

$$\mathcal{A}_{\mu,\mu+1} \mathbf{W}^0 - \mathbf{W}^0 \Lambda = \mathcal{U}_{\neq \mu, \mu+1}^\top (\mathbf{A} \mathbf{U} - \mathbf{U} \Lambda),$$

and also $\Lambda = \mathbf{U}^\top \mathbf{A} \mathbf{U}$. Thus, the choice (3.16) can at the same time be interpreted as the projection of the global negative residual onto the local subspace of the local DMRG problem. In this sense, global residual information is injected into the local problem.

Our experiments (see section 4) demonstrate that it is beneficial to use a preconditioned DMRG residual

$$(3.17) \quad \mathbf{R}_{\mu,\mu+1} = -\mathcal{B}_{\mu,\mu+1}^{-1} (\mathcal{A}_{\mu,\mu+1} \mathbf{W}^0 - \mathbf{W}^0 \Lambda)$$

instead of (3.16). The derivation of such a local preconditioner $\mathcal{B}_{\mu,\mu+1}^{-1}$ for $\mathcal{A}_{\mu,\mu+1}$ will be discussed in the next section. An alternative to (3.17) would be to use $\mathbf{R}_{\mu,\mu+1} = -\mathcal{U}_{\mu,\mu+1}^\top \mathbf{B}^{-1} (\mathbf{A} \mathbf{U} - \mathbf{U} \Lambda)$, where \mathbf{B}^{-1} is a preconditioner for \mathbf{A} . This corresponds to injecting information on the preconditioned global residual into the local DMRG problem. Although this choice is better than using no preconditioner information, our numerical experiments indicate that it is inferior to the preconditioned local DMRG residual (3.17).

Incorporating the residual increases the TT rank to $r_\mu \leftarrow r_\mu + s_\mu$, where s_μ is the TT rank of $\mathbf{R}_{\mu,\mu+1}$. The exact computation of $\mathbf{R}_{\mu,\mu+1}$ by (3.17) will typically lead to an unacceptably large value of s_μ . To avoid this, we use repeated low-rank truncation to limit this additional rank to $s_\mu \leq 2$. Such a truncation can be justified by the fact that the addition of $\mathbf{R}_{\mu,\mu+1}$ to \mathbf{W}^0 only serves as a very rough subspace correction, with the fine tuning performed by ALS.

3.3. Construction of local preconditioners. In our algorithms, preconditioning is important at two points:

- The reduced eigenvalue problems (3.2) are solved by LOBPCG [28]. The convergence, and hence the time needed, by LOBPCG crucially depends on the availability of a good preconditioner for the reduced matrix $\mathcal{A}_\mu = \mathcal{U}_{\neq\mu}^\top \mathbf{A} \mathcal{U}_{\neq\mu}$;
- the preconditioned DMRG residual (3.17) for the augmentation steps in EVAMEn requires a preconditioner for $\mathcal{A}_{\mu,\mu+1} = \mathcal{U}_{\neq\mu,\mu+1}^\top \mathbf{A} \mathcal{U}_{\neq\mu,\mu+1}$.

Both problems consist of finding a preconditioner for a reduced matrix $\mathcal{A} = \mathcal{U}^\top \mathbf{A} \mathcal{U}$, where \mathcal{U} has orthonormal columns. As explained in [29], a preconditioner \mathbf{B}^{-1} for \mathbf{A} leads to an at least equally good preconditioner

$$(3.18) \quad \mathcal{B}^{-1} \approx (\mathcal{U}^\top \mathbf{B} \mathcal{U})^{-1}$$

for $\mathcal{U}^\top \mathbf{A} \mathcal{U}$. It is a nontrivial task to efficiently construct and apply $(\mathcal{U}^\top \mathbf{B} \mathcal{U})^{-1}$.

A special case is Laplace-like operators of the form (2.17). In this case, it follows from (2.4) and (2.7) that $\mathcal{A}_{\neq\mu}$ and $\mathcal{A}_{\neq\mu,\mu+1}$ themselves can be written as Laplace-like operators of order three. For example, when all M_ν , $\nu = 1, 2, \dots, d$ in (2.17) are identity matrices, then

$$\mathcal{A}_{\neq\mu} = \mathcal{L}_3 \otimes I_{r_\mu} \otimes I_{n_1 \dots n_{\mu-1}} + I_{n_{\mu+1} \dots n_d} \otimes \mathcal{L}_2 \otimes I_{n_1 \dots n_{\mu-1}} + I_{n_{\mu+1} \dots n_d} \otimes I_{r_\mu} \otimes \mathcal{L}_1$$

with

$$\mathcal{L}_1 = \mathbf{U}_{\leq\mu-1}^\top \left(\sum_{\nu=1}^{\mu-1} I_{n_{\mu-1}} \otimes \dots \otimes L_\nu \otimes \dots \otimes I_{n_1} \right) \mathbf{U}_{\leq\mu-1},$$

$\mathcal{L}_2 = L_\mu$, and

$$\mathcal{L}_3 = \mathbf{U}_{\geq\mu+1}^\top \left(\sum_{\nu=\mu+1}^d I_{n_d} \otimes \dots \otimes L_\nu \otimes \dots \otimes I_{n_{\mu+1}} \right) \mathbf{U}_{\geq\mu+1}.$$

Both \mathcal{L}_1 and \mathcal{L}_3 are assembled using low-dimensional contractions. As a consequence of this Laplace-like representation, $\mathcal{A}_{\neq\mu}^{-1}$ and $\mathcal{A}_{\neq\mu,\mu+1}^{-1}$ can be well approximated by a short sum of Kronecker products of exponentials; see section 2.3.1. This approximation is then used as the preconditioner.

In the examples considered in our numerical experiments, \mathbf{A} is always composed of a Laplace-like operator and a potential. Our preconditioner is then based on the Laplace-like part alone.

4. Numerical experiments. To assess the performance of Algorithm 4 (EVAMEn), we calculate the p smallest eigenvalues of the PDE eigenvalue problem

$$(4.1) \quad \begin{aligned} -\Delta u(\mathbf{x}) + V(\mathbf{x})u(\mathbf{x}) &= \lambda u(\mathbf{x}) & \text{for } \mathbf{x} \in \Omega = (a, b)^d, \\ u(\mathbf{x}) &= 0 & \text{for } \mathbf{x} \in \partial\Omega, \end{aligned}$$

where Δ is the d -dimensional Laplace operator, and V is a potential. The finite difference discretization using a regular grid with n grid points in every dimension yields an n^d -dimensional matrix eigenvalue problem

$$\mathbf{A} \mathbf{u} = \lambda \mathbf{u}, \quad \mathbf{A} \in \mathbb{R}^{n^d \times n^d},$$

where $\mathbf{A} = \mathbf{L} + \mathbf{V}$ is composed of the discretized Laplace operator

$$\mathbf{L} = \sum_{\mu=1}^d I_n \otimes \cdots \otimes L \otimes \cdots \otimes I_n$$

and the discretized potential \mathbf{V} . Finding the p smallest eigenvalues of \mathbf{A} then corresponds to the solution of the trace minimization problem (1.1).

Both algorithms, ALS and EVAMEn, have been implemented in MATLAB version 2013b.³ To solve the reduced eigenvalue problems (3.2), we use Knyazev's implementation⁴ of LOBPCG with the stopping tolerance

$$\max\{10^{-6}, \min\{10^{-2}, 10^{-2}\|\mathbf{R}_{\mu,\mu+1}\|\}\}.$$

The preconditioner for LOBPCG is a rank-three approximation of $(\mathcal{U}_{\neq\mu}^T \mathbf{L} \mathcal{U}_{\neq\mu})^{-1}$ by exponential sums; see section 3.3). Similarly, the preconditioner for $\mathcal{A}_{\mu,\mu+1}$ needed in EVAMEn (see (3.17)) is a rank-three approximation of $(\mathcal{U}_{\neq\mu,\mu+1}^T \mathbf{L} \mathcal{U}_{\neq\mu,\mu+1})^{-1}$ by exponential sums.

In our numerical implementation, we use repeatedly truncated singular value decompositions to prevent excessive rank growth. In particular, this concerns the following two points:

- The residual $\mathbf{R}_{\mu,\mu+1}$ (see (3.16)) need for EVAMEn is truncated to rank two before and after the preconditioner $\mathcal{B}_{\mu,\mu+1}^{-1}$ is applied. This is only done when more than one eigenvalue is sought ($p > 1$);
- when shifting the index α (Algorithms 1 and 2) we discard, as suggested in [6, 35], all singular values below a threshold tol_{sv} . We choose a relative tolerance 10^{-8} , but do not allow the ranks to grow larger than 40 in both ALS and EVAMEn.

All computations and timings were performed on a 12-core Intel Xeon CPU X5675, 3.07 GHz with 192 GB RAM running 64-Bit Linux version 2.6.32.

4.1. Setting 1: One eigenvalue for Newton potential. First, we consider the Newton potential

$$V(\mathbf{x}) = \frac{1}{\|\mathbf{x}\|}.$$

We use an exponential sum with ten terms [16, 17] to construct a rank-10 approximation of the discretized potential \mathbf{V} , with an accuracy of $3.6 \cdot 10^{-5}$. This approximate potential is applied to vectors in TT format using the Hadamard product in the TT format. We use $d = 10$, $\Omega = (-1, 1)^d$, and $n = 128$ grid points in every dimension, yielding a discretized eigenvalue problem of size 128^{10} .

As a first experiment we calculate only the smallest eigenvalue, that is, $p = 1$. In this case, no shifting of the index α is performed and therefore ALS is unable to adapt the ranks. We initialize ALS with a random block-1 TT tensor having all ranks equal to $r_\mu = 8$, corresponding to the ranks obtained by EVAMEn, which is initialized with a random block-1 TT tensor. Figure 4 shows the obtained convergence in terms of the eigenvalue error (dashed lines) and the residual (solid lines), both with respect to the number of iterations and with respect to execution time.

³The latest version of our code is available at <http://anchp.epfl.ch/TTeMPS>.

⁴Available at <http://www.mathworks.com/matlabcentral/fileexchange/48-lobpcg-m>.

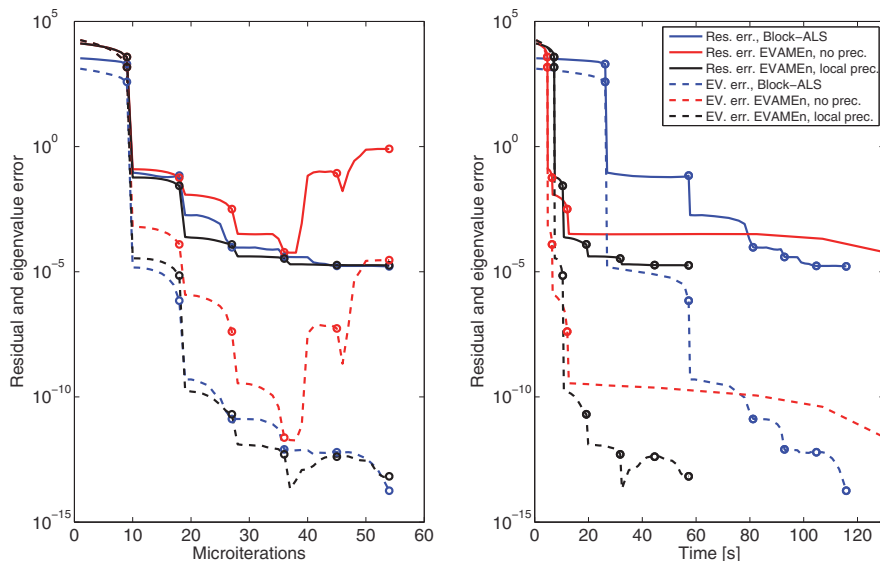


FIG. 4. Comparison of ALS (blue lines) to EVAMEn without preconditioning (red lines) and with preconditioning (black lines) for calculating the smallest eigenvalue of (4.1) using the Newton potential, $\Omega = (-1, 1)^{10}$, and $n = 128$ grid points in every dimension. Each marker indicates the completion of a half-sweep.

Surprisingly, the preconditioned version (3.17) of EVAMEn makes almost identical progress per iteration as ALS, but the first few half-sweeps take significantly less time. The latter is explained by the fact that EVAMEn operates with smaller TT ranks for the first few half-sweeps.

It can be clearly observed that preconditioning the residual $\mathbf{R}_{\mu, \mu+1}$ is important, as the convergence of the unpreconditioned version (3.16) is significantly worse. The errors even start growing for later iterations, due to convergence failures of LOBPCG. This experiment allows us to conclude that the choice of the core augmentation in EVAMEn matters, and should not be taken randomly, which would already allow for rank adaptivity. Therefore, we only consider the preconditioned version of EVAMEn in the following experiments.

4.2. Setting 2: Multiple eigenvalues for Newton potential. This setting is identical to Setting 1, except that we now seek for the $p > 1$ smallest eigenvalues. In this case, shifting the index α provides a mechanism for rank adaption in both ALS and EVAMEn. Therefore, ALS is now also initialized with a random rank-1 block-1 TT tensor.

We compare ALS with EVAMEn for $p = 3$ and $p = 11$ in Figures 5 and 6, respectively. Solid lines correspond to the scaled Frobenius norm of the residual, $p^{-1/2} \|\mathbf{A}\mathbf{X} - \mathbf{X}\mathbf{\Lambda}\|_F$, while dashed lines correspond to the trace error, $|\text{trace}(\mathbf{X}^\top \mathbf{A}\mathbf{X}) - \text{trace}(\mathbf{X}_{\text{final}}^\top \mathbf{A}\mathbf{X}_{\text{final}})|$, where $\mathbf{X}_{\text{final}}$ is a reference solution computed by the ALS procedure with high accuracy.

Since $d = 10$, the second largest eigenvalue of the discrete Laplace operator has multiplicity 10 and therefore the eigenvalues $\lambda_2, \dots, \lambda_{11}$ of \mathbf{A} form a cluster. This cluster is broken when using $p = 3$, resulting in very slow convergence of the LOBPCG iterations when no local preconditioner is used.

In both cases, EVAMEn outperforms ALS in terms of both required number of

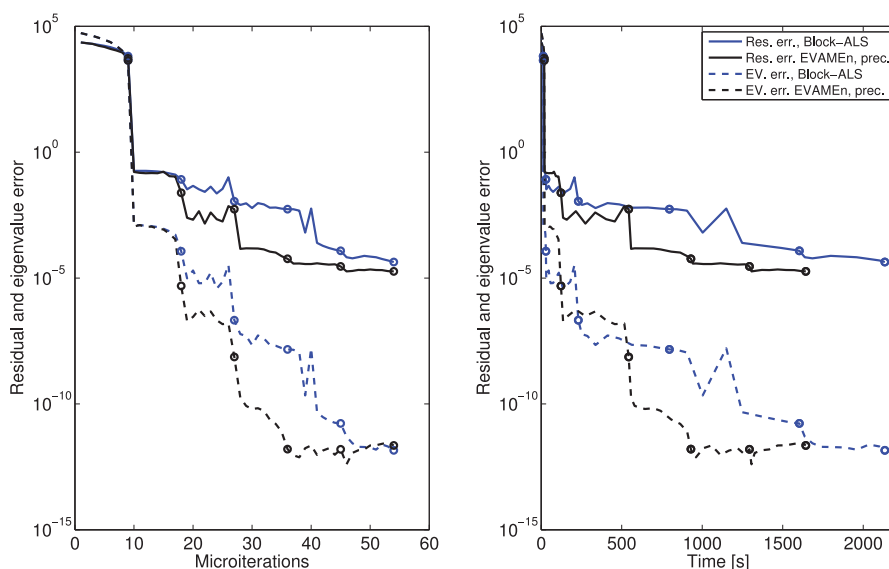


FIG. 5. Comparison of block ALS (blue lines) and preconditioned EVAMEn (black lines) for calculating the three smallest eigenvalues of (4.1) using the Newton potential, $\Omega = (-1, 1)^{10}$, and $n = 128$ grid points in every dimension.

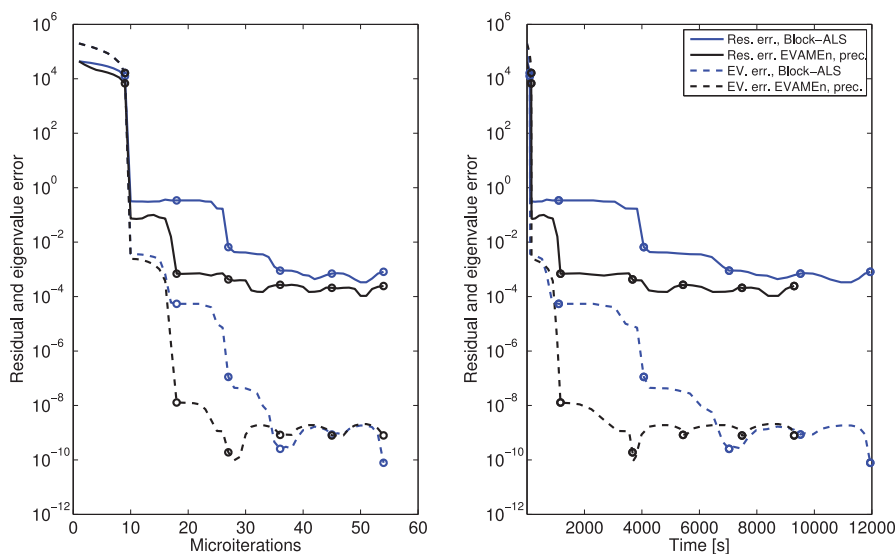


FIG. 6. Comparison of block ALS (blue lines) and preconditioned EVAMEn (black lines) for calculating the eleven smallest eigenvalues of (4.1) using the Newton potential, $\Omega = (-1, 1)^{10}$, and $n = 128$ grid points in every dimension.

iterations and computational time. The final TT ranks in the block format of the solutions are bounded by 26 for $p = 3$, and attain the allowed limit of 40 for $p = 11$. Consequently, the obtained final accuracy for $p = 11$ is less than for $p = 3$.

4.3. Setting 3: Henon–Heiles potential. As a second test case, we take the modified Henon–Heiles potential [12, 31, 36]

$$V(\mathbf{x}) = \frac{1}{2} \sum_{\mu=1}^d x_{\mu}^2 + \sum_{\mu=1}^{d-1} \left(\sigma_* \left(x_{\mu} x_{\mu+1}^2 - \frac{1}{3} x_{\mu}^3 \right) + \frac{\sigma_*^2}{16} (x_{\mu}^2 + x_{\mu+1}^2)^2 \right),$$

modeling a coupled oscillator. This potential is usually defined on the entire real space. As in [6], we apply spectral collocation, using a tensor product grid based on the zeros ξ_1, \dots, ξ_n of the n th Hermite polynomial. The corresponding discrete one-dimensional Laplace operator is given by [3]

$$(L_H)_{ij} = \begin{cases} \frac{1}{6}(4n-1-2\xi_i^2), & i=j, \\ (-1)^{(i-j)} \left(\frac{1}{(\xi_i-\xi_j)^2} - \frac{1}{2} \right), & i \neq j. \end{cases}$$

The discretized matrix \mathbf{A} takes the form

$$(4.2) \quad \sum_{\mu=1}^d I_n \otimes \cdots \otimes I_n \otimes L_{\mu} \otimes I_n \otimes \cdots \otimes I_n + \sum_{\mu=1}^{d-1} I_n \otimes \cdots \otimes C_{\mu+1} \otimes B_{\mu} \otimes \cdots \otimes I_n,$$

with $B_{\mu} = \sigma_* D + \frac{\sigma_*^2}{8} D^2$, $C_{\mu+1} = D^2$, and

$$L_{\mu} = \begin{cases} L_H + \frac{1}{2} D^2 - \frac{\sigma_*}{3} D^3 + \frac{\sigma_*^2}{16} D^4 & \text{for } \mu = 1, \\ L_H + \frac{1}{2} D^2 - \frac{\sigma_*}{3} D^3 + \frac{\sigma_*^2}{8} D^4 & \text{for } 1 < \mu < d, \\ L_H + \frac{1}{2} D^2 + \frac{\sigma_*^2}{16} D^4 & \text{for } \mu = d, \end{cases}$$

where $D = \text{diag}(\xi_1, \xi_2, \dots, \xi_n)$ contains the grid points for one dimension. Hence \mathbf{A} allows for an operator TT representation with all ranks equal to 3 using the cores described in section 2.3.2.

The first term in (4.2), which is composed of the discrete Laplacian \mathbf{L} and the noninteracting parts of the potential, is taken as an approximation \mathbf{B} of \mathbf{A} for the local preconditioner.

In our experiments we choose $\sigma_* = 0.11$, and consider (4.1) with $d = 10$ and $n = 28$ collocation points in every dimension. Thus, the discretized eigenvalue problem is of dimension 28^{10} .

Figure 7 shows the obtained results when calculating $p = 3$ eigenvalues. Similarly to the Newton potential, EVAMEn performs better than ALS in terms of the number of iterations. Since this is offset by the higher cost per iteration, both algorithms perform equally well with respect to time. For $p = 11$, EVAMEn converges only slightly better and the obtained timings are similar for both algorithms; see Figure 8. Possibly, the local nature of the potential benefits ALS to an extent that it can hardly be improved by injecting residual information.

5. Conclusion. We have developed a new algorithm called EVAMEn, which allows us to incorporate preconditioned residual information into the ALS procedure for computing one or several eigenvectors in the (block) TT format. For the Newton potential, the numerical experiments clearly demonstrate the benefits obtained from incorporating this information and the importance of using local preconditioners. We

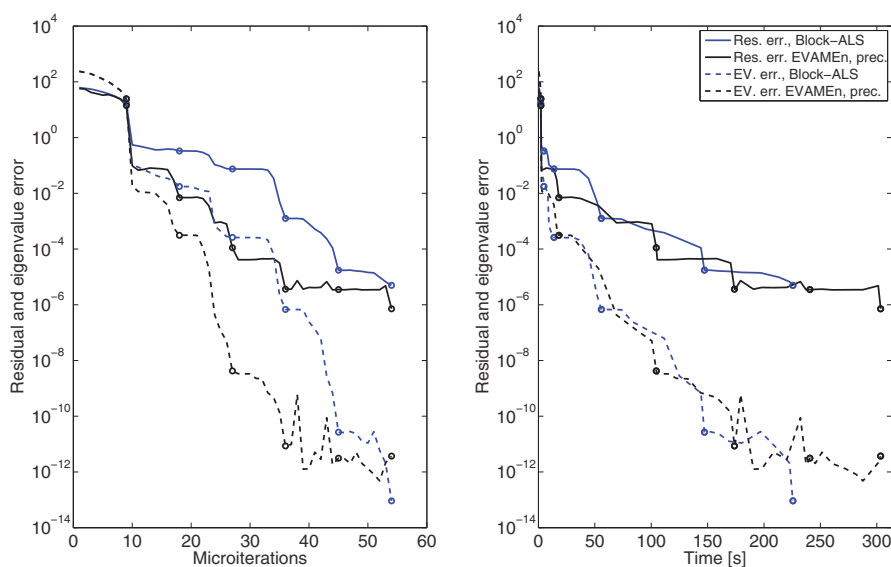


FIG. 7. Comparison of block ALS (blue lines) and EVAMEn (black lines) for calculating the three smallest eigenvalues of (4.1) using the Henon–Heiles potential with $d = 10$ and $n = 28$ collocation points in every dimension.

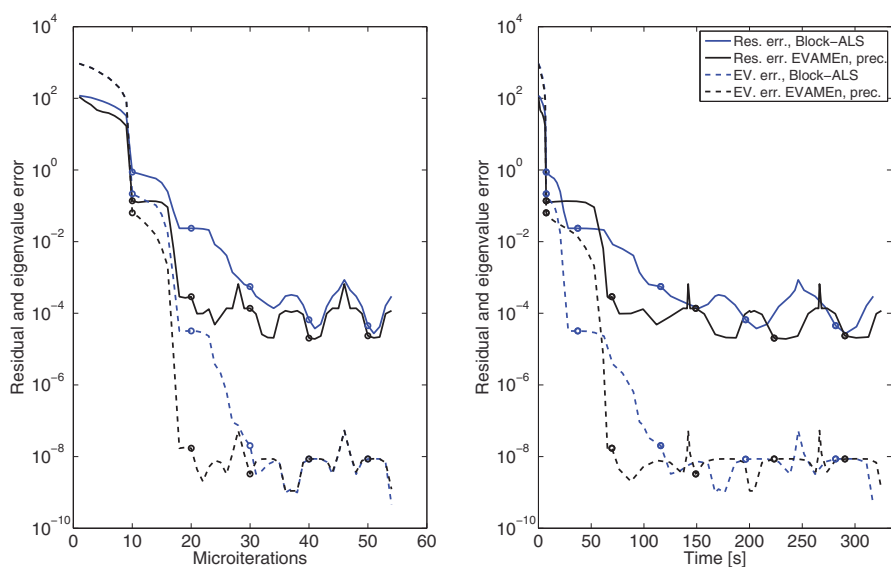


FIG. 8. Comparison of block ALS (blue lines) and EVAMEn (black lines) for calculating the eleven smallest eigenvalues of (4.1) using the Henon–Heiles potential with $d = 10$ and $n = 28$ collocation points in every dimension.

expect that using such preconditioners would also be beneficial in the AMEn algorithm for solving linear systems.

The use of the TT format makes it expensive to work with large mode sizes n_μ , as n_μ enters directly into the size of the TT cores $\mathbf{U}_\mu \in \mathbb{R}^{r_{\mu-1} \times n_\mu \times r_\mu}$. Unless the ranks are very small, this imposes computational restrictions on n_μ and hence on

the discretization. A simple way to avoid this is to use low-dimensional subspaces of \mathbb{R}^{n_μ} in every mode. This has been considered for the TT format in [33] and it is commonplace for the hierarchical Tucker (HT) format [19, 14]. Concerning the latter, it is certainly possible to extend the block TT format to the HT format, but the derivations and the resulting algorithms can be expected to become significantly more technical. We refer to [29] for a discussion on the ALS and DMRG algorithms for computing a single eigenvector in the HT format.

Acknowledgment. We thank F. Verstraete for pointing out the references [35] and [43].

REFERENCES

- [1] P.-A. ABSIL, R. MAHONY, AND R. SEPULCHRE, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, NJ, 2008.
- [2] P. A. ABSIL, R. MAHONY, R. SEPULCHRE, AND P. VAN DOOREN, *A Grassmann-Rayleigh quotient iteration for computing invariant subspaces*, SIAM Rev., 44 (2002), pp. 57–73.
- [3] D. BAYE AND P.-H. HEENEN, *Generalised meshes for quantum mechanical problems*, J. Phys. A, 19 (1986), pp. 2041–2059.
- [4] J. H. BRAMBLE, J. E. PASCIAK, AND A. V. KNYAZEV, *A subspace preconditioning algorithm for eigenvector/eigenvalue computation*, Adv. Comput. Math., 6 (1996), pp. 159–189.
- [5] E. CANCÈS, V. EHRLACHER, AND Y. MADAY, *Periodic Schrödinger operators with local defects and spectral pollution*, SIAM J. Numer. Anal., 50 (2012), pp. 3016–3035.
- [6] S. V. DOLGOV, B. N. KHOROMSKIJ, I. V. OSELEDETS, AND D. V. SAVOSTYANOV, *Computation of extreme eigenvalues in higher dimensions using block tensor train format*, Comput. Phys. Comm., 185 (2014), pp. 1207–1216.
- [7] S. V. DOLGOV, B. N. KHOROMSKIJ, I. V. OSELEDETS, AND E. E. TYRTYSHNIKOV, *Low-rank tensor structure of solutions to elliptic problems with jumping coefficients*, J. Comput. Math., 30 (2012), pp. 14–23.
- [8] S. V. DOLGOV AND D. V. SAVOSTYANOV, *Alternating Minimal Energy Methods for Linear Systems in Higher Dimensions. Part I: SPD Systems*, preprint, arXiv:1301.6068, 2013.
- [9] S. V. DOLGOV AND D. V. SAVOSTYANOV, *Alternating Minimal Energy Methods for Linear Systems in Higher Dimensions. Part II: Faster Algorithm and Application to Nonsymmetric Systems*, preprint, arXiv:1304.1222, 2013.
- [10] S. V. DOLGOV AND D. V. SAVOSTYANOV, *One-Site Density Matrix Renormalization Group and Alternating Minimum Energy Algorithm*, preprint, arXiv:1312.6542, 2013.
- [11] A. EDELMAN, T. A. ARIAS, AND S. T. SMITH, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 303–353.
- [12] E. FAOU, V. GRADINARU, AND C. LUBICH, *Computing semiclassical quantum dynamics with Hagedorn wavepackets*, SIAM J. Sci. Comput., 31 (2009), pp. 3027–3041.
- [13] L. GRASEDYCK, *Existence and computation of low Kronecker-rank approximations for large linear systems of tensor product structure*, Computing, 72 (2004), pp. 247–265.
- [14] L. GRASEDYCK, *Hierarchical singular value decomposition of tensors*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2029–2054.
- [15] L. GRASEDYCK, D. KRESSNER, AND C. TOBLER, *A literature survey of low-rank tensor approximation techniques*, GAMM Mitt. Ges. Angew. Math. Mech., 36 (2013), pp. 53–78.
- [16] W. HACKBUSCH, *Entwicklungen nach Exponentialsummen*, Technical report 4/2005, MPI MIS, Leipzig, Germany, 2010.
- [17] W. HACKBUSCH, *Tensor Spaces and Numerical Tensor Calculus*, Springer, Heidelberg, 2012.
- [18] 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 (2012), pp. 133–151.
- [19] W. HACKBUSCH AND S. KÜHN, *A new scheme for the tensor representation*, J. Fourier Anal. Appl., 15 (2009), pp. 706–722.
- [20] R. J. HARRISON, G. I. FANN, T. YANAI, Z. GAN, AND G. BEYLKIN, *Multiresolution quantum chemistry: Basic theory and initial applications*, J. Chem. Phys., 121 (2004), pp. 11587–11598.
- [21] S. HOLTZ, T. ROHWEDDER, AND R. SCHNEIDER, *The alternating linear scheme for tensor optimization in the tensor train format*, SIAM J. Sci. Comput., 34 (2012), pp. A683–A713.

- [22] T. HUCKLE AND K. WALDHERR, *Subspace iteration methods in terms of matrix product states*, PAMM Proc. Appl. Math. Mech., 12 (2012), pp. 641–642.
- [23] V. A. KAZEYEV AND B. N. KHOROMSKIY, *Low-rank explicit QTT representation of the Laplace operator and its inverse*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 742–758.
- [24] V. KHOROMSKAIA, *Black-box Hartree–Fock solver by tensor numerical methods*, Comput. Methods Appl. Math., 14 (2014), pp. 89–111.
- [25] B. N. KHOROMSKIY, V. KHOROMSKAIA, AND H.-J. FLAD, *Numerical solution of the Hartree–Fock equation in multilevel tensor-structured format*, SIAM J. Sci. Comput., 33 (2011), pp. 45–65.
- [26] B. N. KHOROMSKIY AND I. V. OSELEDETS, *DMRG+QTT Approach to Computation of the Ground State for the Molecular Schrödinger Operator*, Technical report 69/2010, MPI MIS, Leipzig, Germany, 2010.
- [27] B. N. KHOROMSKIY AND I. V. OSELEDETS, *Quantics-TT collocation approximation of parameter-dependent and stochastic elliptic PDEs*, Comput. Methods Appl. Math., 10 (2010), pp. 376–394.
- [28] A. V. KNYAZEV, *Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comput., 23 (2001), pp. 517–541.
- [29] D. KRESSNER AND C. TOBLER, *Preconditioned low-rank methods for high-dimensional elliptic PDE eigenvalue problems*, Comput. Methods Appl. Math., 11 (2011), pp. 363–381.
- [30] O. S. LEBEDEVA, *Tensor conjugate-gradient-type method for Rayleigh quotient minimization in block QTT-format*, Russian J. Numer. Anal. Math. Modelling, 26 (2011), pp. 465–489.
- [31] H.-D. MEYER, U. MANTHE, AND L.-S. CEDERBAUM, *The multiconfigurational time-dependent Hartree approach*, Chem. Phys. Lett., 165 (1990), pp. 73–78.
- [32] K. NEYMEYER, *A geometric theory for preconditioned inverse iteration applied to a subspace*, Math. Comp., 71 (2002), pp. 197–216.
- [33] I. V. OSELEDETS, *Tensor-train decomposition*, SIAM J. Sci. Comput., 33 (2011), pp. 2295–2317.
- [34] 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 (2009), pp. 3744–3759.
- [35] I. PIŻORN AND F. VERSTRAETE, *Variational numerical renormalization group: Bridging the gap between NRG and density matrix renormalization group*, Phys. Rev. Lett., 108 (2012), 067202.
- [36] A. RAAB AND H.-D. MEYER, *A numerical study on the performance of the multiconfigurational time-dependent Hartree method for density operators*, J. Chem. Phys., 112 (2000), pp. 10718–10729.
- [37] T. ROHWEDDER AND A. USCHMAJEV, *On local convergence of alternating schemes for optimization of convex problems in the tensor train format*, SIAM J. Numer. Anal., 51 (2013), pp. 1134–1162.
- [38] A. H. SAMEH AND Z. TONG, *The trace minimization method for the symmetric generalized eigenvalue problem*, J. Comput. Appl. Math., 123 (2000), pp. 155–175.
- [39] A. H. SAMEH AND J. A. WISNIEWSKI, *A trace minimization algorithm for the generalized eigenvalue problem*, SIAM J. Numer. Anal., 19 (1982), pp. 1243–1259.
- [40] U. SCHOLLWÖCK, *The density-matrix renormalization group in the age of matrix product states*, Ann. Physics, 326 (2011), pp. 96–192.
- [41] A. WEICHSELBAUM, F. VERSTRAETE, U. SCHOLLWÖCK, J. I. CIRAC, AND J. VON DELFT, *Variational matrix-product-state approach to quantum impurity models*, Phys. Rev. B, 80 (2009), 165117.
- [42] S. R. WHITE, *Density matrix formulation for quantum renormalization groups*, Phys. Rev. Lett., 69 (1992), pp. 2863–2866.
- [43] S. R. WHITE, *Density matrix renormalization group algorithms with a single center site*, Phys. Rev. B, 72 (2005), 180403.