Max-Planck-Institut für Mathematik in den Naturwissenschaften Leipzig

Computing eigenspaces with low rank constraints

(revised version: February 2021)

by

Christian Krumnow, Max Pfeffer, and André Uschmajew

Preprint no.: 102 2019

Computing eigenspaces with low rank constraints

Christian Krumnow* Max Pfeffer* André Uschmajew*

Abstract

In this work, the task to find a simultaneous low rank approximation of several lowest eigenpairs of a matrix-valued symmetric operator is considered. This problem arises for instance in the density matrix renormalization group algorithm (DMRG) for the accurate simulation of quantum chains. The usual approach is to compute the desired set of eigenvectors and then to identify a low rank approximation by truncating the singular value decomposition (SVD). Since SVD truncation is a norm projection, this yields only sub-optimal results for the eigenvalues. As an alternative, this article explores a direct trace minimization on the intersection of the Stiefel and a low rank manifold using a Riemannian optimization method, which gives better approximations to the eigenspace. A second algorithm based on alternating optimization is also considered but it is less stable. Compared to SVD truncation, the proposed Riemannian method can be seen as a more natural choice for a sub-solver in the DMRG algorithm, and appears to yield better results when applied to spin chains for which the singular values of the exact eigenvectors decay only moderately.

1 Introduction

The approximation of low lying eigenvectors of Hamiltonians derived or inspired from the electronic Schrödinger equation is one of the key tasks in modern numerical physics to which a considerable amount of computation power is assigned worldwide. There exist several methods to tackle this problem, emphasizing different aspects of it, see for instance [4, 35, 19, 14, 33]. Among them, the *density matrix renormalization group* (DMRG), developed by White [39] and later connected to *matrix product states* (MPS), is one of the most reliable methods for the computation of low energies of large systems with limited entanglement, see the overview articles [34, 36].

In essence, the DMRG can be seen as an algorithm for constructing low rank tensor approximations to the solutions of high-dimensional eigenvalue problems. In the last decade, this theory has therefore also aroused interest in numerical mathematics with the development of similar low rank tensor train (TT) and hierarchical Tucker (HT) formats, and corresponding tensor calculus for high-dimensional problems [13, 30, 29, 20, 12, 21]. In this viewpoint, the DMRG algorithm is a modification of the "workhorse" algorithm [23] called *alternating least squares* or *alternating linear scheme* (ALS) when applied to the Rayleigh quotient, see [16]. While a general convergence theory is still missing or – rather – unpromising, the benefits of the procedure have not been deminished.

*Max Planck Institute for Mathematics in the Sciences, 04103 Leipzig, Germany;
christian.krumnow@mis.mpg.de, max.pfeffer@mis.mpg.de, uschmajew@mis.mpg.de

Many different aspects of the DMRG algorithm and its variants have been investigated in the literature, see e.g. [40, 34]. In [6], a notable alteration has been made in order to allow for the computation not only of the ground state, but of a number of excited states at the same time, which is accompanied by earlier developments and ideas in the physics community [26, 31]. The algorithm proposed in [6] uses the so-called block TT format for representing a low rank orthonormal basis of the eigenspace and can be made rank adaptive even in the single site version. An extension of this approach that also includes preconditioning has been proposed in [24] by transferring the ideas of the alternating minimal energy (AMEn) algorithm from [7, 8] to eigenvalue problems.

A common feature of most tensor methods for eigenvalue problems is a procedure for rank truncation via the singular value decomposition (SVD) during the iteration. This has the advantage of being easily implemented and computable. However, while SVD truncation is known to provide optimal low rank approximations with respect to the Frobenius and other unitarily invariant norms, the same is not necessarily true with respect to the energy. In this work, we address this potential drawback. We propose a direct energy minimization of the two site DMRG subproblems on a low rank matrix manifold with orthogonality constraint.

In the recent work [32], a Jacobi-Davidson type method for the minimization of the energy function on the fixed rank matrix manifold has been introduced for computing the minimal eigenstate. A main observation for this is that the manifold of fixed rank matrices intersects the unit sphere transversally, so that energy minimization can be performed naturally on the intersection using Riemannian optimization. In this work, we show that a transversality property also holds in the case of several low energy states, where the eigenvectors are represented as low rank matrices that are pairwise orthonormal to each other. This leads to a minimization on the intersection of a Stiefel manifold with a manifold of fixed rank matrices. We develop a simple Riemannian conjugate gradient method for trace minimization on this intersection manifold and compare it to alternative methods that are however inferior. One of these methods consists of a novel variation of the block ALS [24] where the dependency on the dimension of the eigenspace stays fixed with one component. This approach seems straightforward, but it leads to complications both in implementation as well as in the results. The Riemannian approach can be improved upon in the future, e.g. by incorporating a subspace acceleration like in the Jacobi-Davidson method, which is not considered here.

Our fixed rank matrix solver is then applied to tensor problems by using it in the DMRG substeps where we consistently minimize the energy function. In order to choose the TT ranks adaptively, we propose a heuristic that is based on the residual of the energy function rather than on the Frobenius norm. While this method does not significantly outperform the DMRG algorithm in problems of good low rank approximability, we show that this somewhat *natural* solver is in principle applicable.

The outline is as follows. In Sec. 2 and 3 a particular low rank format with orthogonality constraints for representing eigenspaces is introduced together with a Riemannian solver for the energy minimization in this format. These considerations are restricted to the real case to avoid some technical subtleties of complex analytic manifolds. However, the methods are expected to be applicable to the complex case without further obstructions. In Sec. 4 the modified tensor DMRG algorithm is explained, whereas Sec. 5 presents some numerical experiments, both for the matrix and tensor case.

2 Low rank eigenspaces

We consider a self-adjoint operator $\mathbf{H}$ on the linear space of real $m \times n$ matrices. Our goal is to compute the eigenspace belonging to the smallest p eigenvalues $\lambda_1, \ldots, \lambda_p$ of $\mathbf{H}$. Such a task arises, for instance, after discretization of an analytic eigenvalue problem on a two-dimensional rectangular domain with a self-adjoint Hamiltonian. Note that in the case $\lambda_p = \lambda_{p+1}$, this eigenspace is not unique and we only compute one of the eigenspaces, the minimization of the energy being the main goal.

After a reordering of the indices, we may treat $\mathbf{H}$ as a real symmetric $mn \times mn$ matrix. Then the goal can be achieved by solving the classical trace minimization problem

$$\text{minimize } f(X) = \frac{1}{2}\mathrm{tr}(X^\mathsf{T}\mathbf{H}X) \quad \text{subject to} \quad X^\mathsf{T}X = I_p \tag{2.1}$$

for $X \in \mathbb{R}^{mn\times p}$, where I_p denotes the $p \times p$ identity matrix. Hence this is an optimization problem on the real Stiefel manifold

$$\mathrm{St}(mn, p) = \{X \in \mathbb{R}^{mn\times p} : X^\mathsf{T}X = I_p\}.$$

For $X \in \mathrm{St}(mn, p)$, the function value $f(X)$ will also be referred to as the energy of X. Note that f is orthogonally invariant in the sense that

$$f(XQ) = f(X) \tag{2.2}$$

for any orthogonal $Q \in \mathbb{R}^{p\times p}$. This reflects the fact that the energy only depends on the space X, and not on a particular chosen basis.

The columns of a minimizer X therefore yield an orthonormal basis of the desired eigenspace, and the minimum function value equals one half of the sum of the p smallest eigenvalues of $\mathbf{H}$. For computing the actual eigenvectors of $\mathbf{H}$, and not just the subspace spanned by them, the $p \times p$ matrix $X^\mathsf{T}\mathbf{H}X$ needs to be diagonalized, but this additional step will not be considered here.

For various reasons it can be required that the matrices $X_1, \ldots, X_p \in \mathbb{R}^{m\times n}$ represented by the columns of X, that is, the basis vectors for the eigenspace, have low rank, or can be well approximated by matrices of low rank. Then, a corresponding constraint needs to be added to the optimization problem (2.1). In Sec. 4 we consider the DMRG algorithm in which such optimization problems arise as a substep. Apart from this, computing the exact eigenspaces can be numerically prohibitive for large m and n and a restriction on the ranks of the representing basis elements can alleviate this problem.

In this work, we consider a slightly more restrictive but ultimately more relevant constraint, namely to find an approximate minimal eigenspace in which *all* matrices either share a common column space or a common row space of low dimension. Since these are linear conditions, it suffices to require them for the basis elements $X_1, \ldots, X_p$. We fix a rank $r \leq \min(m, n)$ and denote with $[X]_m^{np} = (X_1 \cdots X_p) \in \mathbb{R}^{m\times np}$ and $[X]_{mp}^n = (X_1^\mathsf{T} \cdots X_p^\mathsf{T})^\mathsf{T} \in \mathbb{R}^{mp\times n}$ the *matricizations* (reshapes) of X into the indicated format. We say that

$$(X_1 \cdots X_p) = [X]_m^{np} = UV^\mathsf{T} = (UV_1^\mathsf{T} \cdots UV_p^\mathsf{T}), \quad U \in \mathbb{R}^{m\times r}, V \in \mathbb{R}^{np\times r}, \tag{2.3}$$

is a *rank-r column decomposition*. Hence in such a decomposition all basis elements

$X_1, \dots, X_p$ share the same column space spanned by U. Likewise, a decomposition

$$\begin{pmatrix} X_1 \\ \vdots \\ X_p \end{pmatrix} = [X]_{mp}^{n} = UV^{\mathsf{T}} = \begin{pmatrix} U_1 V^{\mathsf{T}} \\ \vdots \\ U_p V^{\mathsf{T}} \end{pmatrix}, \quad U \in \mathbb{R}^{mp \times r},\, V \in \mathbb{R}^{n \times r}, \tag{2.4}$$

will be called a *rank-r row decomposition*, since $X_1, \dots, X_p$ share the row space spanned by V. The existence of such decompositions is equivalent to the conditions

$$\operatorname{rank}([X]_{m}^{np}) \leq r \quad \text{or} \quad \operatorname{rank}([X]_{mp}^{n}) \leq r, \tag{2.5}$$

respectively. We note that these are effectively conditions on the first two multilinear ranks of X when regarded as an $m \times n \times p$ tensor.

Example 2.1. Let $m = n$ and $L \in \mathbb{R}^{n \times n}$ be a symmetric matrix with eigenvalues $\lambda_1 < \ldots < \lambda_n$, and corresponding eigenvectors $u_1, \dots, u_n \in \mathbb{R}^n$. Then the eigenvalues of the operator

$$\mathbf{H} = L \otimes I_n + I_n \otimes L \in \mathbb{R}^{n^2 \times n^2}$$

are $\lambda_i + \lambda_j$ corresponding to rank-one eigenvectors $u_i \otimes u_j$. Note that for example the discretization of the two-dimensional Laplace operator on a rectangular grid leads to an operator $\mathbf{H}$ of this form. Consider for example $p = 3$, that is, the three-dimensional eigenspace that minimizes (2.1). This space is spanned by the eigenvectors $u_1 \otimes u_1$, $u_1 \otimes u_2$, and $u_2 \otimes u_1$ and the value of the trace is $2\lambda_1 + 2(\lambda_1 + \lambda_2)$. It can be seen that u_1 and u_2 span both the row space and the column space of the eigenspace. Thus, while the dimension of the eigenspace is three, all matrices in it only have (at most) two-dimensional column and row spaces. Therefore, any minimizer X of (2.1) will satsify the rank conditions (2.5) with $r = 2$ and allow for rank-two column and row decompositions. If $\lambda_1 + \lambda_3 > 2\lambda_2$, then the fourth eigenvector will be $u_2 \otimes u_2$, so even in the case $p = 4$ we can find the solution with $r = 2$.

If one of the conditions (2.5) is added to the minimization problem (2.1), several approaches for its solution can be considered. We first discuss two strategies from the literature (see e.g. [34, 6]) before presenting two new approaches in the next section, Sec. 3. We conclude this section by discussing the resdiual of the eigenvalue problem as potential error measure for a given low rank approximation of the desired eigenspaces.

2.1 Truncated eigenspaces

In almost all applications we are aware of, in particular the ones involving the DMRG, the procedure of choice is to first calculate a basis for the eigenspace, or even the p eigenvectors, without constraints using a standard solver, and then truncating the rank in one of the above formats using the singular value decomposition (SVD). This method of *truncated eigenspaces* has two immediate drawbacks. First of all, it is necessary to calculate the full solution in the space $\mathbb{R}^{m \times n}$, which might be prohibitive if m and n are large. More importantly, even for smaller problems, the rank truncation by SVD is not a natural method of choice. It constitutes a projection (best rank-r approximation) for various matrix norms, but does not necessarily yield the minimal energy in the sense of cost function f under the rank constraint. Note that an SVD truncation of the matricizations $[X]_{m}^{np}$ or $[X]_{mp}^{n}$ will not preserve the orthogonality condition $X^{\mathsf{T}}X = I_p$ itself, so before evaluating the resulting trace energy an additional reorthogonalization is required (which does not affect the low rank structure).

2.2 Block ALS approach

A second approach that is closely related to the block ALS (single site DMRG) method switches between the column and row decomposition in (2.3) and (2.4). Choosing a column decomposition of X we keep U fixed and only update V such that f is minimal for the current U. The optimal V is given as an eigenspace of the p lowest eigenvalues of the 'projected' operator $(U^\mathsf{T} \otimes I_n)\mathbf{H}(U \otimes I_n)$. Computing an SVD truncation of $[V]_n^{rp}$ of rank r then allows to switch to a row decomposition in order to update U (for now fixed V) accordingly, and so forth.

It should be mentioned that when switching from one decomposition to the other, one could abstain from truncating the rank at all, or truncating it at a higher rank $\tilde{r} \leq \min(rp, n)$, thus gaining some kind of rank adaptivity in case $p \geq 2$, as noted in [6].

Such a block ALS method produces in every iteration a row or column decomposition respecting the orthogonality constraint. However, similar to truncated eigenspaces, rank truncation via SVD is again not optimal and if truncation is omitted when switching the format, the ranks will grow. In addition, the switching between the two formats can lead to a non-monotonic behavior of the cost function during the optimization. The whole procedure is only reasonable if the eigenspace admits both the low rank row and column decomposition, at least approximately, as is the case for the important class of (perturbations of) Laplace-like operators in Example 2.1.

For tensor problems in the TT format, the block ALS approach becomes the block TT method proposed in [6]. Here the switching from column to row rank decomposition in a subproblem naturally provides initializations for the neighboring subproblem and hence allows sweeping through the factors when optimizing the energy of the TT decomposition. This is explained in more detail Sec. 4.2. In the numerical experiments for computing excited states using TT tensors in Sec. 5.2, we will use this block ALS approach, as it is a standard benchmark, but we refrain from using it in the matrix case in Sec. 5.1, where in our experiments it showed inconsistent behavior in switching between the row and column decompositions.

2.3 Residual

Provided with a (low rank) approximation $X \in \mathrm{St}(mn, p)$ of an eigenspace basis, we employ the residual of the eigenvalue problem in order to certify its accuracy. The residual is defined as

$$\mathrm{Res}(X, \mathbf{H}) = \mathbf{H}X - X(X^\mathsf{T}\mathbf{H}X). \tag{2.6}$$

In Sec. 3.1 below, we introduce the tangent space $T_X\mathrm{St}(mn, p)$ and a projection $\mathcal{P}_{T_X\mathrm{St}(mn,p)}$ onto it. One can then show that $\mathrm{Res}(X, \mathbf{H})$ equals the projected gradient of the initial cost function f onto the tangent space $T_X\mathrm{St}(mn, p)$, that is, $\mathrm{Res}(X, \mathbf{H}) = \mathcal{P}_{T_X\mathrm{St}(mn,p)}(\nabla f(X)) = \mathcal{P}_{T_X\mathrm{St}(mn,p)}(\mathbf{H}X)$. The residual hence vanishes at critical points of f on $\mathrm{St}(mn, p)$, which is the case if and only if the columns of X form an orthonormal basis of a p-dimensional subspace spanned by eigenvectors of $\mathbf{H}$. The norm of the residual is thus a possible measure of closeness to such a space. If it remains large in the course of the algorithm, the rank r of the approximation might be too low and could be increased. In the Riemannian DMRG pendant of our method for tensors, we use a *local* residual in order to decide whether ranks should be adapted, as will be explained in Sec. 4.3 below.

3 Low rank optimization with orthogonality constraint

In this work we propose two new strategies for the approximation of multiple eigenvectors under low rank constraints. The first one is close in spirit to the block ALS approach, but instead of switching between column and row rank decompositions it keeps the chosen decomposition fixed during the full optimization. The second strategy is a Riemannian optimization method on the intersection of the low rank manifold with orthogonality constraint. Such an approach has been proposed in [32] for a single eigenvector, where the fixed rank manifold is intersected with a sphere. When dealing with several eigenvectors, a Stiefel manifold needs to be considered instead. Our second method is hence an extension of the work [32], and in a sense a conflation with [24].

All appearing norms are the Frobenius norm and all orthogonal projections are with respect to the Frobenius inner product. Additionally, in the remainder of this chapter, we focus on the low rank column decomposition of X. Everything works analogously for the low rank row decomposition with interchanged roles of U and V.

3.1 Semi-Riemannian alternating optimization

In our first approach we decompose X into a low rank column or row format and then minimize the trace function f in (2.1) with respect to the factors U and V in an alternating manner, but without switching the dependency on p (that is, without changing the type of decomposition). For this it is important to understand how to ensure the orthogonality constraint $X^\mathsf{T}X = I_p$. Denoting with $[V]_{rn}^p = \big(\mathrm{vec}(V_1^\mathsf{T}) \cdots \mathrm{vec}(V_p^\mathsf{T})\big) \in \mathbb{R}^{rn\times p}$, it can be seen that the low rank column decomposition is equivalent to

$$X = \begin{pmatrix}\mathrm{vec}(X_1) & \cdots & \mathrm{vec}(X_p)\end{pmatrix} = (U \otimes I_n)\,[V]_{rn}^p, \tag{3.1}$$

where $\mathrm{vec}(\,\cdot\,)$ denotes the vectorization operation with a lexicographical ordering of the indices.[1] We can always assume that U has pairwise orthonormal columns, that is, $U^\mathsf{T}U = I_r$. Then the orthogonality constraint $X^\mathsf{T}X = I_p$ corresponds to the matrices $V_1, \dots, V_p$ being pairwise orthogonal in the Frobenius inner product. Hence, in the column low rank format (2.3) the global orthogonality constraint $X^\mathsf{T}X = I_p$ is equivalent to the two orthogonality conditions

$$U^\mathsf{T}U = I_r \quad \text{and} \quad ([V]_{rn}^p)^\mathsf{T}[V]_{rn}^p = I_p.$$

Regarding the update rules, it follows from (3.1) that for a fixed U obeying $U^\mathsf{T}U = I_r$ the optimal V in (2.1) can be obtained by solving the trace minimization problem for the 'projected' operator $(U^\mathsf{T} \otimes I_n)\mathbf{H}(U \otimes I_n)$, e.g. we can choose the optimal V such that the columns of $[V]_{rn}^p$ are the p lowest eigenvectors of $(U^\mathsf{T} \otimes I_n)\mathbf{H}(U \otimes I_n)$.

The update for U on the other hand, when V is fixed, cannot be turned into an eigenvalue problem, but it can be posed as an optimization problem on a suitable manifold. Namely, by the orthogonality constraint, U is an element of the real Stiefel manifold

[1] We use the lexicographical ordering indicated above troughout the paper. In the alternative `Matlab` notation that uses reverse lexicographical order, (3.1) would read

$$X = \begin{pmatrix}\mathrm{vec}(X_1) & \cdots & \mathrm{vec}(X_p)\end{pmatrix} = (I_n \otimes U)\,[V]_{rn}^p.$$

All other expressions involving vectorized quantities can be easily translated correspondingly.

$\mathrm{St}(m,r) = \{U \in \mathbb{R}^{m\times r} \colon U^\mathsf{T}U = I_r\}$. For updating U we hence try to minimize the function $U \mapsto f((U \otimes I_n)\,[V]_{rn}^p)$ on that manifold, using Riemannian optimization.

Although the Riemannian optimization on Stiefel manifolds is well established, see e.g. [9], we outline its basics ingredients which we will also need do derive for our Riemannian method in the next section.

The manifold $\mathrm{St}(m,r)$ is an embedded submanifold of $\mathbb{R}^{m\times r}$. For the Riemannian methods on $\mathrm{St}(m,r)$ based on the projected Euclidean gradient, one needs to know the *projection* on its tangent space from the ambient space. The tangent space of $\mathrm{St}(m,r)$ is given by

$$T_U\mathrm{St}(m,r) = \{\zeta \in \mathbb{R}^{m\times r} : \zeta^\mathsf{T}U = -U^\mathsf{T}\zeta\}. \tag{3.2}$$

Thus for any $\zeta \in \mathbb{R}^{m\times r}$, the orthogonal projection $\mathcal{P}_{T_U\mathrm{St}(m,r)}(\zeta)$ on $T_U\mathrm{St}(m,r)$ can be computed according to

$$\mathcal{P}_{T_U\mathrm{St}(m,r)}(\zeta) = \zeta - \frac{1}{2}UU^\mathsf{T}\zeta - \frac{1}{2}U\zeta^\mathsf{T}U. \tag{3.3}$$

Furthermore, we need to define a *retraction* and a *vector transport*. A retraction is a smooth map $(U,\zeta) \mapsto \mathcal{R}_U(\zeta)$ on the tangent bundle $T\mathrm{St}(m,r)$ that maps any $\zeta \in T_U\mathrm{St}(m,r)$ to $\mathrm{St}(m,r)$ and satisfies

$$\mathcal{R}_U(\zeta) = U + \zeta + o(\|\zeta\|) \tag{3.4}$$

for $\zeta \to 0$. A possible retraction is obtained by noting that every matrix $Z \in \mathbb{R}^{m\times r}$ possesses a best approximation $\mathcal{P}_{\mathrm{St}(m,r)}(Z)$ in Frobenius norm on the Stiefel manifold $\mathrm{St}(m,r)$, which is given by the orthogonal factor in a polar decomposition of Z, which itself can be obtained from an SVD of Z:

$$\mathcal{P}_{\mathrm{St}(m,r)}(Z) = \hat{U}\hat{V}^T, \qquad Z = \hat{U}\hat{\Sigma}\hat{V}^T, \tag{3.5}$$

see [18, §7.4.4]. This metric projection is unique if Z has full column rank r. In fact, being a projection on a smooth manifold, $\mathcal{P}_{\mathrm{St}(m,r)}$ is a smooth map in a neighborhood (in $\mathbb{R}^{m\times r}$) of any $U \in \mathrm{St}(m,r)$. Moreover, it holds for any $\zeta \in \mathbb{R}^{m\times r}$ that

$$\mathcal{P}_{\mathrm{St}(m,r)}(U+\zeta) = U + \mathcal{P}_{T_U\mathrm{St}(m,r)}(\zeta) + o(\|\zeta\|), \tag{3.6}$$

see, e.g., [27] for the Stiefel manifold or [2] for a general result. In particular, this shows that $\mathcal{P}_{\mathrm{St}(m,r)}$ can be used as a retraction and thus also allows to define an appropriate vector transport, as will be explained below.

The resulting procedure for the Riemannian subiteration on a Stiefel manifold is standard and can be found in many previous works, see e.g. [1, 9]. We employ the `manopt` toolbox in `Matlab` for this task [5].

Since the update for V is an eigenvalue problem, while the update for U requires manifold optimization, we will refer to this alternating optimization (AO) approach for finding a low rank column decomposition of (2.1) as the *semi-Riemannian AO* method. The corresponding method for a low rank row solution is of course analogous.

3.2 Riemannian low rank optimization with orthogonality constraint

When enforcing the column decomposition with a fixed rank r, our goal is to minimize the function $f(X) = \frac{1}{2}\mathrm{tr}(X^\mathsf{T}\mathbf{H}X)$ on the intersection

$$\mathcal{N} = \mathcal{S} \cap \mathcal{M}$$

(or its closure), where

$$\mathcal{S} \coloneqq \mathrm{St}(mn, p)$$

is again the real Stiefel manifold and

$$\mathcal{M} \coloneqq \mathcal{M}_r = \{X \in \mathbb{R}^{mn\times p} : \mathrm{rank}([X]_m^{np}) = r\}$$

is (isomorphic to) a fixed rank matrix manifold. Both are smooth embedded submanifolds in $\mathbb{R}^{mn\times p}$, with $\dim(\mathcal{S}) = mnp - \frac{1}{2}p(p+1)$ and $\dim(\mathcal{M}) = mnp - (m-r)(np-r)$ (for $\mathcal{M}$ see; e.g., [10, p. 27]). We recall from the conditions (2.5) that the intersection $\mathcal{N}$ is not empty if and only if

$$r \leq \min(m, np),$$

which is assumed for the rest of this section.

We first show that the submanifolds $\mathcal{S}$ and $\mathcal{M}$ intersect transversally, implying that $\mathcal{N}$ is itself an embedded submanifold. For this, we recall the tangent spaces of the individual manifolds. The tangent space of the Stiefel manifold at $X \in \mathcal{S}$ is given in (3.2). On the other hand, given an $X \in \mathcal{M}$, we write its *full* SVD as

$$[X]_m^{np} = \mathbf{U}\begin{pmatrix}\Sigma & 0\\ 0 & 0\end{pmatrix}\mathbf{V}^\mathsf{T} = U\Sigma V^\mathsf{T} \tag{3.7}$$

with $\mathbf{U} = \begin{pmatrix}U & U_\perp\end{pmatrix} \in \mathbb{R}^{m\times m}$ and $\mathbf{V} = \begin{pmatrix}V & V_\perp\end{pmatrix} \in \mathbb{R}^{np\times np}$ orthogonal matrices and $\Sigma \in \mathbb{R}^{r\times r}$ diagonal with positive descending entries. One can then write the tangent space of $\mathcal{M}$ as

$$T_X\mathcal{M} = \left\{\xi \in \mathbb{R}^{mn\times p} : [\xi]_m^{np} = \mathbf{U}\begin{pmatrix}\Lambda_1 & \Lambda_2\\ \Lambda_3 & 0\end{pmatrix}\mathbf{V}^\mathsf{T}\right\}. \tag{3.8}$$

To see this, note first that the dimension matches. Further, if one sets $\Lambda_2 = 0$ or $\Lambda_3 = 0$, one obtains two subspaces of matrices of rank at most r containing X, which are hence both subspaces of $T_X\mathcal{M}$.

Proposition 3.1. *At any point $X \in \mathcal{N}$, the orthogonal complement of $T_X\mathcal{M}$ is contained in $T_X\mathcal{S}$. In particular, the tangent spaces $T_X\mathcal{S}$ and $T_X\mathcal{M}$ intersect transversally. As a result, $\mathcal{N}$ is an embedded submanifold of $\mathbb{R}^{mn\times r}$, and the tangent space equals the intersection of the tangent spaces: $T_X\mathcal{N} = T_X\mathcal{S} \cap T_X\mathcal{M}$.*

Proof. The normal of $T_X\mathcal{M}$ of $\mathcal{M}$ consist of all matrices of the form $\xi = U_\perp Z V_\perp^\mathsf{T}$. For such matrices, we obtain $\xi^\mathsf{T}X = 0$. Therefore, the normal space of $T_X\mathcal{M}$ is contained in $T_X\mathcal{S}$. This implies $T_X\mathcal{S} + T_X\mathcal{M} = \mathbb{R}^{mn\times r}$, which is the asserted transversality condition and ensures that the intersection is an embedded submanifold; see, e.g., [10, Ch. 1, §5]. □

The simplest framework to consider in Riemannian optimization is an embedded submanifold equipped with the metric from the ambient space. When restricting a smooth function f from the ambient space to the submanifold, its Riemannian gradient $\mathrm{grad} f$ at a point is then just given by the orthogonal projection of the Euclidean gradient to the tangent space, that is,

$$\mathrm{grad} f(X) \coloneqq \mathcal{P}_{T_X\mathcal{N}}(\nabla f(X)).$$

Note that for our cost function $f(X) = \frac{1}{2}\mathrm{tr}(X^\mathsf{T}\mathbf{H}X)$, the Euclidean gradient (i.e., with respect to the trace inner product) is just

$$\nabla f(X) = \mathbf{H}X.$$

In order to derive the orthogonal projection of any $\zeta \in \mathbb{R}^{mn\times p}$ onto the tangent space $T_X\mathcal{N}$, we need the orthogonal projection $\mathcal{P}_{T_X\mathcal{S}}(\zeta)$ onto the tangent space $T_X\mathcal{S}$ given in (3.3). On the other hand, rewriting

$$[\zeta]_m^{np} = \mathbf{U}\begin{pmatrix}\Omega_1 & \Omega_2\\ \Omega_3 & \Omega_4\end{pmatrix}\mathbf{V}^\mathsf{T},$$

the orthogonal projection onto $T_X\mathcal{M}$ is given by

$$[\mathcal{P}_{T_X\mathcal{M}}(\zeta)]_m^{np} = \mathbf{U}\begin{pmatrix}\Omega_1 & \Omega_2\\ \Omega_3 & 0\end{pmatrix}\mathbf{V}^\mathsf{T}. \tag{3.9}$$

The following result is a direct consequence of Proposition 3.1 and shows that the orthogonal projection onto $T_X\mathcal{N}$ is readily available in practice through the formulas (3.3) and (3.9).

Proposition 3.2. *Let $X \in \mathcal{N}$. The orthogonal projection onto the tangent space $T_X\mathcal{N}$ is given by the composition of the orthogonal projections onto the individual tangent spaces, in any order:*

$$\mathcal{P}_{T_X\mathcal{N}} = \mathcal{P}_{T_X\mathcal{S}} \circ \mathcal{P}_{T_X\mathcal{M}} = \mathcal{P}_{T_X\mathcal{M}} \circ \mathcal{P}_{T_X\mathcal{S}}.$$

Proof. Clearly, both compositions of projections reduce to the identity for elements in $T_X\mathcal{N} = T_X\mathcal{S} \cap T_X\mathcal{M}$. On the other hand, let $\xi \in (T_X\mathcal{N})^\perp$. Then we can decompose $\xi = \xi_1 + \xi_2$ with $\xi_1 \in T_X\mathcal{M} \cap (T_X\mathcal{S})^\perp$ and $\xi_2 \in (T_X\mathcal{M})^\perp$. This implies $(\mathcal{P}_{T_X\mathcal{S}} \circ \mathcal{P}_{T_X\mathcal{M}})(\xi) = 0$. At the same time, by Proposition 3.1, we have $\xi_2 \in T_X\mathcal{S}$, yielding $(\mathcal{P}_{T_X\mathcal{M}} \circ \mathcal{P}_{T_X\mathcal{S}})(\xi) = 0$. Hence, both $\mathcal{P}_{T_X\mathcal{S}} \circ \mathcal{P}_{T_X\mathcal{M}}$ and $\mathcal{P}_{T_X\mathcal{M}} \circ \mathcal{P}_{T_X\mathcal{S}}$ equal the orthogonal projection on $T_X\mathcal{N}$. □

In order to formulate a Riemannian optimization method on $\mathcal{N}$, we are left to define a retraction and with this a vector transport. Due to the special structure of the problem, a retraction can again be obtained by composition as follows.

Proposition 3.3. *Let $\mathcal{R}^\mathcal{M}$ denote any retraction for the manifold $\mathcal{M}$ and $P_\mathcal{S}(Z)$ the metric projection of $Z \in \mathbb{R}^{mn\times p}$ on the manifold $\mathcal{S} = \mathrm{St}(mn, p)$ obtained from the polar decomposition; see* (3.5)*. Then*

$$\mathcal{R}_X(\xi) = \mathcal{P}_\mathcal{S}(\mathcal{R}_X^\mathcal{M}(\xi)), \quad X \in \mathcal{N},\ \xi \in T_X\mathcal{N},$$

defines a retraction on $\mathcal{N}$.

Proof. We first show that $\mathcal{R}_X$ maps $T_X\mathcal{N}$ to $\mathcal{N}$. For this, it is enough to show that $\mathcal{P}_\mathcal{S}(X) \in \mathcal{M}$ for every $X \in \mathcal{M}$. This can be seen from the representation (3.1)

$$X = (U \otimes I_n)\,[V]_{rn}^p,$$

which is equivalent to $X \in \mathcal{M}$. Here $U \in \mathbb{R}^{m\times r}$ can be chosen to have pairwise orthonormal columns, e.g. U from the SVD (3.7). Then letting $[V]_{rn}^p = Q_1\Sigma Q_2^\mathsf{T}$ be an

SVD of $[V]_{rn}^{p}$, we find that $(U \otimes I_n)Q_1 \Sigma Q_2^{\mathsf{T}}$ yields an SVD of X. Hence, the orthogonal polar factor of X,

$$\mathcal{P}_{\mathcal{S}}(X) = (U \otimes I_n)Q_1 Q_2^{\mathsf{T}},$$

is of the form (3.1), which means $\mathcal{P}_{\mathcal{S}}(X) \in \mathcal{M}$.

Now given that $\mathcal{R}_X^{\mathcal{M}}$ is smooth on the tangent bundle $T\mathcal{M}$ and $\mathcal{P}_{\mathcal{S}}$ is smooth in a neighborhood of $\mathcal{S}$ (as noted in the previous subsection), the required smoothness properties of $(X, \xi) \mapsto \mathcal{R}_X(\xi)$ follow from the chain rule. Note that $\xi \in T_X\mathcal{N} = T_X\mathcal{S} \cap T_X\mathcal{M}$. Then, since $\mathcal{R}_X^{\mathcal{M}}(\xi) = X + \xi + o(\|\xi\|)$ and using (3.6), we have

$$\mathcal{R}_X(\xi) = \mathcal{P}_{\mathcal{S}}(X + \xi + \eta) = X + \xi + \mathcal{P}_{T_X\mathcal{S}}(\eta) + o(\|\xi + \eta\|),$$

with $\eta = o(\|\xi\|)$ for $\xi \to 0$. Exploiting $\|\mathcal{P}_{T_X\mathcal{S}}(\eta)\| \leq \|\eta\|$, this implies

$$\limsup_{\xi \to 0} \frac{\|\mathcal{R}_X(\xi) - X - \xi\|}{\|\xi\|} = 0,$$

which shows the retraction property (3.4) for $\mathcal{R}_X$. □

In our implementation, we choose the following retraction for the manifold $\mathcal{M}$. Let

$$[\xi]_m^{np} = \mathbf{U} \begin{pmatrix} \Lambda_1 & \Lambda_2 \\ \Lambda_3 & 0 \end{pmatrix} \mathbf{V}^{\mathsf{T}} = U\Lambda_1 V^{\mathsf{T}} + U\Lambda_2 V_\perp^{\mathsf{T}} + U_\perp \Lambda_3 V^{\mathsf{T}}$$

be a tangent vector in $T_X\mathcal{N} \subseteq T_X\mathcal{M}$ given the SVD (3.7) of $[X]_m^{np}$. Then we take

$$\begin{aligned} \mathcal{R}_X^{\mathcal{M}}(\xi) &= \left[\big(U + U_\perp \Lambda_3 \Sigma^{-1}\big)\big(\Sigma V^{\mathsf{T}} + \Lambda_1 V^{\mathsf{T}} + \Lambda_2 V_\perp^{\mathsf{T}}\big) \right]_{mn}^{p} \\ &= X + \xi + \left[U_\perp \Lambda_3 \Sigma^{-1}\big(\Lambda_1 V^{\mathsf{T}} + \Lambda_2 V_\perp^{\mathsf{T}}\big) \right]_{mn}^{p}, \end{aligned}$$

as a retraction for $\mathcal{M}$ (note that $U_\perp \Lambda_3 \Sigma^{-1}\big(\Lambda_1 V^{\mathsf{T}} + \Lambda_2 V_\perp^{\mathsf{T}}\big) = O(\|\xi\|^2)$). The resulting retraction for the manifold $\mathcal{N}$ is

$$\mathcal{R}_X(\xi) = \mathcal{P}_{\mathcal{S}}\left(\left[\big(U + U_\perp \Lambda_3 \Sigma^{-1}\big)\big(\Sigma V^{\mathsf{T}} + \Lambda_1 V^{\mathsf{T}} + \Lambda_2 V_\perp^{\mathsf{T}}\big) \right]_{mn}^{p} \right).$$

Regarding vector transport, it is known that for embedded submanifolds, the projection onto the tangent space provides the simplest possible vector transport; see [1]. Given $\xi \in T_X\mathcal{N}$, such a vector transport takes another tangent vector $\eta \in T_X\mathcal{N}$ (in the same tangent space) to the tangent space at $\mathcal{R}_X(\xi)$ by simply projecting it. We also need to address the orthogonal invariance (2.2) of the function f. In a simple gradient descent method, the search direction is the Riemannian gradient which is always orthogonal to the orbit of such orthogonal transformations. However, for more sophisticated methods, one has to deal with general search directions in the tangent space $T_X\mathcal{N}$, and all directions parallel to the orbits have to be factored out. This is usually done by projecting onto a *horizontal space* for the Grassmannian $\mathrm{Gr}(mn, p) = \mathcal{S}/\mathcal{O}(p)$ with $\mathcal{O}(p)$ denoting the $p \times p$ dimensional real orthogonal matrices. A possible choice is

$$H_X \mathrm{Gr}(mn, p) = \{\xi \in \mathbb{R}^{mn \times p} : \xi^{\mathsf{T}} X = 0\} \subset T_X\mathcal{S}$$

with the orthogonal projection

$$\mathcal{P}_{H_X \mathrm{Gr}(mn,p)}(\xi) = (I_{mn} - XX^{\mathsf{T}})\xi \tag{3.10}$$

for $\xi \in T_X\mathcal{S}$.

Algorithm 1: A Riemannian nonlinear conjugate gradient method for the computation of a low rank eigenspace.

input : Symmetric matrix $\mathbf{H} \in \mathbb{R}^{mn\times mn}$, rank r, dimension of eigenspace M, accuracy $\varepsilon > 0$.

output : A matrix $X \in \mathbb{R}^{mn\times M}$ with rank-r column decomposition.

Set $i = 0$;

Set initial guess $X^{[0]}$;

Set initial search direction $\xi^{[0]} := -\mathrm{grad} f(X^{[0]}) = -\mathcal{P}_{T_{X^{[0]}}\mathcal{N}}(\mathbf{H}X^{[0]})$;

repeat

Find stepsize $\alpha_i = \sigma^k$ via Armijo backtracking, i.e., k is the smallest nonnegative integer that fulfills

$$f(X^{[i]}) - f\big(\mathcal{R}_{X^{[i]}}(\sigma^k\xi^{[i]})\big) \geq -c\,\langle \mathrm{grad} f(X^{[i]}), \sigma^k\xi\rangle$$

for parameters $\sigma \in (0,1), c > 0$;

Set

$$X^{[i+1]} := \mathcal{R}_{X^{[i]}}(\alpha_i\xi^{[i]});$$

Update search direction (Hestenes-Stiefel)

$$\beta_i := \frac{\langle \mathrm{grad} f(X^{[i+1]}), \mathrm{grad} f(X^{[i+1]}) - \mathcal{T}_{X^{[i]},\alpha_i\xi^{[i]}}(\mathrm{grad} f(X^{[i]}))\rangle}{\langle \mathcal{T}_{X^{[i]},\alpha_i\xi^{[i]}}(\xi^{[i]}), \mathrm{grad} f(X^{[i+1]}) - \mathcal{T}_{X^{[i]},\alpha_i\xi^{[i]}}(\mathrm{grad} f(X^{[i]}))\rangle};$$

$$\xi^{[i+1]} := -\mathrm{grad} f(X^{[i+1]}) + \beta_i\mathcal{T}_{X^{[i]},\alpha_i\xi^{[i]}}(\xi^{[i]});$$

$i \leftarrow i + 1$;

until $\|\mathrm{grad} f(X^{[i]})\| < \varepsilon$;

Proposition 3.4. *If $\xi \in T_X\mathcal{N}$, then also $\mathcal{P}_{H_X\mathrm{Gr}(mn,p)}(\xi) \in T_X\mathcal{N}$.*

Proof. In light of Proposition 3.1, since $H_X\mathrm{Gr}(m,n,p) \subset T_X\mathcal{S}$, we only need to show that $P_{H_X\mathrm{Gr}(mn,p)}(\xi) \in T_X\mathcal{M}$. Since $\xi \in T_X\mathcal{M}$, this follows from (3.10) if we verify that $XX^\mathsf{T}\xi \in T_X\mathcal{M}$. This, however, is obvious: By (3.1), $XX^\mathsf{T}\xi = (U \otimes I_n)[W]^p_{rn}$ for some $W \in \mathbb{R}^{np\times r}$, where U contains a basis for the column space of $[X]^{np}_m$. Hence, $[XX^\mathsf{T}\xi]^{np}_m = UW^T$, which implies that this matrix is in $T_X\mathcal{M}$ as given in (3.8). □

The vector transport therefore consists of projecting first onto the tangent space at the new iterate and then projecting onto the horizontal space. We denote this operation as

$$\mathcal{T}_{X,\xi}(\eta) = \mathcal{P}_{H_X\mathrm{Gr}(mn,p)}\big(\mathcal{P}_{T_{\mathcal{R}_X(\xi)}\mathcal{N}}(\eta)\big).$$

Other options for vector transport are possible but more involved.

Having all the ingredients necessary for Riemannian optimization, we can now perform a gradient descent or even a nonlinear CG method on the manifold $\mathcal{N}$; see Algorithm 1. Analysis of this algorithm can be performed within the generic framework of Riemannian optimization [1]. Basically, if the algorithm produces an accumulation point on the manifold $\mathcal{N}$, it will be a stationary point of f on $\mathcal{N}$. However, just like in [37], the manifold $\mathcal{N}$ is not closed, and so the existence of such accumulation points cannot be

guaranteed. The convergence analysis fails when approaching the singularities, where the rank of $[X]_m^{np}$ drops. This issue is intrinsic to low rank optimization and shall not be further discussed here.

4 A modified DMRG

Our main application is the DMRG algorithm, which in its traditional form uses the truncated eigenspace method in order to solve its subproblem. This turns out to work reasonably well in spite of the flawed nature of its solver, as we will see in our experiments. Nevertheless, we will propose an alternative DMRG algorithm that makes use of our Riemannian method. We will also introduce a heuristic rank adaptation procedure based on the residual of the function rather than a less related error in the Frobenius norm.

First, we give a short introduction into the TT format and the DMRG algorithm. For a more thorough overview, we refer the reader to the review articles [34, 36]. The DMRG algorithm is traditionally applied to quantum many-body systems. Given a self-adjoint tensor operator $\mathcal{H} : \mathbb{C}^{n_1 \times \ldots \times n_d} \to \mathbb{C}^{n_1 \times \ldots \times n_d}$, the aim is to find the lowest energy state $\Psi \in \mathbb{C}^{n_1 \times \ldots \times n_d}$, i.e., the tensor that minimizes the Frobenius inner product $\langle \Psi, \mathcal{H}(\Psi) \rangle$. In applications, the Hamilton operator can describe systems of distinguishable particles (for example, *spin chains*) or indistinguishable particles (e.g. in the context of quantum chemistry or cold atom systems). In both cases, typically, the physical dimensions $n_1, \ldots, n_d$ are equal and small, e.g. $n_1 = \ldots = n_d = n$ with $n \in \{2, 4\}$. But even in these cases, the computational complexity of the above problem grows exponentially with the tensor order d. This effect is often called the *curse of dimensionality*, and many techniques of overcoming it have been proposed in the recent past.

4.1 Tensor trains

A common approach to overcome the curse of dimensionality is to restrict the *tensor rank* of the state Ψ. This is particularly fruitful for systems of low spatial dimension. The restriction of the ranks is nontrivial since there are many possibilities to define a tensor rank; see e.g. [12]. We will not go into detail here and only consider the approach that is most widely used in physics, the matrix product states or TT format [3, 34, 29]. The tensor $\Psi \in \mathbb{C}^{n_1 \times \cdots \times n_d}$ is decomposed into d components. For each multi-index $(i_1, \ldots, i_d)$, the entry $\Psi(i_1, \ldots, i_d)$ is given by a product of matrices $\Psi^{(j)}(i_j) \in \mathbb{C}^{r_{j-1} \times r_j}$, i.e.

$$\Psi(i_1, \ldots, i_d) = \Psi^{(1)}(i_1) \cdots \Psi^{(d)}(i_d),$$

hence the name. Usually, we set $r_0 = r_d = 1$, and the first and last matrices are vectors. By varying $i_j = 1, \ldots, n_j$, the matrices $\Psi_j(i_j)$ become tensors $\Psi^{(j)} \in \mathbb{C}^{r_{j-1} \times n_j \times r_j}$ of order 3 (respectively, of order 2 in the case the first and last one), which we call the *components* of Ψ. The minimal numbers $r_1, \ldots, r_{d-1}$ needed to represent Ψ are called TT ranks of Ψ.

In order to facilitate the discussion, we introduce some additional notation: Two neighboring components $\Psi^{(j)}$ and $\Psi^{(j+1)}$ can be *contracted* by performing elementwise the matrix product over the common index

$$\big(\Psi^{(j)} \bullet \Psi^{(j+1)}\big)(i_j, i_{j+1}) := \Psi^{(j)}(i_j)\, \Psi^{(j+1)}(i_{j+1})$$

and yielding a tensor of order 4 denoted by

$$\Psi^{(j,j+1)} := \Psi^{(j)} \bullet \Psi^{(j+1)} \in \mathbb{C}^{r_{j-1} \times n_j \times n_{j+1} \times r_{j+1}}.$$

Furthermore, if we contract all tensors up to but not including the j-th component, we write

$$\Psi^{<j} := \Psi^{(1)} \bullet \ldots \bullet \Psi^{(j-1)} \in \mathbb{C}^{n_1 \times \cdots \times n_{j-1} \times r_{j-1}}$$

and analogously $\Psi^{>j}$ for the contraction of all tensors to the right of the j-th component. This way, the TT decomposition of Ψ can be written also as

$$\Psi = \Psi^{(1)} \bullet \ldots \bullet \Psi^{(d)} = \Psi^{<j} \bullet \Psi^{(j)} \bullet \Psi^{>j}$$

for any $j = 1, \ldots, d$.

The main purpose of the TT format is to break the curse of dimensionality: For $r := \max(r_1, \ldots, r_{d-1})$ and $n := \max(n_1, \ldots, n_d)$, we only need to store $\mathcal{O}(ndr^2)$ entries instead of $\mathcal{O}(n^d)$ in the full format. Restricting the ground state computation for $\mathcal{H}$ to a manifold of low rank TT tensors [17] therefore only makes sense if we can expect a reasonable approximation in this space. This is, for instance, fulfilled if $\mathcal{H}$ describes a one-dimensional model with finite interaction length and gapped spectrum. Here it can be shown that the lowest eigenstate fulfills an area law for the entanglement entropies [15], which implies that it can be efficiently approximated by a TT tensor with increasing system size d [38]. Beyond these rigorously established beacons, however, practical expertise shows that a TT approximation can be successfully applied in much broader settings, including systems from quantum chemistry or two-dimensional spin systems; see e.g. [34, **?**, 28, 36] for overviews.

4.2 Block TT, DMRG, and modified version

In order to simultaneously approximate p eigenstates, we use the block TT format introduced in [6]. The tensor $\Psi \in \mathbb{C}^{n_1 \times \cdots \times n_d \times p}$ combining the p tensors $\Psi_\ell \in \mathbb{C}^{n_1 \times \cdots \times n_d}$, $\ell = 1, \ldots, p$, is then decomposed into a block TT format, with one chosen index $k \in \{1, \ldots, d\}$ carrying the index ℓ. This means the entries of Ψ are again given by a chain of tensors $\Psi^{(j)} \in \mathbb{C}^{r_{j-1} \times n_j \times r_j}$ for $j \neq k$ and $\Psi_\ell^{(k)} \in \mathbb{C}^{r_{k-1} \times n_k \times r_k}$ such that

$$\Psi_\ell = \Psi^{<k} \bullet \Psi_\ell^{(k)} \bullet \Psi^{>k}.$$

In essence, this generalizes row and column decomposition of a matrix in (2.3) and (2.4) in the sense that each eigenstate Ψ_ℓ shares with the other states the common subspaces spanned by $\Psi^{<k}$ and $\Psi^{>k}$.

In the DMRG algorithm at each subiteration, we fix all but *two* neighboring components $\Psi^{(j)}$ and $\Psi^{(j+1)}$ and only optimize these two components together. Alternatively, in the block ALS algorithm (also called single-site DMRG), all but one component $\Psi^{(j)}$ are fixed at a time. Both approaches decompose for each subiteration Ψ as block TT, where the component k that carries the dependency on ℓ is chosen as follows: In case of block ALS, one sets $k = j$, whereas for the DMRG, one sets $k = j + 1$ or $k = j$ depending on whether we are currently sweeping through the tensors from left to right or from right to left. This determines whether we require a column decomposition (2.3) or a row decomposition (2.4) for the combined component $\Psi^{(j,j+1)}$.

We then aim to solve

$$\text{minimize } \sum_{\ell=1}^{p} \langle \Psi_\ell, \mathcal{H}(\Psi_\ell) \rangle \quad \text{subject to} \quad \langle \Psi_\ell, \Psi_{\ell'} \rangle = \delta_{\ell,\ell'} \tag{4.1}$$

by varying one or two components in subsequent subiterations. In the following, we describe the structure of the DMRG method. The block ALS method can be readily derived by making the obvious adjustments for optimizing one component in each subiteration only. Furthermore, we describe the case of a sweep from left to right, i.e., $k = j + 1$, where one has to choose the column decomposition (2.3) for the subproblem.

We assume that the block TT decomposition is chosen such that the TT is left-normalized for components to the left of $\Psi^{(j)}$ and right-normalized for components to the right of $\Psi^{(j+1)}$, i.e.

$$\left[\Psi^{<j}\right]_{r_{j-1}}^{n_1\cdots n_{j-1}} \left[\Psi^{<j}\right]_{n_1\cdots n_{j-1}}^{r_{j-1}} = I_{r_{j-1}} \quad \text{and} \quad \left[\Psi^{>j+1}\right]_{r_{j+1}}^{n_{j+2}\cdots n_d} \left[\Psi^{>j+1}\right]_{n_{j+2}\cdots n_d}^{r_{j+1}} = I_{r_{j+1}}.$$

This means that for each $j = 1, \ldots, d-1$, the subiteration step consists of solving

$$\begin{aligned} &\min_{\operatorname{rank}(\Phi^{(j,j+1)})=r_j} \sum_{\ell=1}^{p} \left\langle \Psi^{<j} \bullet \Phi_\ell^{(j,j+1)} \bullet \Psi^{>j+1}, \mathcal{H}\big(\Psi^{<j} \bullet \Phi_\ell^{(j,j+1)} \bullet \Psi^{>j+1}\big) \right\rangle \\ &\text{subject to} \quad \langle \Phi_\ell^{(j,j+1)}, \Phi_{\ell'}^{(j,j+1)} \rangle = \delta_{\ell,\ell'}. \end{aligned} \tag{4.2}$$

Indeed, thanks to the orthogonalization of the TT, the constraints $\langle \Phi_\ell^{(j,j+1)}, \Phi_{\ell'}^{(j,j+1)} \rangle = \delta_{\ell,\ell'}$ in this subproblem ensure the global constraints $\langle \Psi_\ell, \Psi_{\ell'} \rangle = \delta_{\ell,\ell'}$ from (4.1). Let us note that for simplicity, we take the liberty to slightly abuse notation, here and in the following, by writing $\Phi^{(j,j+1)}$ and meaning its appropriate matricization. This allows us to readily transfer the tensor problem to our discussed matrix case.

We also define the local operator

$$\mathbf{H}_{j,j+1} : \mathbb{C}^{r_{j-1}\times n_j \times n_{j+1} \times r_{j+1}} \to \mathbb{C}^{r_{j-1}\times n_j \times n_{j+1} \times r_{j+1}},$$

which acts on $\Phi^{(j,j+1)}, \tilde{\Phi}^{(j,j,+1)} \in \mathbb{C}^{r_{j-1}\times n_j \times n_{j+1} \times r_{j+1}}$ as

$$\left\langle \Phi^{(j,j+1)}, \mathbf{H}_{j,j+1}(\tilde{\Phi}^{(j,j+1)}) \right\rangle = \left\langle \Psi^{<j} \bullet \Phi^{(j,j+1)} \bullet \Psi^{>j+1}, \mathcal{H}\big(\Psi^{<j} \bullet \tilde{\Phi}^{(j,j+1)} \bullet \Psi^{>j+1}\big) \right\rangle$$

is clearly Hermitian and linear if we take the whole domain. In the traditional DMRG algorithm, the solution of (4.2) is approximated along the lines of the truncated eigenspace method by first calculating the full solution

$$\begin{aligned} \Psi^{(j,j+1)} = &\operatorname*{argmin}_{\Phi^{(j,j+1)} \in \mathbb{C}^{r_{j-1}\times n_j \times n_{j+1} \times r_{j+1} \times p}} \sum_{\ell=1}^{p} \left\langle \Phi_\ell^{(j,j+1)}, \mathbf{H}_{j,j+1}(\Phi_\ell^{(j,j+1)}) \right\rangle \\ &\text{subject to} \quad \langle \Phi_\ell^{(j,j+1)}, \Phi_{\ell'}^{(j,j+1)} \rangle = \delta_{\ell,\ell'}, \end{aligned} \tag{4.3}$$

that is, the p lowest eigenvectors of $\mathbf{H}_{j,j+1}$. Then the ranks are truncated as described in section 2.1 in order to obtain the low rank approximation

$$\Psi^{(j)} \bullet \Psi_\ell^{(j+1)} \approx \Psi_\ell^{(j,j+1)}.$$

Alternatively, this last step allows for a rank adaptivity; i.e., a new rank can be chosen according to some error threshold when projecting to the low rank decomposition. However, as we have discussed above in detail, this rank truncation and rank adaptivity is done with respect to the Frobenius norm, whereas the goal is to minimize the energy of the Hamiltonian $\mathbf{H}_{j,j+1}$.

We can hence formulate an alternative DMRG method that uses our Riemannian optimization method presented in section 3 for solving the DMRG subproblem (4.3) with a rank constraint directly, which can be written as

$$\begin{aligned} \Psi^{(j,j+1)} = \underset{\operatorname{rank}(\Phi^{(j,j+1)})=r_j}{\operatorname{argmin}} & \sum_{\ell=1}^{p} \left\langle \Phi_\ell^{(j,j+1)}, \mathbf{H}_{j,j+1}(\Phi_\ell^{(j,j+1)}) \right\rangle \\ \text{subject to} \quad & \langle \Phi_\ell^{(j,j+1)}, \Phi_{\ell'}^{(j,j+1)} \rangle = \delta_{\ell,\ell'}, \end{aligned} \tag{4.4}$$

exploiting the normalized form of the TT. After a suitable matricization, (4.4) takes the form of a complex version of (2.1) subject to a rank-r_j column decomposition (2.3). This yields a low rank solution of the reduced eigenvalue problem, which is directly obtained in two separate factors, thus allowing us to move to the neighboring core without further rank truncation.

4.3 Rank adaptivity

As mentioned above, the classical DMRG method has a straightforward rank adaptive extension. Here, the algorithm is provided with a threshold for the error of the low rank projection of (4.3) performed in each subiteration, and the rank is chosen accordingly.

Our modified DMRG method, however, works with a fixed rank r_j that needs to be chosen in each subproblem (4.4). We therefore propose a rank adaptive strategy based on the local residual $\operatorname{Res}(\Psi^{(j,j+1)}, \mathbf{H}_{j,j+1})$ of the optimization problem, as defined in (2.6). We define a threshold for the Frobenius norm of the local residual and decide to increase the allowed rank by one the next time the corresponding pair of components is optimized if the residual violates the threshold. Technically, for implementing the rank increase, initial guesses for the subproblems with higher ranks have to be provided. This can be done entirely randomly or by adding new random vectors to the subspaces provided by the current solution. In our numerical experiments below, we instead use the correspondingly truncated eigenspace solution (i.e. the DMRG update of the subproblem) as an initialization.

Note that many other policies for increasing the rank could be designed, e.g. by increasing the ranks by more than one and/or based on the size of the singular values of the residual. Furthermore, as a more adaptable strategy, we can alter the rank r_j and rerun the current subiteration until a rank is found such that the residual does not violate the threshold. However, we have found that this strategy can overestimate the necessary ranks.

5 Numerical experiments

In the following, we present two classes of numerical experiments in order to investigate the performance of the discussed methods. First, we directly consider the matrix optimization problem (2.1) for different operators $\mathbf{H}$ and compare the results of the truncated eigenspace method, semi-Riemannian AO, and the proposed Riemannian scheme. Second, we compare the performance of the different DMRG-like methods of section 4 for a spin chain model. In all experiments, we choose $\sigma = \frac{1}{2}, c = 10^{-4}$ for the parameters in the Armijo condition with backtracking to find the stepsizes α_i in Algorithm 1.

5.1 Computing low rank eigenspaces

We compare the truncated eigenspace method with the two algorithms introduced in section 3. For this, we generate four different operators $\mathbf{H}$, a symmetric indefinite operator, a two-dimensional Laplace operator on a grid, and a Laplace operator that is perturbed by a small potential without and with noise. Our aim is to approximate the minimal eigenspace using the low rank column format (2.3). All numerical experiments are done in `Matlab` R2018a.

As explained in section 3.1, the update for V is a standard eigenvalue problem for the operator $(U^\mathsf{T} \otimes I_n)\mathbf{H}(U \otimes I_n)$ and is solved using the `Matlab` function `eigs`. The update for U is an optimization problem on the Stiefel manifold $\mathrm{St}(m, r)$ and is solved using the `manopt` toolbox [5] with random starting point. The Riemannian method in Algorithm 1 also uses random initializations. As our implementations have not been optimized for performance, we refrain from comparing the CPU time of the computations. The Riemannian method in particular could be presumably improved by implementing it within the `manopt` environment as well.

5.1.1 Random symmetric operator

We set $m = n = 16$ and generate a random symmetric operator $\mathbf{H} \in \mathbb{R}^{(16\times16)\times(16\times16)} \cong \mathbb{R}^{256\times256}$ with normally distributed entries. In Figure 1, we report on the value of the function f for the lowest $p = 5$ eigenvectors on the left. On the right, we display the distances of the computed spaces to the full rank state. For each rank $r = 1, \ldots, 16$, we perform the semi-Riemannian AO and our Riemannian method 10 times and average over the resulting energies. This is done because each of the methods can produce different results for different starting points due to local minima. The truncated eigenspace method is strictly deterministic (up to the implementation of the SVD), and therefore an averaging is unnecessary. Note that in order to measure the energy of the space X resulting from the truncated eigenspace method, we need to orthogonalize first since the columns of X after truncation are in general not orthonormal.

We recall that the distance of two linear spaces V and W can be measured via the distance of the orthogonal projections $\mathcal{P}_V$ and $\mathcal{P}_W$. In our case, let $X_0 \in \mathrm{St}(mn, p)$ be the matrix that has the orthonormal basis of the full rank eigenspace as its columns (this is computed using the `eigs` routine of `Matlab` for the matricization $[\mathbf{H}]^{mn}_{mn} \in \mathbb{R}^{mn\times mn}$; i.e., it will actually consist of the lowest eigenvectors). Then we measure the distance of the linear spaces $\mathrm{span}(X_0)$ and $\mathrm{span}(X)$ using the Frobenius error for the orthogonal projectors

$$\mathrm{dist}(\mathrm{span}(X_0), \mathrm{span}(X)) = \|X_0 X_0^\mathsf{T} - X X^\mathsf{T}\|.$$

Note that the Frobenius error is at most $\sqrt{p}$ times larger than the classical distance measured in the spectral norm.

We can see that the truncated eigenspace method produces the worst results in terms of energy. However, as expected, the computed eigenspace is closest to the actual one in the Frobenius norm. The other two methods perform equally well in terms of the Frobenius distance to the full rank eigenspace, and they all find the correct space once we allow full rank. In terms of energy, they produce the same results on average.

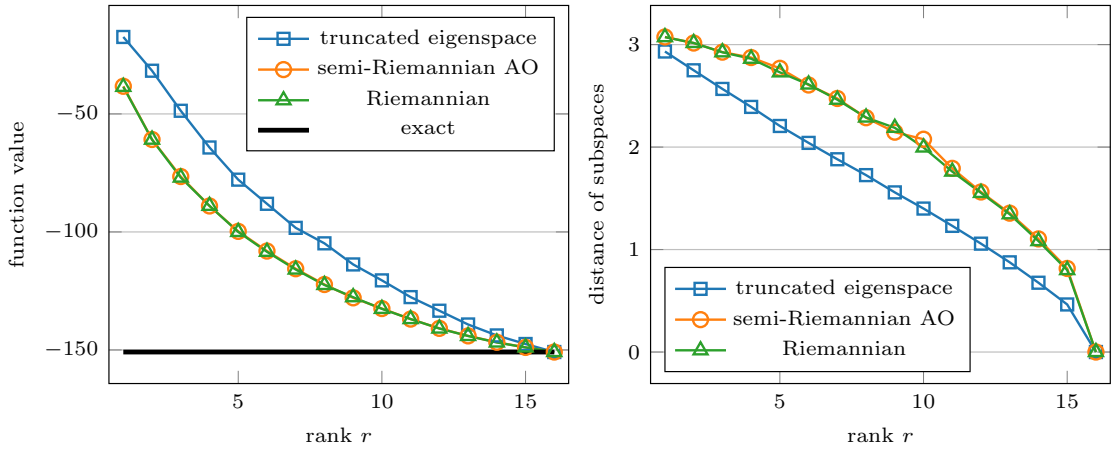

Figure 1: Left: Function values for the different methods for $p = 5$ and $r = 1, \ldots, 16$ for a random symmetric operator. The black line corresponds to the exact energy. Right: Distance in the Frobenius norm of the computed low rank eigenspaces to the actual eigenspace.

5.1.2 Discretized Laplace operator in two dimensions

Next, we tested the setting from Example 2.1 for the two-dimensional discrete negative Laplace operator

$$\mathbf{H} = L \otimes I_n + I_m \otimes L \tag{5.1}$$

on an equidistant grid of $m = n = 128$ with zero boundary condition. Here L is the one-dimensional finite difference matrix $h^{-2} \cdot \mathrm{tridiag}(-1, 2, -1)$ with $h = 2/129$. We compute the eigenspace to the lowest $p = 22$ eigenvalues. As explained above, the eigenvectors share column spaces, and one can check that for the lowest 22 eigenvectors, the common column space is spanned by the five lowest eigenvectors of L only. Hence, the minimal eigenspace can be represented by a matrix $X \in \mathbb{R}^{16384 \times 22}$ with $\mathrm{rank}([X]_{128}^{2816}) = 5$. We investigate the behavior of the different algorithms for the rank values $r = 1, \ldots, 6$. It should be noted that full convergence is usually not necessary, as the function value does not change significantly for a small gradient. Thus, we consider both iterative algorithms to be converged once the norm of the projected gradient is small enough, $\|\mathrm{grad} f(X)\| < 10^{-2}$.

The results are depicted on the left of Figure 2. For every rank $r = 1, \ldots, 6$, we report the relative energy error at termination, i.e., $(f(X) - f(X_0))/|f(X_0)|$ with X_0 again consisting of the exact lowest eigenvectors. The Riemannian method consistently finds the lowest energy. The truncated eigenspace method performs worse, but both algorithms find the exact solution once the rank is high enough. In contrast, semi-Riemannian AO does not find the exact solution. In fact, after one sweep, it finds a critical point, i.e., a point at which the projected gradient is zero, which turned out to be a saddle point, that is, a different *exact* eigenspace that has a higher energy. We observed that this happens frequently when multiples of eigenvalues are present, and we suspect that this is due to the alternating nature of the algorithm.

As a third experiment, we investigate again the case of a two-dimensional discrete Laplace operator but now perturbed by a Newton potential. The Hamiltonian $\mathbf{H}$ is a

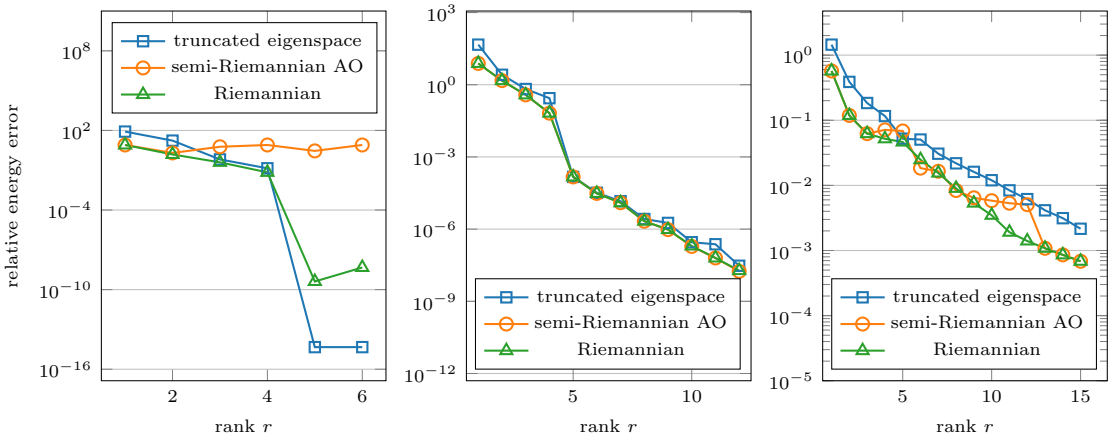

Figure 2: Left: Relative energy error for the different methods for $p = 22$ and $r = 1, \ldots, 6$ for a two-dimensional Laplace operator in (5.1). Center: Relative energy error for the different methods for $p = 22$ and $r = 1, \ldots, 12$ for a two-dimensional Laplace operator plus a potential. Right: Relative energy error for the different methods for $p = 22$ and $r = 1, \ldots, 15$ for a two-dimensional Laplace operator plus a potential plus a small perturbation with a Kronecker product operator.

discretization of the operator

$$\mathcal{H} = -\Delta_x + \frac{1}{\|x\|}.$$

We again approximate this operator with 128 equidistant grid points on the interval $[-1, 1]$ in each dimension. For the approximation of the potential, we use an exponential sum with 8 terms [11, 12] for an accuracy of $1.82 \cdot 10^{-2}$ measured in the L^∞-norm. The whole discretized operator then can then be written as a sum of 10 Kronecker product operators. Again, we compute the eigenspace to the lowest 22 eigenvalues exactly. Due to the perturbation, the exact solution $[X_0]_m^{np}$ does not have rank 5, but only approximately; see the singular values of the solution to the two problems in Figure 3 on the right.

The results of this experiment can be seen in the center of Figure 2. Here, the Riemannian method produces approximately the same energy as the semi-Riemannian AO, which is lower than the one obtained with the truncated eigenspace method for $r < 5$. As seen in Figure 3, even in the presence of the potential, the singular values of the exact solution drop, allowing for an effective low rank approximation for $r \geq 5$. Accordingly, we find that for $r \geq 5$, all three methods produce similar results. We also report on the development of the residuals of the three methods for the case with an added potential; see Figure 3 on the right. It can be seen that the residuals shrink with better accuracy of the solution, and they are therefore a reasonable error estimator. As expected, the residuals of the iterative methods are lower than for the truncated eigenvalue solver.

The most significant difference between the algorithms occurs when we add a noise term to the operator. Here, we take the operator of the previous experiment and add a symmetric Kronecker product operator $\epsilon A_1^\mathsf{T} A_1 \otimes A_2^\mathsf{T} A_2$, where ϵ is chosen to be 10^{-1} and the entries of A_1 and A_2 are normally distributed. The singular values of the full rank result of this operator decay much slower than before; see Figure 3. Heuristically, this signifies that it is not well approximable by a low rank matrix. However, as can be seen on the right of Figure 2, the Riemannian method somewhat alleviates this problem

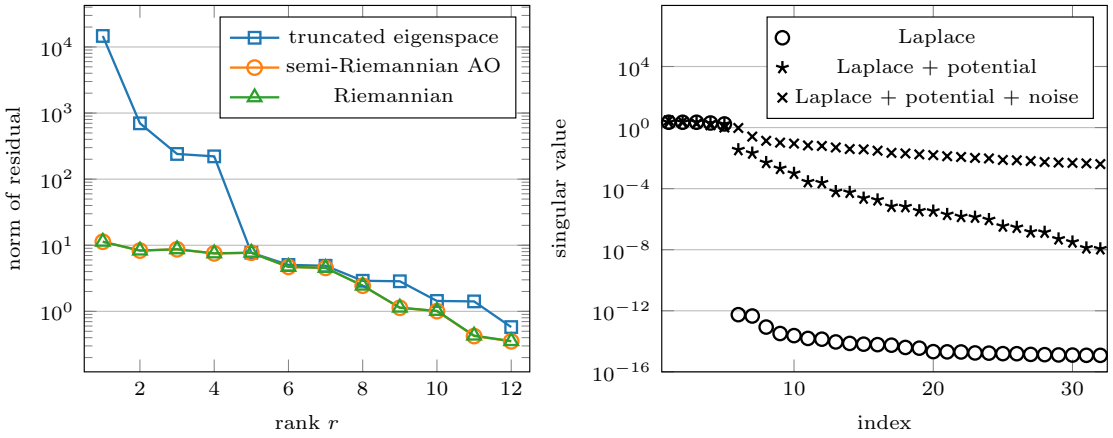

Figure 3: Left: Norm of residual for the different methods for $p = 22$ and $r = 1, \ldots, 12$ for a two-dimensional Laplace operator plus a potential. Right: Singular values of the exact solution $[X_0]_m^{np}$ to the Laplace operator and the Laplace operator plus potential without and with a small perturbation with a Kronecker product operator. Again, $p = 22$.

and clearly outperforms the truncated eigenspace method. This supports the results from our experiments with random operators: If the singular values of the exact solution indicate bad approximability, it can be helped by minimizing the energy rather than the Frobenius norm. The semi-Riemannian AO proved to be again unstable in this setting.

5.2 Computing excited states

We compare the three methods discussed in section 4 for the approximation of the eigenstates of a quantum system: block ALS, DMRG, and our alternative Riemannian strategy. As we found in section 5.1 that the Riemannian method of section 3.2 yields more stable results for the subproblems than the semi-Riemannian AO method of section 3.1, we refrain from using the latter here. We discuss the chosen system below and compare the three methods based on their accuracy for fixed maximal ranks as well as their efficiency in the context of adaptive ranks. Unlike in the matrix experiments in section 5.1, where the Riemannian solver was initialized randomly, we now take the truncated eigenspace solution, i.e. the DMRG subsolution, for its initialization, which we observed to work better in this problem. Hence, the Riemannian solver acts here as a local improvement of the truncated eigenspace/DMRG approach for further decreasing the energy. It is then also easy to implement the rank-increasing strategy from section 4.3 by simply using a truncation to higher rank.

We use a `python` implementation for the DMRG as well as the Riemannian optimization method of section 3.2 utilizing `numpy` 1.16.4 and `scipy` 1.2.2.

5.2.1 Considered operator and reference energy

As a test case, we present an exactly solvable, real spin chain model on d sites. For this, define the Pauli operators

$$\mathcal{S}^{(x)} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \mathcal{S}^{(z)} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

as well as the notation $\mathcal{S}_k^{(x)} = \bigotimes_{j=1}^{k-1} I_2 \otimes \mathcal{S}^{(x)} \otimes \bigotimes_{j=k+1}^{d} I_2$ and $\mathcal{S}_k^{(z)}$ accordingly. We then construct the so-called transverse field Ising Hamiltonian

$$\mathcal{H} = -\sum_{k=1}^{d-1} \mathcal{S}_k^{(z)} \mathcal{S}_{k+1}^{(z)} - t \sum_{k=1}^{d} \mathcal{S}_k^{(x)}, \tag{5.2}$$

with $t \in \mathbb{R}$. This Hamiltonian has four notable properties. First, its eigenvalues can be efficiently computed by diagonalizing a $2d \times 2d$ matrix and some combinatorial arguments because it maps to a free fermionic model by virtue of the Jordan–Wigner transformation. Second, the model displays a quantum phase transition at $t = 1$, rendering the low-lying eigenstates harder to approximate with growing system size d due to logarithmic corrections to the area laws of the entanglement entropies; see e.g. [25]. Third, note that $\mathcal{H}$ has a TT operator decomposition with rank 3 independent of d (which can be, for instance, derived from [34, section 6]), allowing for an efficient implementation of all methods. Finally, it is a real symmetric operator such that all eigenvectors are real, allowing us to discuss this example within the scope of this work.

5.2.2 Results for fixed ranks

We set $d = 256$ and perform 4 sweeps for each method in order to approximate the $p = 4$ lowest eigenstates of $\mathcal{H}$ for $t = 1$ and various maximal ranks r of the TT. In the Riemannian solver, we set the tolerance for the gradient to 10^{-4} where in the case of block ALS and DMRG, the lowest eigenstates are computed with `scipy`, which wraps into a Lanczos method of `ARPACK`.

In Figure 4, we report on the left the minimal relative energy error encountered during each run. For the DMRG, we show both the minimal value encountered before and after truncation of all subiterations, where it is important to note that the former corresponds to a TT approximation where one r_j usually exceeds r. On the right, we show the norm of the global residual of the optimization problem (4.1) as well as the local residual of the optimization problem (4.2) that occurs when the energy is minimal. Concretely, we show the Frobenius norm of $\mathrm{Res}(\Psi, \mathcal{H})$ and $\mathrm{Res}(\Psi^{(j,j+1)}, \mathbf{H}_{j,j+1})$, where the former can be efficiently computed by exploiting the low rank structure of $\mathcal{H}$.

We find that the Riemannian method yields better approximations for the eigenvalues of $\mathcal{H}$ than the block ALS and DMRG algorithms over the range of tested ranks. In addition, it usually yields the lowest residuals, as expected. It is noteworthy that the Riemannian scheme also outperforms the eigenvalues obtained from DMRG before truncation, i.e., a TT approximation that violates the rank constraint in one component. However, in many smaller cases that we considered, the energy produced by the Riemannian method was lower than the one computed by the DMRG, but not before the truncation is performed. In general, we found only very small gains for the Riemannian scheme over block ALS or DMRG, and it is notable how well these algorithms perform in practice despite their flawed nature.

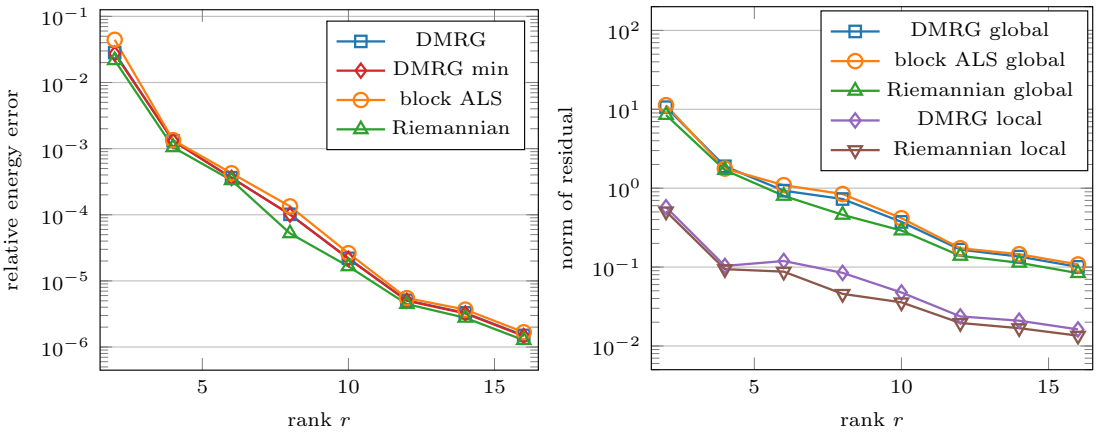

Figure 4: Left: Minimal relative energy error for the $p = 4$ lowest eigenvalues of (5.2) for $d = 256$ with increasing rank r, obtained with block ALS, DMRG, and the Riemannian scheme. For the DMRG, we additionally show the minimal value before the truncation obtained from (4.3), indicated as 'DMRG min'. Right: Norm of the global residual for block ALS, DMRG, and the Riemannian method as well as the local residual for the latter two.

5.2.3 Results for adaptive ranks

Finally, we present experiments on the adaptive rank strategy discussed in section 4.3. The Hamiltonian is chosen as above, and we approximate the $p = 16$ lowest eigenstates. Note here that the tolerances ϵ_{DMRG} chosen for the truncation error in the DMRG algorithm and $\epsilon_{\text{Riemann}}$ for the norm of the residual of the subiteration problem (4.2) are not directly comparable. We therefore choose two different ranges for the tolerances and report on the corresponding result in joint plots. The results are in Figure 5, where in all three subplots, the tolerance $\epsilon_{\text{Riemann}}$ is displayed on top and ϵ_{DMRG} is displayed at the bottom. Since in each sweep the ranks of the Riemannian method with rank adaption can grow at most by 1, we perform 20 sweeps for each configuration. The Riemannian iteration in the subproblem is performed until the norm of the Riemannian gradient is smaller than $10^{-2}\epsilon_{\text{Riemann}}$. This ensures sufficient accuracy of the solver.

In the left plot of Figure 5, we show the minimal relative energy error encountered in the corresponding configuration during the 20 sweeps; for the DMRG before and after truncation, respectively; and for the Riemannian method. For the DMRG, we again show the error for the energy (4.1) before and after truncation. The middle plot displays the global and local residual of the DMRG algorithm and the Riemannian method for the lowest energy.

In the right plot of Figure 5, we indicate as a measure of complexity the number of parameters of the TT solution with minimal energy, that is, $\sum_{j=1}^{d} n_j r_j r_{j+1}$. However, the ranks in the DMRG method fluctuate strongly in the course of one sweep, while the ranks of the Riemannian scheme stay essentially constant once convergence is achieved. As the algorithms have to pass through points of potentially large ranks in order to achieve the energy error displayed in the left of the figure, we also plot the minimal and maximal value of the number of parameters used during the sweep where the minimal energy was found (indicated as the shaded region in Figure 5 for both methods).

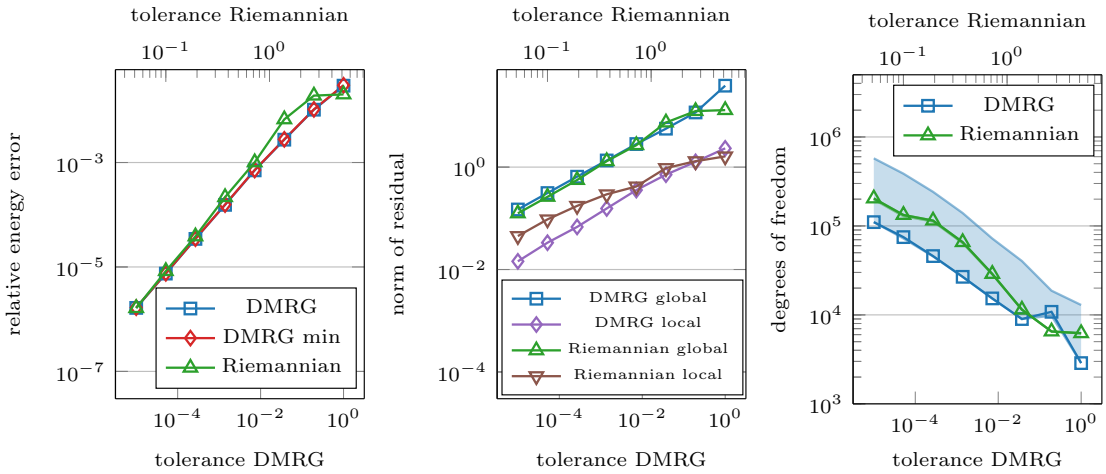

Figure 5: Left: Minimal relative energy error of the rank adaptive schemes for the $p = 16$ lowest eigenvalues of (5.2), $d = 128$, with respect to the tolerance for the TT approximation based on DMRG (tolerance indicated at the bottom) and Riemannian method (tolerance indicated at the top). For the DMRG, the minimal energy value obtained before the truncation is indicated as 'DMRG min'. Middle: Norms of the global and local residuals computed for the subiteration with minimal energy. Right: Summed number of degrees of freedom of the TT approximation with the lowest energy for the given tolerance (indicated by the corresponding markers). Additionally, minimal and maximal values for the used degrees of freedom during the sweep in which the minimal energy was found are indicated as the shaded region.

We find that for the chosen ranges of tolerances, the relative energy error is comparable for both methods and decreases monotonically with decreasing tolerance, respectively. Similarly, the norms of the residuals, global as well as local, are comparable and decrease with the tolerance. As expected, the number of degrees of freedom increases with decreasing tolerance. At the configuration with minimal energy, the number of degrees of freedom is usually significantly smaller for the DMRG as compared to the Riemann method with similar energy error, but its maximal value encountered during the best energy sweep lies always above the maximal value encountered with the Riemann method, meaning that the computational complexity of the DMRG algorithm can be significantly larger at stages where it has not converged yet.

In conclusion, both methods allow us to preset the accuracy of the computation using an appropriate tolerance and to adapt the ranks correspondingly. Comparing maximal ranks, we find that the Riemannian scheme here shows a visible advantage over the DMRG method. However, this is compensated when typical ranks are considered.

6 Conclusion

We have proposed a Riemannian approach for approximating eigenspaces of symmetric operators on matrix and tensor spaces subject to low rank constraints. One is a semi-Riemannian alternating optimization approach, and the other is a nonlinear Riemannian CG method on a low rank manifold. Compared to the SVD truncation of the eigenspace

or a block ALS method, which also employs SVD truncation, our methods minimize the trace energy (2.1) directly subject to the rank constraint. They produce in all iterations low rank approximations that respect the orthogonality constraint and yield a monotonically decreasing cost function. They can be used for improving the SVD-based solutions. This is in particular useful in the DMRG method where our methods can provide lower energies for the DMRG subproblems.

In the current implementation, the Riemannian methods are slower compared to the truncated eigenspace or the block ALS method. We expect that a dedicated Riemannian Jacobi–Davidson method, as in [32], can compete with the default methods also in terms of computational cost. Another goal could be to incorporate ideas of more complex but faster iterative eigenvalue solvers, like, for instance, preconditioning and subspace acceleration as in LOBPCG [22], into the manifold-based methods.

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 and J. Malick. Projection-like retractions on matrix manifolds. *SIAM J. Optim.*, 22(1):135–158, 2012.

[3] I. Affleck, T. Kennedy, E. H. Lieb, and H. Tasaki. Rigorous results on valence-bond ground states in antiferromagnets. *Phys. Rev. Lett.*, 59(7):799, 1987.

[4] N. W. Ashcroft and N. D. Mermin. *Solid State Physics*. Brooks/Cole, Belmont, CA, 1976.

[5] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a MATLAB toolbox for optimization on manifolds. *J. Mach. Learn. Res.*, 15:1455–1459, 2014.

[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(4):1207–1216, 2014.

[7] S. V. Dolgov and D. V. Savostyanov. Alternating minimal energy methods for linear systems in higher dimensions. *SIAM J. Sci. Comput.*, 36(5):A2248–A2271, 2014.

[8] S. V. Dolgov and D. V. Savostyanov. Corrected one-site density matrix renormalization group and alternating minimal energy algorithm. In *Numerical Mathematics and Advanced Applications—ENUMATH 2013*, volume 103 of *Lect. Notes Comput. Sci. Eng.*, pages 335–343. Springer, Cham, 2015.

[9] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353, 1999.

[10] V. Guillemin and A. Pollack. *Differential Topology*. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1974.

[11] W. Hackbusch. Entwicklungen nach Exponentialsummen. Technical report, Max Planck Institute for Mathematics in the Sciences, 2005.

[12] W. Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*. Springer, Heidelberg, 2012.

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

[14] B. L. Hammond, W. A. Lester, and P. J. Reynolds. *Monte Carlo Methods in Ab Initio Quantum Chemistry*. World Scientific, Singapore, 1994.

[15] M. B. Hastings. An area law for one-dimensional quantum systems. *J. Stat. Mech.*, 2007(08):P08024, 2007.

[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.

[17] S. Holtz, T. Rohwedder, and R. Schneider. On manifolds of tensors of fixed TT-rank. *Numer. Math.*, 120(4):701–731, 2012.

[18] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, second edition, 2013.

[19] R. O. Jones. Density functional theory: Its origins, rise to prominence, and future. *Rev. Mod. Phys.*, 87(3):897, 2015.

[20] B. N. Khoromskij. $O(d \log N)$-quantics approximation of N-d tensors in high-dimensional numerical modeling. *Constr. Approx.*, 34(2):257–280, 2011.

[21] B. N. Khoromskij. *Tensor Numerical Methods in Scientific Computing*. De Gruyter, Berlin, 2018.

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

[23] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Rev.*, 51(3):455–500, 2009.

[24] D. Kressner, M. Steinlechner, and A. Uschmajew. Low-rank tensor methods with subspace correction for symmetric eigenvalue problems. *SIAM J. Sci. Comput.*, 36(5):A2346–A2368, 2014.

[25] J. I. Latorre, E. Rico, and G. Vidal. Ground state entanglement in quantum spin chains. *Quant. Inf. Comput.*, 4(1):48–92, 2004.

[26] Ö. Legeza, J. Röder, and B. A. Hess. QC-DMRG study of the ionic–neutral curve crossing of LiF. *Mol. Phys.*, 101:2019–2028, 2002.

[27] J. H. Manton. Optimization algorithms exploiting unitary constraints. *IEEE Trans. Signal Process.*, 50(3):635, 2002.

[28] R. Olivares-Amaya, W. Hu, N. Nakatani, S. Sharma, J. Yang, and G. K.-L. Chan. The ab-initio density matrix renormalization group in practice. *J. Chem. Phys.*, 142(3):034102, 2015.

[29] I. V. Oseledets. Tensor-train decomposition. *SIAM J. Sci. Comput.*, 33(5):2295–2317, 2011.

[30] 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.

[31] I. Pižorn and F. Verstraete. Variational numerical renormalization group: bridging the gap between NRG and Density Matrix Renormalization Group. *Phys. Rev. Lett.*, 108(6):067202, 2012.

[32] M. V. Rakhuba and I. V. Oseledets. Jacobi-Davidson method on low-rank matrix manifolds. *SIAM J. Sci. Comput.*, 40(2):A1149–A1170, 2018.

[33] A. W. Sandvik. Computational studies of quantum spin systems. *AIP Conf. Proc.*, 1297(1):135, 2010.

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

[35] A. Szabo and N. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Dover Publications, Mineola, NY, 1996.

[36] S. Szalay, M. Pfeffer, V. Murg, G. Barcza, F. Verstraete, R. Schneider, and Ö. Legeza. Tensor product methods and entanglement optimization for ab initio quantum chemistry. *Int. J. Quantum Chem.*, 115(19):1342–1391, 2015.

[37] B. Vandereycken. Low-rank matrix completion by Riemannian optimization. *SIAM J. Optim.*, 23(2):1214–1236, 2013.

[38] F. Verstraete and J. I. Cirac. Matrix product states represent ground states faithfully. *Phys. Rev. B*, 73(9):094423, 2006.

[39] S. R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69(19):2863–2866, Nov 1992.

[40] S. R. White. Density matrix renormalization group algorithms with a single center site. *Phys. Rev. B*, 72(18):180403, 2005.