A FLEXIBLE POWER METHOD FOR SOLVING INFINITE DIMENSIONAL TENSOR EIGENVALUE PROBLEMS[*]

ROEL VAN BEEUMEN[†], LANA PERIŠA[‡], DANIEL KRESSNER[§], AND CHAO YANG[†]

Abstract. We propose a flexible power method for computing the leftmost, i.e., algebraically smallest, eigenvalue of an infinite dimensional tensor eigenvalue problem, $Hx = \lambda x$, where the infinite dimensional symmetric matrix H exhibits a translational invariant structure. We assume the smallest eigenvalue of H is simple and apply a power iteration of e^{-H} with the eigenvector represented in a compact way as a translational invariant infinite Tensor Ring (iTR). Hence, the infinite dimensional eigenvector can be represented by a finite number of iTR cores of finite rank. In order to implement this power iteration, we use a small parameter t so that the infinite matrix-vector operation $e^{-Ht}x$ can efficiently be approximated by the Lie product formula, also known as Suzuki–Trotter splitting, and we employ a low rank approximation through a truncated singular value decomposition on the iTR cores in order to keep the cost of subsequent power iterations bounded. We also use an efficient way for computing the iTR Rayleigh quotient and introduce a finite size iTR residual which is used to monitor the convergence of the Rayleigh quotient and to modify the timestep t. In this paper, we discuss 2 different implementations of the flexible power algorithm and illustrate the automatic timestep adaption approach for several numerical examples.

Key words. infinite eigenvalue problem, tensor eigenvalue problem, tensor ring, tensor train, translational invariance, matrix product states

AMS subject classifications. 15A18, 15A21, 15A69, 65F15

1. Introduction.

We consider the problem of computing the leftmost, i.e., algebraically smallest, eigenvalue of an *infinite dimensional* tensor eigenvalue problem

$$\mathbf{H}\mathbf{x} = \lambda \mathbf{x}, \tag{1.1}$$

where the infinite dimensional symmetric matrix $\mathbf{H}$ is defined as the infinite sum of Kronecker products of an infinite number of finite matrices, i.e.,

$$\mathbf{H} = \sum_{k=-\infty}^{+\infty} \mathbf{H}_k, \qquad \mathbf{H}_k = \cdots \otimes I \otimes I \otimes M_{k,k+1} \otimes I \otimes I \otimes \cdots, \tag{1.2}$$

with I the $d \times d$ identity matrix and $M_{k,k+1} \in \mathbb{R}^{d^2 \times d^2}$. We assume that the matrices $M_{k,k+1}$ are identical for all k, hence $\mathbf{H}$ is called to be *translational invariant*. The double subscript $k, k+1$ for the matrix M is used here to merely indicate the overlapping positions of M in each Kronecker product. This type of eigenvalue problems originates from the study of simple quantum many-body systems such as a quantum spin chain with nearest neighbor interactions [3]. For some specific choices of M, the analytical solutions of these eigenvalue problems are known [2, 6, 1]. However, in general, the smallest eigenvalue and its corresponding eigenvector need to be computed numerically.

[*]Submitted to the editors February 2, 2021.

Funding: This work was partially funded by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through Advanced Computing (SciDAC) program under Contract No. DE-AC0205CH11231.

[†]Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, United States. (rvanbeeumen@lbl.gov, cyang@lbl.gov).

[‡]Visage Technologies, Diskettgatan 11A, SE-583 35 Linköping, Sweden. (lana.perisa@visagetechnologies.com).

[§]Department of Mathematics, École Polytechnique Fédérale de Lausanne, Station 8, 1015 Lausanne, Switzerland. (daniel.kressner@epfl.ch).

The eigenvectors of $\mathbf{H}$ are infinite dimensional vectors which obviously cannot be computed or stored directly in memory. One way to study such a problem computationally is to start with a finite $H_\ell \in \mathbb{R}^{d^\ell \times d^\ell}$ that only contains $(\ell - 1)$ terms in the summation and Kronecker products of $(\ell - 1)$ matrices, and examine how the smallest eigenvalue of H_ℓ changes as ℓ increases. Note that the Kronecker structure of H_ℓ allows for efficiently representing this matrix in, e.g., Tensor Train (TT) format [10]. The corresponding eigenvector can then be represented by the TT

$$x_\ell(i_1, i_2, \ldots, i_\ell) = X_1(i_1)X_2(i_2)\cdots X_\ell(i_\ell),$$

where $X_k(i_k)$ is an $r_k \times r_{k+1}$ matrix, with $r_1 = r_{\ell+1} = 1$, and the indices $i_k = 1, \ldots, d$, for $k = 1, \ldots, \ell$. Because there is a limit on how large ℓ can be chosen, we may never know the true solution in the limit $\ell \to \infty$. Such a limit is known in the physics literature as the *thermodynamic limit* and is important for describing macroscopic properties of quantum materials when $\mathbf{H}$ corresponds to a quantum many-body Hamiltonian [14].

In this paper, we will directly compute the smallest eigenvalue, and corresponding eigenvector, of the infinite dimensional eigenvalue problem (1.1)–(1.2). In order to do that, we represent the eigenvector in a compact form and make use of the translational invariance property of $\mathbf{H}$. Such invariance property was studied by Bethe [2] and Hulthén [6], known as the *Bethe–Hulthén hypothesis*, and suggests that the elements of the eigenvector are invariant with respect to a cyclic permutation of the tensor indices, i.e.,

$$\mathbf{x}(\ldots, i_{-1}, i_0, i_1, \ldots) = \mathbf{x}(\ldots, i_0, i_1, i_2, \ldots),$$

where all indices $i_k = 1, \ldots, d$, and the sequence of indices $\ldots, i_{-1}, i_0, i_1 \ldots$ specifies a particular element of the infinite dimensional vector $\mathbf{x}$.

We assume that the smallest eigenvalue of $\mathbf{H}$ is simple and propose to apply a flexible power iteration to $e^{-\mathbf{H}t}$ for some small and variable parameter t to compute the desired eigenpair. We represent the eigenvector to be computed as a translational invariant *infinite Tensor Ring* (iTR), defined as follows

$$\mathbf{x}(\ldots, i_{-1}, i_0, i_1, \ldots) = \operatorname{Tr}\left[\prod_{k=-\infty}^{+\infty} X(i_k)\right],$$

where all indices $i_k = 1, \ldots, d$, and each $X(i_k)$ is a matrix of size $r \times r$. Note that due to the translational invariance, we only need to store, and work, with d matrices of size $r \times r$, which is manageable as long as the rank r is not too large. An iTR can be seen as the infinite limit of a finite size tensor ring [17] and is also known as a uniform Matrix Product State (uMPS) [16].

In order to implement the power method, we must be able to multiply a matrix exponential with an iTR and keep the product in iTR form. Computing $e^{-\mathbf{H}t}\mathbf{x}$, with $\mathbf{H}$ being an infinite dimensional matrix and $\mathbf{x}$ an iTR, is in general not possible. However, the special structure of (1.2) allows us to split $\mathbf{H}$ in even and odd terms, $\mathbf{H}_\mathrm{e}$ and $\mathbf{H}_\mathrm{o}$, respectively. When t is sufficiently small, we can use the Lie product formula, also known as Suzuki–Trotter splitting, to simplify $e^{-\mathbf{H}t}\mathbf{x}$ into only local tensor contractions with a $d^2 \times d^2$ matrix exponential e^{-Mt}. Such contractions can be implemented in a way that the translational invariance of the contracted tensor is preserved, although the rank generally increases. To keep the cost of subsequent power iterations bounded, a low rank approximation through the use of a truncated singular value decomposition is applied.

We will use the translational invariant iTR $\mathbf{x}$, obtained from the power method, to approximate the smallest eigenvalue of $\mathbf{H}$. Because (1.2) contains an infinite number of terms in the summation, the classical definition of the Rayleigh quotient involves a diverging infinite sum. Therefore, we define the iTR Rayleigh quotient in an average sense, and introduce the canonical form of an iTR that can simplify such a definition. In the numerical experiments, we will show that the iTR Rayleigh quotient used within the flexible power method converges from above to the exact solution. We also give an alternative way to obtain an eigenvalue approximation that can be more accurate than the Rayleigh quotient. Such an approximation relies on viewing an iTR as the product of an infinite sized frame matrix and a vectorized tensor of finite size.

The flexible power iteration was first introduced by Vidal in [15] as the iTEBD algorithm. We call such a scheme a flexible power iteration because it is similar to applying a power method to the matrix exponential $e^{-\mathbf{H}t}$, except that the matrix exponential is approximated, in each power iteration, through Trotter splitting, and rank truncation is applied to the iTR representation of $e^{-\mathbf{H}t}\mathbf{x}$. This is where the term "flexible" comes from. In [15] and other papers, the accuracy of the eigenvalue approximate is assessed by either comparing the approximation with the exact solution or examining the difference between the approximate eigenvalues obtained from two consecutive iterations. The former is not practical when the exact solution of the problem is unknown. The latter can be misleading when the flexible iteration starts to stagnate, which does happen as shown in our numerical experiments.

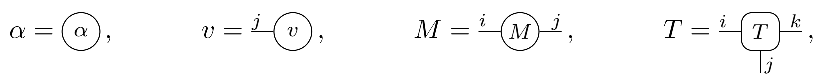
Based on the canonical form of the iTR, we also introduce a finite size iTR residual that can be used to monitor the convergence of the Rayleigh quotient. We give a practical procedure for adjusting the parameter t in the flexible power iteration to ensure rapid but stable convergence and provide a stopping criterion for terminating the power iteration. As illustrated in the numerical experiments, such a procedure is very effective.

The paper is organized as follows. In section 2, we introduce diagrammatic notations for scalars, vectors, matrices, and tensors, as well as for tensor operations that make it easier to explain operations performed on tensor rings. In section 3, we begin with a formal definition of a single core translational invariant infinite Tensor Ring, describe its properties, and show how an iTR can be converted into a so-called canonical form, which is convenient for manipulating iTRs in various calculations. We also extend these definitions and properties to 2-core translational invariant iTRs which are used to represent the approximate eigenvector of $\mathbf{H}$ (1.2) containing nearest neighbor interactions. In section 4, we properly define the iTR Rayleigh quotient and introduce the finite iTR residual. In section 5, we present the flexible power iteration for computing the smallest eigenpair of $\mathbf{H}$. We also show how $e^{\mathbf{H}t}\mathbf{x}$ is computed to preserve the iTR structure and discuss a number of practical computational considerations regarding the convergence and error assessment of the method. Three numerical examples are given in section 6 to demonstrate the effectiveness of the flexible power method and the adaptive strategy proposed in section 5. Finally, the main conclusions are summarized in section 7.

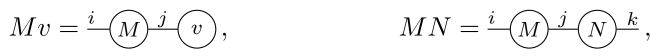
2. Notation.

Throughout the paper, we denote scalars by lower Greek characters, vectors by lower Roman characters, and matrices and higher order tensors by upper Roman characters, e.g., α, x, and M, respectively. I_d is the $d \times d$ identity matrix and diagonal matrices are denoted by upper Greek characters, e.g., Σ. The trace of a matrix M is given by $\mathrm{Tr}(M)$ and its vectorization by $\mathrm{vec}(M)$. For infinite dimensional vectors and matrices we use bold face Roman characters, e.g., $\mathbf{v}$ and $\mathbf{M}$, respectively.

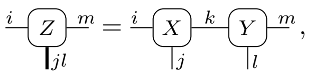
We will make use of the powerful tensor diagram notation [11] in order to graphically represent tensor contractions. The first 4 low-order tensors, i.e., a scalar α, a vector v, a matrix M, and a 3rd-order tensor T, are depicted as follows



where i, j, k are the corresponding tensor indices. This graphical notation has the advantage that tensor contractions can be represented by connecting the lines that correspond to the indices to sum over. For example, the matrix-vector product and the matrix-matrix product are given by, respectively,



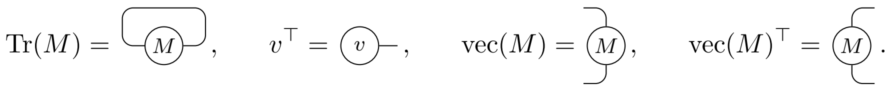
where the connections between M and v, and M and N correspond to the sum over index j. Because contracting a tensor over one of its indices with the identity matrix has no effect, we represent the identity matrix I by just a line. Another important tensor contraction, that we will often use, is the contraction of two 3rd-order tensors into a so called *supercore*



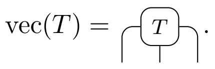
$$Z_{ijlm} = \sum_k X_{ijk} Y_{klm},$$

where the thick leg indicates combining the indices j and l into a single index.

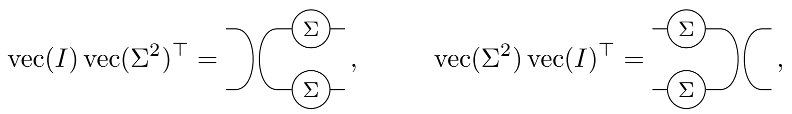
Some other commonly used matrix operations such as the trace, the transpose, and the vectorizations of a matrix are given by the following diagrams



In order to save space, we sometimes rotate the diagrams counterclockwise by 90 degrees, e.g., the vectorization of a 3-rd order tensor



Throughout the paper, reshaping tensors into matrices and vice versa will play a key role, for example, reshaping the following rank-1 matrices into 4th-order tensors

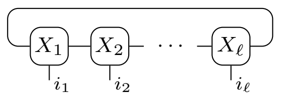
(2.1)


where Σ is a diagonal matrix. We again obtain a matrix by combining respectively the left and right pointing legs into one index.

3. Translational invariant infinite tensor ring. In this section, we define the translational invariant infinite tensor ring and discuss its properties that will be used in the next sections to construct an approximation to the eigenvector associated with the smallest eigenvalue of the infinite dimensional matrix $\mathbf{H}$ defined in (1.2). We start by defining translational invariance for a (finite) tensor ring.

3.1. Finite tensor ring. The tensor ring decomposition [17, 8] is a way to represent high-order tensors, or vectors that can be reshaped into high-order tensors, by a sequence of 3rd-order tensors that are circularly multiplied. We define a finite size tensor ring as follows.

Definition 3.1 (Tensor Ring). *Let x_ℓ be an ℓth-order tensor of size $d_1 \times d_2 \times \cdots \times d_\ell$, then its tensor ring representation is defined as*

(3.1) $x_\ell =$ 

and element-wise as $x_\ell(i_1, i_2, \ldots, i_\ell) := \operatorname{Tr}[X_1(i_1)X_2(i_2)\cdots X_\ell(i_\ell)]$, *where the indices i_k, $k = 1, 2, \ldots, \ell$, run from 1 to d_k, and each $X_k(i_k)$ is a matrix of size $r_k \times r_{k+1}$, with $r_1 = r_{\ell+1}$.*

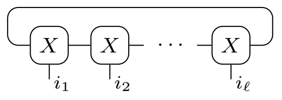
We call the 3rd-order tensors X_k the *cores* of x_ℓ, $X_k(i_k)$ the *slices* of the kth core, and r_k the corresponding tensor ring *ranks*. In the physics literature, a tensor ring is called a matrix product state with periodic boundary conditions [12]. When $r_1 = r_{\ell+1} = 1$, a tensor ring becomes a *tensor train* [10], also known in the physics literature as a matrix product state with open boundary conditions [12].

When all dimensions $d_k = d$, all ranks $r_k = r$, and all slices $X_k(i_k) = X(i_k)$, for $k = 1, 2, \ldots, \ell$, the tensor ring x_ℓ is said to be translational invariant. It follows from the trace operation property that

$$x_\ell(i_1, i_2, \ldots, i_\ell) = x_\ell(i_2, \ldots, i_\ell, i_1),$$

i.e., x_ℓ is invariant under cyclic permutations of its indices. This is a desirable property because the eigenvectors of $\mathbf{H}$ defined in (1.2) are known to have such a property for certain $M_{k,k+1}$. We define a translational invariant (finite) tensor ring as follows.

Definition 3.2 (Translational invariant TR). *Let x_ℓ be an ℓth-order tensor of size d^ℓ, then its translational invariant tensor ring representation is defined as*

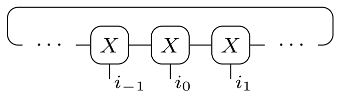
(3.2) $x_\ell =$ 

and element-wise as $x_\ell(i_1, i_2, \ldots, i_\ell) := \operatorname{Tr}[X(i_1)X(i_2)\cdots X(i_\ell)]$, *where the indices i_k, $k = 1, 2, \ldots, \ell$, run from 1 to d, and each $X(i_k)$ is a matrix of size $r \times r$.*

Note that, due to the translational invariance, the tensor ring (3.2) is fully determined by only one core X of size $r \times d \times r$.

3.2. Infinite tensor ring. Generalizing Definition 3.1 to tensor rings with an infinite number of cores is impossible because this would require an infinite amount of data to represent $\mathbf{x}$. On the other hand, using the translational invariance property allows us to generalize Definition 3.2 to infinite tensor rings (iTR).

Definition 3.3 (Translational invariant iTR). *A translational invariant iTR is defined as*

$\mathbf{x} =$ 

and element-wise as

$$\mathbf{x}(\ldots, i_{-1}, i_0, i_1, \ldots) := \operatorname{Tr}\left[\prod_{k=-\infty}^{+\infty} X(i_k)\right], \tag{3.3}$$

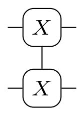
where all indices i_k run from 1 to d and each $X(i)$ is a matrix of size $r \times r$, with r referred to as the rank *of* $\mathbf{x}$.

Note that even though $\mathbf{x}$ contains an infinite number of cores, each core is completely defined by the d *slices* of X so that $\mathbf{x}$ can be represented by a finite amount of data although it is infinite dimensional. A translational invariant iTR is also known as a uniform matrix product state [16]. In the remainder of the paper, we will only work with translational invariant iTRs and just call them iTRs for brevity.

3.3. Transfer matrix. The core tensor X of each iTR defines a special matrix called *transfer matrix*, which plays a key role in many operations performed on the corresponding iTR, e.g., in the Rayleigh quotient and the residual calculations.

Definition 3.4 (Transfer matrix). *Let* $\mathbf{x}$ *be an iTR as defined in Definition 3.3. Then we define the transfer matrix T_X associated with* $\mathbf{x}$ *as the $r^2 \times r^2$ matrix*

$$T_X := \sum_{i=1}^{d} X(i) \otimes X(i) = \tag{3.4}$$



,

where $X(i)$ is the ith slice of X.

The transfer matrix (3.4) and in particular its dominant left and right eigenvectors, i.e., the eigenvectors corresponding to the eigenvalue with largest magnitude, will play an indispensable role in operations involving iTRs.

Conjecture 3.5. *Let* $\mathbf{x}$ *be a real iTR as defined in Definition 3.3 and assume that the dominant eigenvalue of the corresponding transfer matrix T_X is simple. Then this eigenvalue, denoted by η, is real and positive, and the corresponding left and right eigenvectors v_L and v_R satisfy the conditions*

$$\sum_{i=1}^{d} X(i)^\top V_L X(i) = \eta V_L, \qquad \sum_{i=1}^{d} X(i) V_R X(i)^\top = \eta V_R, \tag{3.5}$$

where $v_L =: \operatorname{vec}(V_L)$ and $v_R =: \operatorname{vec}(V_R)$, with V_L and V_R positive definite matrices.

When the dominant eigenvalue of the transfer matrix is simple, $v_L^\top v_R \neq 0$ holds. We will assume that the eigenvectors v_L and v_R satisfy the normalization condition $v_L^\top v_R = 1$, which is equivalent to $\operatorname{Tr}\left(V_L^\top V_R\right) = 1$. A graphically depiction of (3.5), and the normalization condition for v_L and v_R are shown in Figure 1.

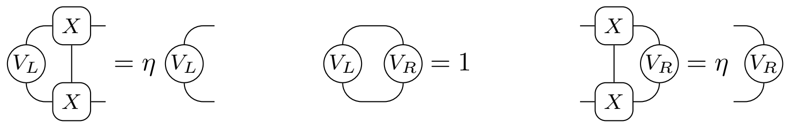


Fig. 1. *A graphical representation of (3.5), and the normalization condition $v_L^\top v_R = 1$.*

Lemma 3.6. *If the transfer matrix T_X is an injective linear map and η is its dominant eigenvalue with v_L and v_R being the corresponding left and right eigenvectors, respectively normalized to satisfy $v_L^\top v_R = 1$, then*

$$\lim_{k \to \infty} \left(\frac{T_X}{\eta}\right)^k = v_R v_L^\top. \tag{3.6}$$

Proof. Let the left and right eigenvalue decompositions of T_X be given by

$$W_L^* T_X = \Lambda W_L^*, \qquad T_X W_R = W_R \Lambda \qquad W_L^* W_R = I,$$

where the diagonal matrix Λ contains the eigenvalues on its diagonal, and W_L and W_R are the corresponding left and right eigenvectors of T_X, respectively, and normalized so that $W_L^* W_R = I$, i.e.,

$$\Lambda := \begin{bmatrix} \eta & \bullet \end{bmatrix}, \qquad W_L := \begin{bmatrix} v_L & \bullet \end{bmatrix}, \qquad W_R := \begin{bmatrix} v_R & \bullet \end{bmatrix}.$$

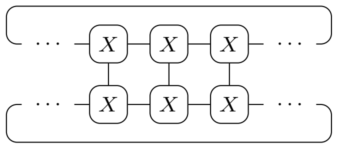
Then, using that η is simple and $W_R^{-1} = W_L^*$, yields

$$\lim_{k\to\infty} \left(\frac{T_X}{\eta}\right)^k = \lim_{k\to\infty} \frac{1}{\eta^k} W_R \Lambda^k W_R^{-1} = \lim_{k\to\infty} \frac{1}{\eta^k} W_R \Lambda^k W_L^* = v_R v_L^\top,$$

which completes the proof. ∎

Corollary 3.7 (Normalized iTR). *Let* $\mathbf{x}$ *be an iTR as defined in Definition 3.3. Then* $\mathbf{x}$ *is normalized, i.e.,* $\mathbf{x}^\top \mathbf{x} = 1$*, if its corresponding transfer matrix* T_X *has a simple dominant eigenvalue* $\eta = 1$*.*

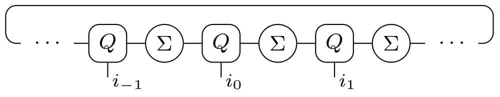
Proof. The proof directly follows from Lemma 3.6, i.e.,

$\mathbf{x}^\top \mathbf{x} =$  $= \operatorname{Tr}\left[\lim_{k\to\infty} (T_X)^k\right] = \operatorname{Tr}\left[v_R v_L^\top\right] = v_L^\top v_R = 1,$

where we also used the trace property $\operatorname{Tr}(AB) = \operatorname{Tr}(BA)$. ∎

3.4. Canonical decomposition. Because we can insert the product of any nonsingular matrix S and its inverse between two consecutive cores of an iTR and redefine each slice as $S^{-1}X(i)S$, $i = 1, ..., d$, the representation of an iTR is not unique. It is useful to define a canonical form that makes it easier to describe computations performed on iTRs.

Definition 3.8 (Canonical form). *Let* $\mathbf{x}$ *be an iTR as defined in Definition 3.3. Then its canonical form is defined as*

$\mathbf{x} =$ 

and element-wise as

$$\mathbf{x}(\ldots, i_{-1}, i_0, i_1, \ldots) := \operatorname{Tr}\left[\prod_{k=-\infty}^{+\infty} Q(i_k)\Sigma\right], \tag{3.7}$$

where $Q(i) \in \mathbb{R}^{r\times r}$*, for* $i = 1, \ldots, d$*, and* $\Sigma \in \mathbb{R}^{r\times r}$ *is a diagonal matrix with decreasing non-negative real numbers on its diagonal and* $\|\Sigma\|_F = 1$*, such that the following left and right orthogonality conditions hold*

$$\sum_{i=1}^{d} Q_L(i)^\top Q_L(i) = \eta I, \qquad \sum_{i=1}^{d} Q_R(i) Q_R(i)^\top = \eta I, \tag{3.8}$$

with $Q_L(i) := \Sigma Q(i)$*,* $Q_R(i) := Q(i)\Sigma$*, and* $\eta \in \mathbb{R}$ *being the dominant eigenvalue of the transfer matrix* T_X*.*

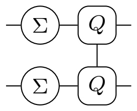
We call the 3rd-order tensor Q the orthogonal core of $\mathbf{x}$ and Σ the corresponding singular value matrix, since it can be obtained from a singular value decomposition. The 3rd-order tensors Q_L and Q_R, defined in Definition 3.8, are called the left and right orthogonal cores, respectively.

Before explaining how to compute the canonical form, we will list some properties. First, note that by definition, Q and Σ satisfy the following relations

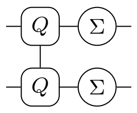
$$\Sigma Q(i)\Sigma = Q_L(i)\Sigma = \Sigma Q_R(i), \tag{3.9}$$

for $i = 1, \ldots, d$. The left and right canonical transfer matrices are defined as follows.

Definition 3.9 (Canonical transfer matrices). *Let $\mathbf{x}$ be an iTR in canonical form as defined in Definition 3.8. Then we define its left and right canonical transfer matrices T_{Q_L} and T_{Q_R} as the following $r^2 \times r^2$ matrices*

$$T_{Q_L} := \sum_{i=1}^{d} Q_L(i) \otimes Q_L(i) = \tag{3.10}$$


,

$$T_{Q_R} := \sum_{i=1}^{d} Q_R(i) \otimes Q_R(i) = \tag{3.11}$$


,

where Q_L and Q_R are defined in Definition 3.8.

The canonical transfer matrices, associated with an iTR in canonical form, and their dominant left and right eigenvectors, which play a crucial role in the evaluation of the iTR Rayleigh quotient and the iTR residual, have the following fixed form.

Lemma 3.10. *Let $\mathbf{x}$ be an iTR in canonical form as defined in Definition 3.8 with its corresponding canonical transfer matrices defined as in Definition 3.9. Then,*

$$\operatorname{vec}(I)^\top T_{Q_L} = \eta \operatorname{vec}(I)^\top, \qquad T_{Q_R} \operatorname{vec}(I) = \eta \operatorname{vec}(I), \tag{3.12}$$

$$T_{Q_L} \operatorname{vec}(\Sigma^2) = \eta \operatorname{vec}(\Sigma^2), \qquad \operatorname{vec}(\Sigma^2)^\top T_{Q_R} = \eta \operatorname{vec}(\Sigma^2)^\top, \tag{3.13}$$

with $\operatorname{vec}(I)$ *being the dominant left eigenvector of T_{Q_L} and the dominant right eigenvector of T_{Q_R}, and* $\operatorname{vec}(\Sigma^2)$ *being the dominant right eigenvector of T_{Q_L} and the dominant left eigenvector of T_{Q_R}.*

Proof. The proof for the dominant right eigenvector of T_{Q_R} directly follows from vectorizing the right equality of (3.8),

$$\operatorname{vec}\left(\sum_{i=1}^{d} Q_R(i) Q_R(i)^\top\right) = \left(\sum_{i=1}^{d} Q_R(i) \otimes Q_R(i)\right) \operatorname{vec}(I) = T_{Q_R} \operatorname{vec}(I) = \eta \operatorname{vec}(I),$$

where we made use of the "vec trick" identity, i.e.,

$$\operatorname{vec}(ABC) = (C^\top \otimes A) \operatorname{vec}(B), \tag{3.14}$$

and Definition 3.9, respectively. For the proof of the dominant left eigenvector of T_{Q_L}, we start from vectorizing the left equality of (3.8) and apply (3.14) to obtain

$$\operatorname{vec}\left(\sum_{i=1}^{d} Q_L(i)^\top Q_L(i)\right) = \left(\sum_{i=1}^{d} Q_L(i)^\top \otimes Q_L(i)^\top\right) \operatorname{vec}(I) = \eta \operatorname{vec}(I). \tag{3.15}$$

Transposing (3.15) and using Definition 3.9 completes the proof of (3.12).

In order to prove that $\mathrm{vec}(\Sigma^2)$ is the dominant right eigenvector of T_{Q_L}, we start again from the vectorization of the right equality of (3.8),

$$\mathrm{vec}\left(\sum_{i=1}^{d} Q(i)\Sigma^2 Q(i)^\top\right) = \left(\sum_{i=1}^{d} Q(i)\otimes Q(i)\right)\mathrm{vec}(\Sigma^2) = \eta\,\mathrm{vec}(I), \tag{3.16}$$

where we substituted $Q_R(i) = Q(i)\Sigma$ and applied (3.14). Next, we multiply both sides of (3.16) from the left by $\Sigma\otimes\Sigma$, yielding

$$\underbrace{(\Sigma\otimes\Sigma)\left(\sum_{i=1}^{d} Q(i)\otimes Q(i)\right)}_{=\,T_{Q_L}}\mathrm{vec}(\Sigma^2) = \eta\,(\Sigma\otimes\Sigma)\,\mathrm{vec}(I) = \eta\,\mathrm{vec}(\Sigma^2).$$

Finally, in order to prove that $\mathrm{vec}(\Sigma^2)$ is also the dominant left eigenvector of T_{Q_R}, we start from the vectorization of the left equality of (3.8),

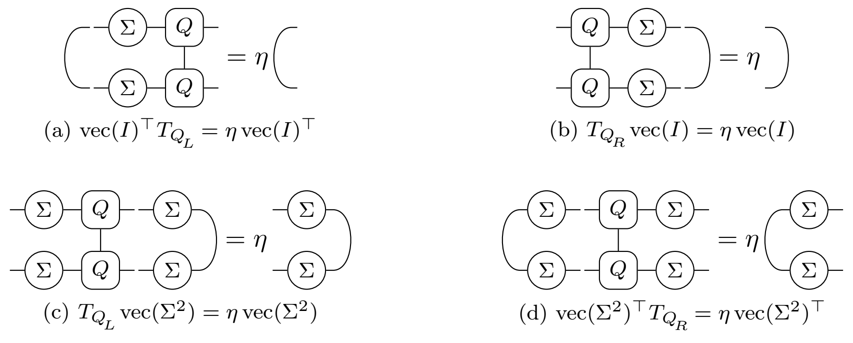
$$\mathrm{vec}\left(\sum_{i=1}^{d} Q(i)^\top\Sigma^2 Q(i)\right) = \left(\sum_{i=1}^{d} Q(i)^\top\otimes Q(i)^\top\right)\mathrm{vec}(\Sigma^2) = \eta\,\mathrm{vec}(I), \tag{3.17}$$

where we substituted $Q_L(i) = \Sigma Q(i)$ and made use of the identity (3.14). Next, we multiply both sides of (3.17) from the left by $\Sigma\otimes\Sigma$ followed by a transposition, yielding

$$\mathrm{vec}(\Sigma^2)^\top\underbrace{\left(\sum_{i=1}^{d} Q(i)\otimes Q(i)\right)(\Sigma\otimes\Sigma)}_{=\,T_{Q_R}} = \eta\,[(\Sigma\otimes\Sigma)\,\mathrm{vec}(I)]^\top = \eta\,\mathrm{vec}(\Sigma^2)^\top,$$

which proves (3.13) and completes the proof. ∎

The eigenvalue decompositions of Lemma 3.10 are graphically shown in Figure 2. Note also that the left and right orthogonality conditions (3.8), defined by the canonical form, can be expressed in terms of the canonical transfer matrices and are equivalent to (3.12).



(a) $\mathrm{vec}(I)^\top T_{Q_L} = \eta\,\mathrm{vec}(I)^\top$ (b) $T_{Q_R}\,\mathrm{vec}(I) = \eta\,\mathrm{vec}(I)$

(c) $T_{Q_L}\,\mathrm{vec}(\Sigma^2) = \eta\,\mathrm{vec}(\Sigma^2)$ (d) $\mathrm{vec}(\Sigma^2)^\top T_{Q_R} = \eta\,\mathrm{vec}(\Sigma^2)^\top$

FIG. 2. *Eigenvalue decompositions of the canonical transfer matrices.*

Using the orthogonality conditions (3.8), we notice that computing the canonical form of an iTR corresponds to finding matrices $Q(i)$ and a diagonal Σ so that

$$\sum_{i=1}^{d} Q(i)^\top \Sigma^2 Q(i) = \eta I = \sum_{i=1}^{d} Q(i)\Sigma^2 Q(i)^\top. \tag{3.18}$$

Algorithm 1 shows how the canonical decomposition of an iTR can be computed. Using the properties $L^{-1} = R\tilde{\Sigma}$, $R^{-1} = \tilde{\Sigma}L$, $V_R = R\tilde{\Sigma}^2R^\top$, and $V_L = L^\top\tilde{\Sigma}^2L$, in combination with the identity (3.14), one can show that Q and Σ obtained in step 6 of Algorithm 1 satisfy (3.18).

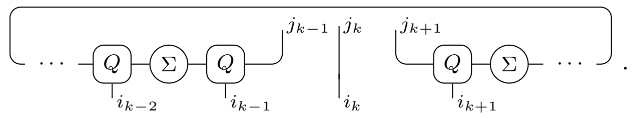
The complexity of Algorithm 1 is dominated by step 1, which requires the computation of the dominant left and right eigenvectors of the $r^2 \times r^2$ transfer matrix T_X. Note that all remaining steps 2–7 only require operations on $r \times r$ matrices.

Algorithm 1: Canonical decomposition of iTR

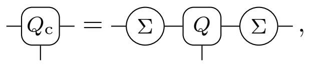
```
Input  : 3rd-order tensor $X$
Output: 3rd-order tensor $Q$ and diagonal matrix $\Sigma$
1 Dominant left and right eigenvectors of transfer matrix $T_X$:
      $\mathrm{vec}(V_L)^\top T_X = \eta\,\mathrm{vec}(V_L)^\top, \qquad T_X\,\mathrm{vec}(V_R) = \eta\,\mathrm{vec}(V_R).$
2 Eigendecompositions of $V_L$ and $V_R$:   $V_L = U_L\Lambda_L U_L^\top, \quad V_R = U_R\Lambda_R U_R^\top.$
3 Form:                                     $\tilde{U}_L = U_L\Lambda_L^{1/2}, \quad \tilde{U}_R = U_R\Lambda_R^{1/2}.$
4 Singular value decomposition:             $V\tilde{\Sigma}W^\top = \tilde{U}_L^\top\tilde{U}_R.$
5 Form:                                     $L = W^\top\tilde{U}_R^{-1}, \quad R = \tilde{U}_L^{-\top}V.$
6 Canonical form:                           $Q(i) = \|\tilde{\Sigma}\|_F\,LX(i)R, \quad \Sigma = \tilde{\Sigma}/\|\tilde{\Sigma}\|_F.$
7 (Normalization):                          $Q(i) = Q(i)/\sqrt{\eta}.$
```

3.5. Frame matrix. Starting from an iTR in canonical form, as defined in Definition 3.8, we can rewrite it as a product of an infinite dimensional matrix with orthogonal columns, called the *frame matrix* [4, 7], and a finite vector of length equal to the number of elements in 1 core.

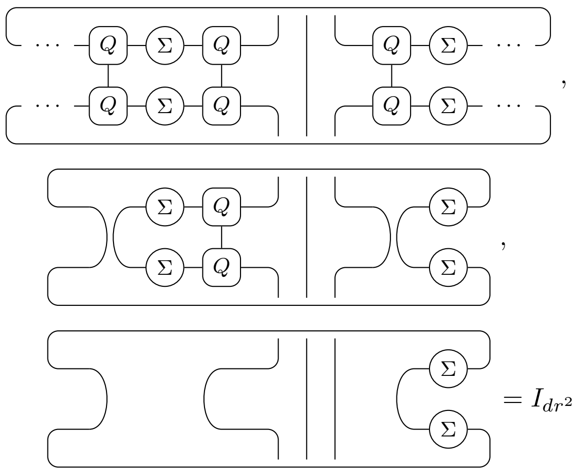
Definition 3.11 (Frame matrix). *Let* $\mathbf{x}$ *be a normalized iTR in canonical form as defined in Definition 3.8. Then we define its* k*th frame matrix as follows*

(3.19) $\mathbf{F}_{\neq k} :=$  .

Note that by this definition the frame matrix $\mathbf{F}_{\neq k}$ has an infinite number of rows, which are represented as down pointing legs, and dr^2 columns, which are represented as up pointing legs. Now the iTR $\mathbf{x}$ can also be written as

(3.20) $\mathbf{x} = \mathbf{F}_{\neq k}\,\mathrm{vec}(Q_c),$  , where Q_c is called the *canonical center core*. Another important property of the frame matrix is that it has orthonormal columns, i.e., $\mathbf{F}_{\neq k}^\top \mathbf{F}_{\neq k} = I_{dr^2}$, since

$\mathbf{F}_{\neq k}^\top \mathbf{F}_{\neq k} =$



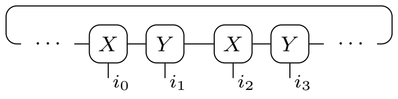
where we used in the first step Lemma 3.6, i.e., $\lim_{k\to\infty}(T_{Q_R})^k = \operatorname{vec}(I)\operatorname{vec}(\Sigma^2)^\top$, in order to replace the infinite repetition of T_{Q_R} on both sides of the free indices, followed by $\operatorname{vec}(I)^\top T_{Q_L} = \operatorname{vec}(I)^\top$, and $\operatorname{vec}(\Sigma^2)^\top \operatorname{vec}(I) = 1$, respectively.

Due to the orthogonality of the frame matrix, this matrix will turn out to be an important tool to extract and update cores in an efficient way. Further, this matrix will also play a role in splitting supercores into normal cores.

3.6. 2-core translational invariance. The translational invariance principle can be generalized to allow the unit of invariance to include more than one core. Such a unit is sometimes referred to as a *supercore*. We now define a 2-core translational invariant iTR, which we will refer to as iTR2 to distinguish it from iTRs with a single core.

Definition 3.12 (2-core translational invariant iTR). *A 2-core translational invariant iTR is defined as*

$\mathbf{x} =$



and element-wise as

$$\mathbf{x}(\ldots, i_0, i_1, i_2, i_3, \ldots) := \operatorname{Tr}\left[\prod_{k=-\infty}^{+\infty} X(i_{2k})\, Y(i_{2k+1})\right], \tag{3.21}$$

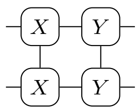
where all indices i_k run from 1 to d and each $X(i)$ and $Y(i)$ are matrices of size $r \times r$.

In contrast to an iTR with a single core, there are now 2 transfer matrices associated with every iTR2, playing a central role in the iTR2 operations.

Definition 3.13 (iTR2 transfer matrices). *Let $\mathbf{x}$ be an iTR2 as defined in Definition 3.12. Then we define the transfer matrices T_{XY} and T_{YX} associated with*

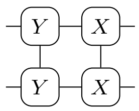
$\mathbf{x}$ *as the* $r^2 \times r^2$ *matrices*

$$T_{XY} := \sum_{i=1}^{d}\sum_{j=1}^{d}\Big[X(i)\otimes X(i)\Big]\Big[Y(j)\otimes Y(j)\Big] = \tag{3.22}$$



,

$$T_{YX} := \sum_{i=1}^{d}\sum_{j=1}^{d}\Big[Y(i)\otimes Y(i)\Big]\Big[X(j)\otimes X(j)\Big] = \tag{3.23}$$



,

where $X(i)$ *is the* i*th slice of* X *and* $Y(j)$ *the* j*th slice of* Y*.*

Note that the transfer matrices T_{XY} and T_{YX} of an iTR2 can also be expressed in terms of the 1-core transfer matrices, i.e., $T_{XY} = T_X T_Y$ and $T_{YX} = T_Y T_X$, where T_X is the transfer matrix formed by only using the core X and T_Y the transfer matrix formed by only using the core Y.

Lemma 3.14. *Let* $\mathbf{x}$ *be an iTR2 as defined in Definition 3.12 and assume that* T_X *and* T_Y *are nonsingular. Then its corresponding transfer matrices* T_{XY} *and* T_{YX}*, defined in Definition 3.13, have the same eigenvalues.*

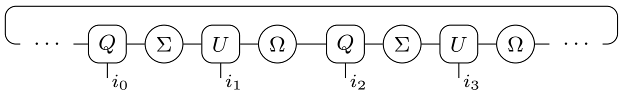
Proof. We will show that every eigenvalue of T_{XY} is also an eigenvalue T_{YX}, and vice versa. First, assume that (λ, v) is an eigenpair of T_{XY}, $T_{XY}v = T_X T_Y v = \lambda v$. Multiplying both sides from the left by T_Y, yields $T_Y T_X(T_Y v) = \lambda(T_Y v)$, hence, the pair $(\lambda, w := T_Y v)$ is an eigenpair of T_{YX}.

Similarly, assume that (λ, w) is an eigenpair of T_{YX}, $T_{YX}w = T_Y T_X w = \lambda w$. Multiplying both sides from the left by T_X, yields again $T_X T_Y(T_X w) = \lambda(T_X w)$, hence, the pair $(\lambda, v := T_X w)$ is an eigenpair of T_{XY}. ∎

We now define the canonical form of an iTR2. Since the canonical form is mainly used for normalized iTRs and iTR2s, we will assume that the dominant eigenvalue of its transfer matrices $\eta = 1$.

Definition 3.15 (iTR2 canonical form). *Let* $\mathbf{x}$ *be an iTR2 as defined in Definition 3.12. Then its canonical form is defined as*

$\mathbf{x} =$



and element-wise as

$$\mathbf{x}(\ldots, i_0, i_1, i_2, i_3, \ldots) := \operatorname{Tr}\left[\prod_{k=-\infty}^{+\infty} Q(i_{2k})\,\Sigma\, U(i_{2k+1})\,\Omega\right], \tag{3.24}$$

where $Q(i), U(i) \in \mathbb{R}^{r\times r}$*, for* $i = 1, \ldots, d$*, and* $\Sigma, \Omega \in \mathbb{R}^{r\times r}$ *are diagonal matrices with decreasing non-negative real numbers on its diagonal and* $\|\Sigma\|_F = \|\Omega\|_F = 1$*, such that the following left and right orthogonality conditions hold*

$$\sum_{i=1}^{d} Q_L(i)^\top Q_L(i) = I, \qquad \sum_{i=1}^{d} Q_R(i) Q_R(i)^\top = I, \tag{3.25}$$

$$\sum_{i=1}^{d} U_L(i)^\top U_L(i) = I, \qquad \sum_{i=1}^{d} U_R(i) U_R(i)^\top = I, \tag{3.26}$$

with $Q_L(i) := \Omega Q(i)$*,* $Q_R(i) := Q(i)\Sigma$*,* $U_L(i) := \Sigma U(i)$*, and* $U_R(i) := U(i)\Omega$*.*

Due to a translational invariance of 2 cores for an iTR2, its canonical form gives rise to 4 canonical transfer matrices with structured dominant eigenvectors as detailed in the following definition and lemma.

DEFINITION 3.16 (iTR2 canonical transfer matrices). *Let* $\mathbf{x}$ *be an iTR2 in canonical form as defined in Definition 3.15. Then we define its left canonical transfer matrices as the* $d^2 \times d^2$ *matrices* $T_{Q_L U_L}$ *and* $T_{U_L Q_L}$*, and its right canonical transfer matrices as* $T_{Q_R U_R}$ *and* $T_{U_R Q_R}$*, i.e.,*

$$T_{Q_L U_L} = T_{Q_L} T_{U_L}, \qquad T_{Q_R U_R} = T_{Q_R} T_{U_R}, \tag{3.27}$$

$$T_{U_L Q_L} = T_{U_L} T_{Q_L}, \qquad T_{U_R Q_R} = T_{U_R} T_{Q_R}, \tag{3.28}$$

where Q_L*,* Q_R*,* U_L*, and* U_R *are defined in Definition 3.15.*

LEMMA 3.17. *Let* $\mathbf{x}$ *be an iTR2 in canonical form as defined in Definition 3.15 and its corresponding canonical transfer matrices as in Definition 3.16. Then,*

$$T_{Q_L U_L} \operatorname{vec}(\Omega^2) = \operatorname{vec}(\Omega^2), \qquad \operatorname{vec}(\Omega^2)^\top T_{Q_R U_R} = \operatorname{vec}(\Omega^2)^\top, \tag{3.29}$$

$$T_{U_L Q_L} \operatorname{vec}(\Sigma^2) = \operatorname{vec}(\Sigma^2), \qquad \operatorname{vec}(\Sigma^2)^\top T_{U_R Q_R} = \operatorname{vec}(\Sigma^2)^\top, \tag{3.30}$$

or $\operatorname{vec}(\Omega^2)$ *being the dominant right eigenvector of* $T_{Q_L U_L}$ *and the dominant left eigenvector* $T_{Q_R U_R}$*, and* $\operatorname{vec}(\Sigma^2)$ *being the dominant right eigenvector of* $T_{U_L Q_L}$ *and the dominant left eigenvector* $T_{U_R Q_R}$*.*

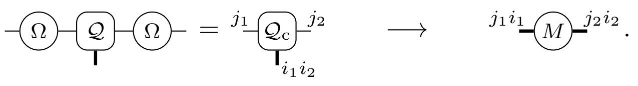
Proof. The proof is similar to the one of Lemma 3.10. ∎

The canonical decomposition of an iTR2 can be computed by making use of Algorithm 1 applied to the supercore, followed by a singular value decomposition of the matrization of the obtained canonical center supercore. This procedure is summarized in Algorithm 2. In case $d \ll r$, the complexity of Algorithm 2 is dominated by step 2, which requires again computing the dominant left and right eigenvectors of the $r^2 \times r^2$ transfer matrix in Algorithm 1.

Algorithm 2: Canonical decomposition of iTR2

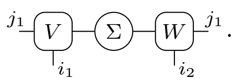
Input : 3rd-order tensors X and Y
Output: 3rd-order tensors Q and U, and diagonal matrices Σ and Ω

```
1 Form supercore $\mathcal{Z}$ as the contraction of the cores $X$ and $Y$.
2 Use Algorithm 1 on $\mathcal{Z}$ resulting in the supercore $\mathcal{Q}$ and the diagonal matrix
    $\Omega$.
3 Form canonical center supercore $\mathcal{Q}_c$ and reshape into a matrix $M$:
```



```
4 Singular value decomposition of $M$ and reshape back into cores:
```

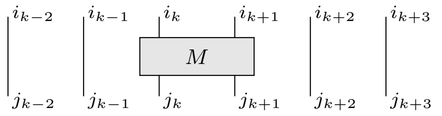
$$V \Sigma W^\top = M \qquad \longrightarrow$$



```
5 Canonical form:    $Q(i) = \Omega^{-1} V(i)$,    $U(j) = W(j)\Omega^{-1}$.
```

4. Eigenvalue approximation. Before we describe the algorithm for computing the leftmost eigenvalue and its corresponding eigenvector, we first need to define an eigenpair of an infinite dimensional matrix $\mathbf{H}$. The use of the iTR *ansatz* for the approximate eigenvector also requires special attention to be paid to the definition of the Rayleigh quotient and corresponding residual.

First, we introduce the graphical notation of the infinite dimensional matrices $\mathbf{H}_k$ defined in (1.2)

$\mathbf{H}_k = \cdots$ 

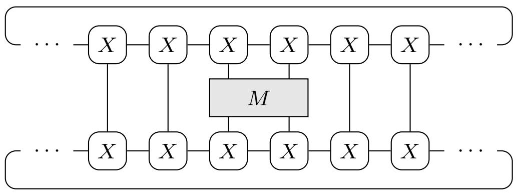
$\cdots,$

where $M \in \mathbb{R}^{d^2 \times d^2}$. Using this notation, the infinite dimensional matrix vector product $\mathbf{H}_k\mathbf{x}$, where $\mathbf{x}$ is an iTR, corresponds to contracting along all i indices.

4.1. Rayleigh quotient. Let $\mathbf{H}$ be the infinite dimensional matrix given in (1.2) and $\mathbf{x}$ a normalized iTR that is an approximation of an eigenvector of $\mathbf{H}$. Because (1.2) contains an infinite number of terms in the summation, the classical definition of the Rayleigh quotient involves an infinite sum, i.e.,

$$\mathbf{x}^\top \mathbf{H}\mathbf{x} = \sum_{k=-\infty}^{+\infty} \mathbf{x}^\top \mathbf{H}_k \mathbf{x}, \tag{4.1}$$

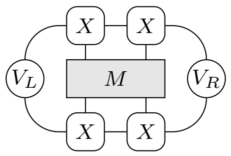
where

(4.2) $\mathbf{x}^\top \mathbf{H}_k \mathbf{x} =$ 

,

for all k and M is a $d^2 \times d^2$ matrix that only interacts with two nearest neighbor cores of $\mathbf{x}$ (and $\mathbf{x}^\top$). In this case, $\mathbf{H}_k$ is said to be a *local* nearest neighbor operator. Since all terms in (4.1) contribute equally to the summation, we define the Rayleigh quotient in an *average sense* as follows.

THEOREM 4.1 (Rayleigh quotient). *Let* $\mathbf{x}$ *be a normalized nonzero iTR as defined in Definition* 3.3 *and its corresponding transfer matrix* T_X *as in Definition* 3.4*. Then the Rayleigh quotient for a given infinite dimensional matrix* $\mathbf{H}$ *of the form* (1.2) *can be represented by*

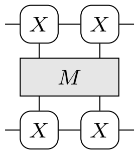
(4.3) $\theta =$ 

,

where V_L *and* V_R *are, respectively, the matricizations of the left and right dominant eigenvectors of* T_X*.*

Proof. We first rewrite (4.2) in matrix notation

$$\mathbf{x}^\top \mathbf{H}_k \mathbf{x} = \operatorname{Tr}\left[\left(\prod_{\ell=-\infty}^{k-1} T_X\right) \tilde{M} \left(\prod_{\ell=k+2}^{+\infty} T_X\right)\right], \qquad \tilde{M} := \tag{4.4}$$



.

It follows from Lemma 3.6 that

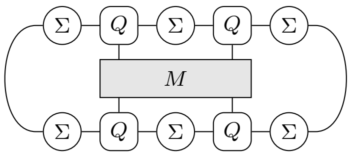
$$\mathbf{x}^\top \mathbf{H}_k \mathbf{x} = \operatorname{Tr}\left[v_R v_L^\top \, \tilde{M} \, v_R v_L^\top\right] = \operatorname{Tr}\left[v_L^\top \, \tilde{M} \, v_R v_L^\top v_R\right] = v_L^\top \tilde{M} v_R = \theta,$$

where we consecutively used the trace cyclic property $\operatorname{Tr}(XYZ) = \operatorname{Tr}(YZX)$ and the normalization of the left and right eigenvectors $v_L^\top v_R = 1$. ∎

If $\mathbf{x}$ is a normalized iTR, the evaluation of the Rayleigh quotient only requires multiplying the matrix $\tilde{M}$ (4.4) from the left and right, respectively, with the left and right dominant eigenvectors of its corresponding transfer matrix T_X. In case $\mathbf{x}$ is given in canonical form, the Rayleigh quotient simplifies further because there is no need to compute the left and right dominant eigenvectors of the transfer matrix, see Lemma 3.10.

Corollary 4.2 (Canonical Rayleigh quotient). *Let* $\mathbf{x}$ *be a normalized iTR in canonical form as defined in Definition* 3.8 *and its corresponding canonical transfer matrices as in Definition* 3.9. *Then the Rayleigh quotient associated with the infinite dimensional matrix* $\mathbf{H}$ (1.2) *can be represented by*

$$\theta = \tag{4.5}$$

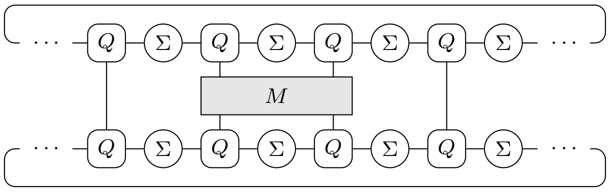


,

where Q *and* Σ *are defined in Definition* 3.8.

Proof. Since $\mathbf{x}$ is in canonical form, the terms in (4.1) are given by

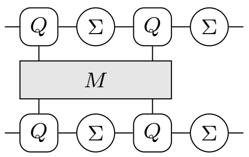
$$\mathbf{x}^\top \mathbf{H}_k \mathbf{x} = \tag{4.6}$$



,

for all k, and in matrix notation as follows

$$\mathbf{x}^\top \mathbf{H}_k \mathbf{x} = \operatorname{Tr}\left[\left(\prod_{\ell=-\infty}^{k-1} T_{Q_R}\right) \tilde{M}_Q \left(\prod_{\ell=k+2}^{+\infty} T_{Q_R}\right)\right], \qquad \tilde{M}_Q :=$$



.

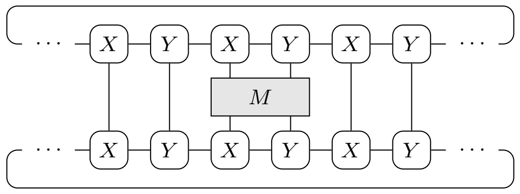
Next, using Lemma 3.6 together with the fact that $\operatorname{vec}(\Sigma^2)$ and $\operatorname{vec}(I)$ are the left and right dominant eigenvectors of T_{Q_R}, respectively, we obtain

$$\mathbf{x}^\top \mathbf{H}_k \mathbf{x} = \operatorname{Tr}\left[\operatorname{vec}(I)\operatorname{vec}(\Sigma^2)^\top \, \tilde{M}_Q \, \operatorname{vec}(I)\operatorname{vec}(\Sigma^2)^\top\right] = \operatorname{vec}(\Sigma^2)^\top \, \tilde{M}_Q \, \operatorname{vec}(I) = \theta,$$

where we consecutively used the trace cyclic property $\mathrm{Tr}(XYZ) = \mathrm{Tr}(YZX)$ and the normalization of the left and right eigenvectors of T_{Q_R}, i.e., $\mathrm{vec}(\Sigma^2)^\top \mathrm{vec}(I) = 1$. ◻

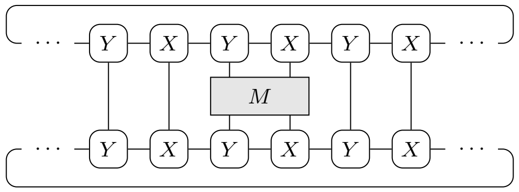
In case $\mathbf{x}$ is an iTR2, the classical definition of the Rayleigh quotient (4.1) now consists of the following even and odd terms

(4.7) $\mathbf{x}^\top \mathbf{H}_{2k}\mathbf{x} =$



,

(4.8) $\mathbf{x}^\top \mathbf{H}_{2k+1}\mathbf{x} =$

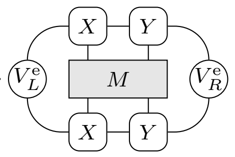


,

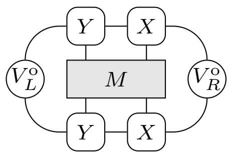
for all k. Hence, we define the iTR2 Rayleigh quotient as the average of 1 even and 1 odd term.

Theorem 4.3 (iTR2 Rayleigh quotient). *Let $\mathbf{x}$ be a normalized nonzero iTR2 as defined in Definition 3.12 and its corresponding transfer matrices be T_{XY} and T_{YX} as in Definition 3.13. Then the Rayleigh quotient for a given infinite dimensional matrix $\mathbf{H}$ (1.2) is defined as*

(4.9) $\theta = \frac{1}{2}$



$+ \frac{1}{2}$



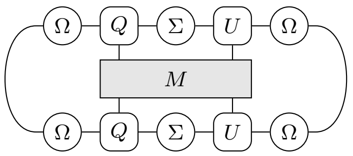
,

where V_L^{e} and V_R^{e} are the matricizations of the left and right dominant eigenvectors of T_{XY}, and V_L^{o} and V_R^{o} are the matricizations of the left and right dominant eigenvectors of T_{YX}, respectively.

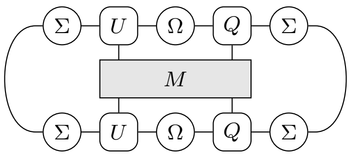
Proof. The proof is similar to the one for Theorem 4.1. ◻

Using the iTR2 canonical form of Definition 3.15, the Rayleigh quotient (4.9) yields

(4.10) $\theta = \frac{1}{2}$



$+ \frac{1}{2}$



,

where Q, U, Σ, and Ω are defined in Definition 3.15.

4.2. Residual. One way to measure the accuracy of an approximate eigenpair $(\theta, \mathbf{x})$ of $\mathbf{H}$ is to evaluate the residual norm. When $\mathbf{H}$ is infinite dimensional and $\mathbf{x}$ an iTR, some care is needed to properly define this residual. By making use of the property that $\mathbf{x}$ can be written as the matrix-vector product of the frame matrix and the canonical center core (3.20), we introduce a local iTR residual involving only 1 core of $\mathbf{x}$.

Because the infinite dimensional $\mathbf{H}$ has an infinite number of terms in the summation (1.2), the standard definition of the residual for a Ritz pair $(\theta, \mathbf{x})$, i.e.,

$$\mathbf{H}\mathbf{x} - \theta\mathbf{x} \tag{4.11}$$

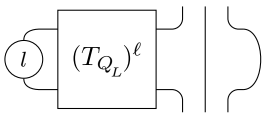
is also infinite. However, by projecting $\mathbf{H}$ into the subspace spanned by the columns of the frame matrix, defined in Definition 3.11, we can define a residual associated with the projected $\mathbf{H}$. Due to the translational invariance of $\mathbf{x}$, we can choose an arbitrary k for the frame matrix. Therefore, without loss of generality, we set $k = 0$ for the remainder of this section.

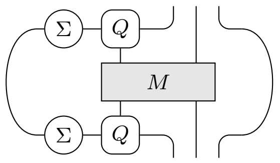
We define the projection of $\mathbf{H}$ onto $\mathbf{F}_{\neq 0}$ as follows

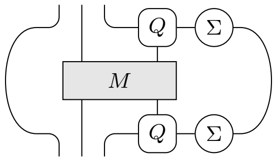
$$H_{\neq 0} := \mathbf{F}_{\neq 0}^{\top}\mathbf{H}\mathbf{F}_{\neq 0} = \sum_{k=-\infty}^{+\infty} \mathbf{F}_{\neq 0}^{\top}\mathbf{H}_k\mathbf{F}_{\neq 0} =: \sum_{k=-\infty}^{+\infty} H_k. \tag{4.12}$$

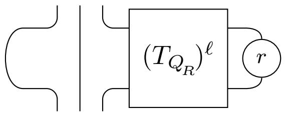
where $\mathbf{H}_k$ is defined in (1.2) and the matrices $H_k \in \mathbb{R}^{dr^2 \times dr^2}$ are obtained by projecting the corresponding $\mathbf{H}_k$ onto $\mathbf{F}_{\neq 0}$. Depending on the relative position of the matrix M in each $\mathbf{H}_k$, with respect to index i_0 in $\mathbf{F}_{\neq 0}$, the matrices H_k have different expressions for each k, i.e.,

(4.13)

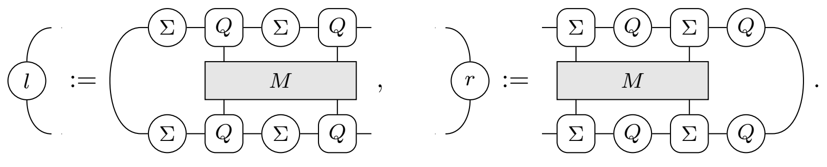
$H_{-2-\ell} =$ , $\ell = 0, 1, \ldots,$

$H_{-1} =$ ,

$H_0 =$ ,

$H_{1+\ell} =$ , $\ell = 0, 1, \ldots,$

with

(4.14) 

Note that, because the transfer matrices T_{Q_L} and T_{Q_R} have a dominant eigenvalue 1, the infinite sums

$$\sum_{\ell=0}^{+\infty}(T_{Q_L})^\ell, \qquad \text{and} \qquad \sum_{\ell=0}^{+\infty}(T_{Q_R})^\ell \tag{4.15}$$

diverge, hence, the infinite sum in (4.12) also diverges.

On the other hand, we can show that for every H_k in (4.13),

$$\mathrm{vec}(Q_c)^\top H_k \,\mathrm{vec}(Q_c) = \theta,$$

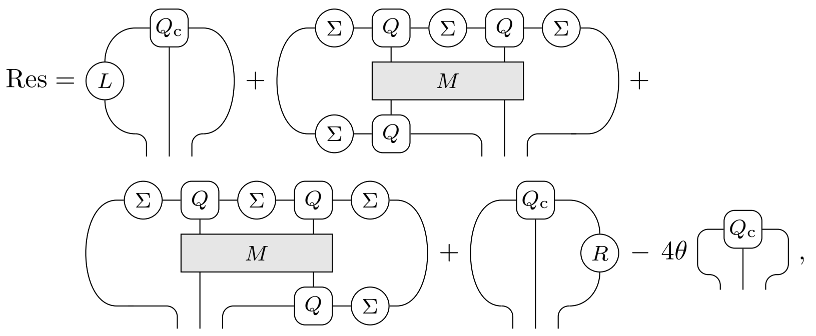
where θ is the the Rayleigh quotient (4.5) and Q_c the canonical center core (3.20). Therefore, we define the iTR residual as follows

$$\mathrm{Res} := \sum_{k=-\infty}^{+\infty} \Big(H_k \,\mathrm{vec}(Q_\mathrm{c}) - \theta\, \mathrm{vec}(Q_\mathrm{c}) \Big) =: \sum_{k=-\infty}^{+\infty} \mathrm{Res}_k \,. \tag{4.16}$$

The following theorem shows how to construct this residual and proves that, in contrast to the infinite sum in (4.12), the iTR residual defined in (4.16) does converge.

THEOREM 4.4 (Residual). *Let* $\mathbf{x}$ *be a normalized nonzero canonical iTR as defined in Definition 3.8. Then the residual for a given infinite dimensional matrix* $\mathbf{H}$ *(1.2) is given by*

(4.17)



where θ *is the Rayleigh quotient* (4.5), Q_c *the canonical center core* (3.20), *and*

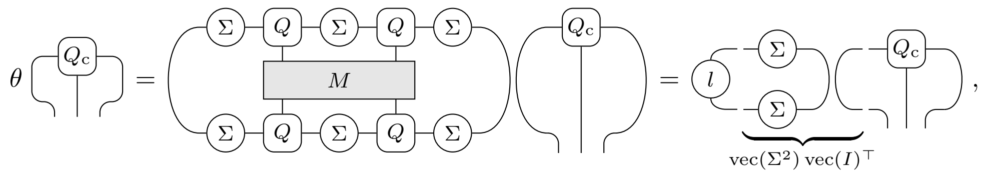
$$\mathrm{vec}(L)^\top := \mathrm{vec}(l)^\top \left[I - \tilde{T}_{Q_L} \right]^{-1}, \qquad \tilde{T}_{Q_L} := T_{Q_L} - \mathrm{vec}(\Sigma^2)\,\mathrm{vec}(I)^\top, \tag{4.18}$$

$$\mathrm{vec}(R) := \left[I - \tilde{T}_{Q_R} \right]^{-1} \mathrm{vec}(r), \qquad \tilde{T}_{Q_R} := T_{Q_R} - \mathrm{vec}(I)\,\mathrm{vec}(\Sigma^2)^\top. \tag{4.19}$$

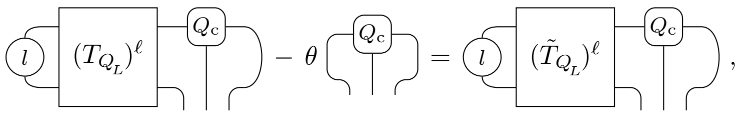
with l *and* r *are defined by the diagrams given in* (4.14).

Proof. Since each term in the infinite sum of (4.16) is defined as the difference $\mathrm{Res}_k = H_k \,\mathrm{vec}(Q_\mathrm{c}) - \theta\, \mathrm{vec}(Q_\mathrm{c})$, we immediately obtain from (4.13) that the 2nd and 3rd term in (4.17) are equal to $H_{-1}\,\mathrm{vec}(Q_\mathrm{c})$ and $H_0\,\mathrm{vec}(Q_\mathrm{c})$, respectively. Next, we will prove that $\sum_{k=-\infty}^{-2} \mathrm{Res}_k$ yields the 1st term in (4.17) minus $\theta\,\mathrm{vec}(Q_\mathrm{c})$, and $\sum_{k=1}^{+\infty} \mathrm{Res}_k$ yields the 4th term minus $\theta\,\mathrm{vec}(Q_\mathrm{c})$. We show that both infinite sums converge.

By using (4.5) and (4.14), we diagrammatically rewrite $\theta\,\mathrm{vec}(Q_c)$ in the following factored form



yielding

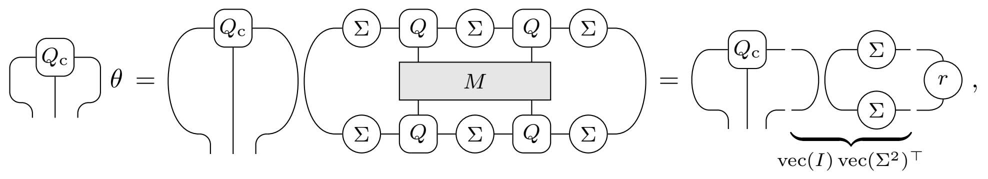
(4.20) $\mathrm{Res}_{-2-\ell} =$ 

for $\ell = 1, 2, \ldots$. Because $\mathrm{vec}(I)$ and $\mathrm{vec}(\Sigma^2)$ are the left and right eigenvectors associated with the dominant eigenvalue 1 of T_{Q_L}, as shown in Lemma 3.10, the largest eigenvalue of the deflated matrix $\tilde{T}_{Q_L}$ is less than 1 (in magnitude). Hence, in contrast to (4.15), the geometric series of $\tilde{T}_{Q_L}$ converges, yielding

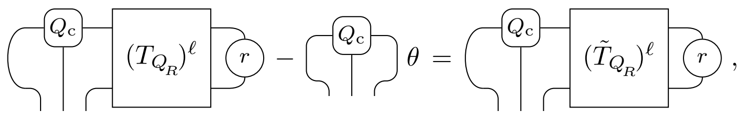
$$\sum_{\ell=0}^{+\infty} (\tilde{T}_{Q_L})^\ell = \left[I - \tilde{T}_{Q_L}\right]^{-1}.$$

Note that this sum starts from 0 and (4.20) starts from 1. Consequently, the infinite sum $\sum_{k=-\infty}^{-2} \mathrm{Res}_k$ yields the 1st term in (4.17) minus $\theta\,\mathrm{vec}(Q_c)$, where L is defined by (4.18).

For the remaining part of the proof, we rewrite $\mathrm{vec}(Q_c)\theta$ as


Q_c
θ =
Q_c
Σ Q Σ Q Σ
M
Σ Q Σ Q Σ
=
Q_c
Σ
r
Σ
,
$\mathrm{vec}(I)\,\mathrm{vec}(\Sigma^2)^\top$

yielding

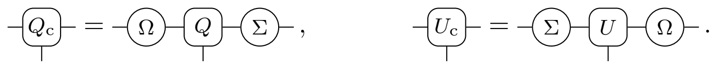
(4.21) $\mathrm{Res}_{1+\ell} =$ 

for $\ell = 1, 2, \ldots$. Using similar arguments as before, we can show that the geometric series of $\tilde{T}_{Q_R}$ converges and that the infinite sum $\sum_{k=1}^{+\infty} \mathrm{Res}_k$ yields the 4th term in (4.17) minus $\theta\,\mathrm{vec}(Q_c)$. Remark that the term $\theta\,\mathrm{vec}(Q_c)$ in Res_k always gets incorporated in $\tilde{T}_{Q_L}$ or $\tilde{T}_{Q_R}$, except for $k = -2, -1, 0, 1$, yielding the 5th term in (4.17). This completes the proof. ∎

In order to define the residual for a Ritz pair $(\theta, \mathbf{x})$ with $\mathbf{x}$ being an iTR2, we first introduce the frame matrices $\mathbf{F}_{\neq Q}$ and $\mathbf{F}_{\neq U}$ which allow us to rewrite $\mathbf{x}$ as follows

$$\mathbf{x} = \mathbf{F}_{\neq Q}\,\mathrm{vec}(Q_c) = \mathbf{F}_{\neq U}\,\mathrm{vec}(U_c), \tag{4.22}$$

where $\mathbf{F}_{\neq Q}^\top \mathbf{F}_{\neq Q} = \mathbf{F}_{\neq U}^\top \mathbf{F}_{\neq U} = I$ and the *canonical center cores* Q_c and U_c are

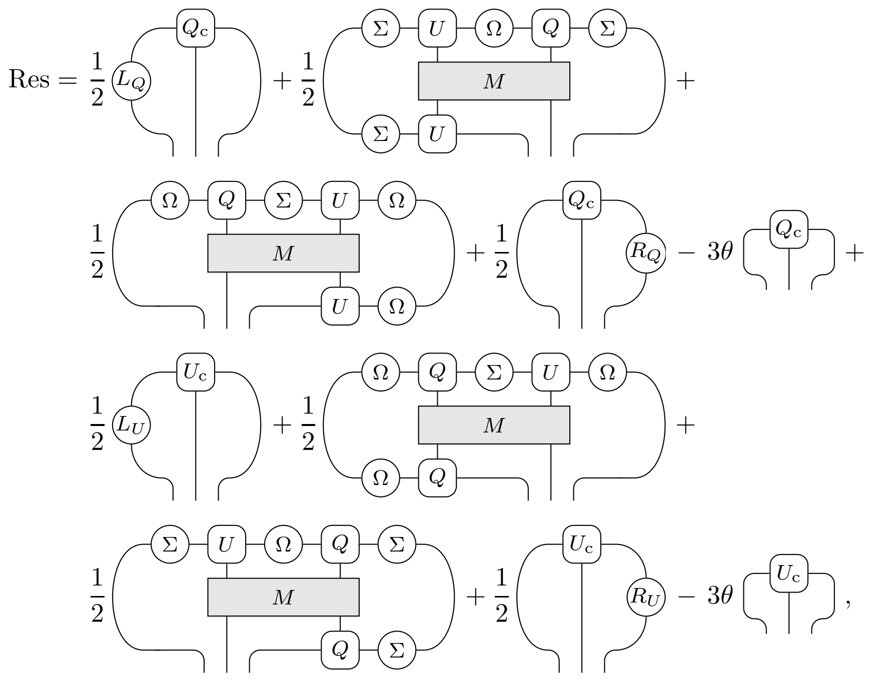
(4.23) 
Q_c = Ω Q Σ ,
U_c = Σ U Ω .

respectively. Next, we define the residual in an average sense as follows

$$\begin{aligned}\mathrm{Res} = \frac{1}{2}\sum_{k=-\infty}^{+\infty} \Big(\mathbf{F}_{\neq Q}^\top \mathbf{H}_k \mathbf{F}_{\neq Q}\,\mathrm{vec}(Q_c) - \theta\,\mathrm{vec}(Q_c)\Big) + \\ \frac{1}{2}\sum_{k=-\infty}^{+\infty} \Big(\mathbf{F}_{\neq U}^\top \mathbf{H}_k \mathbf{F}_{\neq U}\,\mathrm{vec}(U_c) - \theta\,\mathrm{vec}(U_c)\Big),\end{aligned}$$

where θ is the Rayleigh quotient (4.10), and Q_c and U_c the canonical center cores (4.23). The following theorem summarizes how to compute this residual.

Theorem 4.5 (iTR2 residual). *Let* $\mathbf{x}$ *be a normalized nonzero canonical iTR2 as defined in Definition 3.15. Then the residual for a given infinite dimensional matrix* $\mathbf{H}$ (1.2) *is given by*

(4.24)


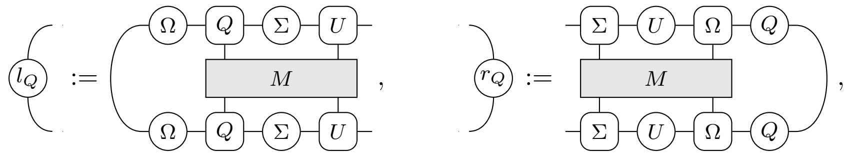
where θ *is the Rayleigh quotient* (4.10), Q_c *and* U_c *the canonical center cores* (4.23),

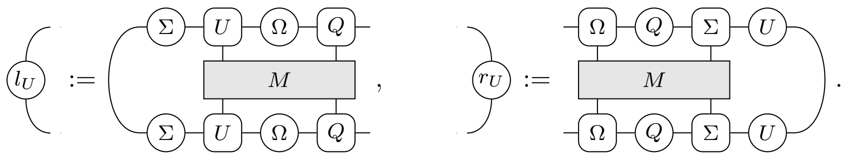
$$
\begin{aligned}
\operatorname{vec}(L_Q)^\top &:= \left[\operatorname{vec}(l_Q)^\top + \operatorname{vec}(l_U)^\top T_{U_L}\right]\left[I - \tilde{T}_{Q_L U_L}\right]^{-1}, \\
\operatorname{vec}(L_U)^\top &:= \left[\operatorname{vec}(l_U)^\top + \operatorname{vec}(l_Q)^\top T_{Q_L}\right]\left[I - \tilde{T}_{U_L Q_L}\right]^{-1}, \\
\operatorname{vec}(R_Q) &:= \left[I - \tilde{T}_{U_R Q_R}\right]^{-1}\left[\operatorname{vec}(r_U) + T_{U_R}\operatorname{vec}(r_Q)\right], \\
\operatorname{vec}(R_U) &:= \left[I - \tilde{T}_{Q_R U_R}\right]^{-1}\left[\operatorname{vec}(r_Q) + T_{Q_R}\operatorname{vec}(r_U)\right],
\end{aligned} \tag{4.25}
$$

with

$$
\begin{aligned}
\tilde{T}_{Q_L U_L} &:= T_{Q_L U_L} - \operatorname{vec}(\Omega^2)\operatorname{vec}(I)^\top, & \tilde{T}_{Q_R U_R} &:= T_{Q_R U_R} - \operatorname{vec}(I)\operatorname{vec}(\Omega^2)^\top, \\
\tilde{T}_{U_L Q_L} &:= T_{U_L Q_L} - \operatorname{vec}(\Sigma^2)\operatorname{vec}(I)^\top, & \tilde{T}_{U_R Q_R} &:= T_{U_R Q_R} - \operatorname{vec}(I)\operatorname{vec}(\Sigma^2)^\top,
\end{aligned}
$$

and



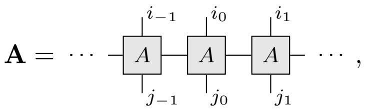


Proof. The proof is similar to the one for Theorem 4.4. ∎

5. Flexible power method.

5. Flexible power method. One way to compute the algebraically smallest eigenvalue λ_1 of $\mathbf{H}$ is to apply the power method to the matrix exponential $e^{-\mathbf{H}}$. If λ_1 is simple, the power method converges linearly to the desired eigenpair $(\lambda_1, \mathbf{x})$ at the rate $e^{\lambda_2-\lambda_1}$. This approach is generally not recommended because working with the matrix exponential $e^{-\mathbf{H}}$ can be costly. In fact, computing $e^{-\mathbf{H}}$ or applying $e^{-\mathbf{H}}$ to a vector may be harder than computing eigenvalues of $\mathbf{H}$. However, when $\mathbf{H}$ has the structure exhibited in (1.2), $e^{-\mathbf{H}t}$ may be approximated by a simpler form that makes it possible to perform a power iteration to $e^{-\mathbf{H}t}$ for a small t.

Throughout this section, we will make use of iTR matrices which are graphically represented as follows



where all indices i_k, j_k run from 1 to d and each $A(j,i)$ is a matrix of size $r \times r$, with r referred to as the *rank* of $\mathbf{A}$.

5.1. Approximating exp(−Ht) by matrix splitting.

5.1. Approximating $\exp(-\mathbf{H}t)$ by matrix splitting. We now discuss how to approximate $e^{-\mathbf{H}t}\mathbf{x}$, with $\mathbf{x}$ being an iTR, required in a single step of the power iteration. Because the infinite matrix $\mathbf{H}$ consists of an infinite sum, it is tempting to rewrite $e^{-\mathbf{H}} = e^{-\sum_k \mathbf{H}_k}$ as $\cdots e^{-\mathbf{H}_{k-1}} e^{-\mathbf{H}_k} e^{-\mathbf{H}_{k+1}} \cdots$. However, this is not valid because the $\mathbf{H}_k$'s do not commute in general. On the other hand, we can use the Lie product formula, also known as Suzuki–Trotter splitting,

$$e^{(A+B)t} = \left[e^{At}e^{Bt}\right] + \mathcal{O}(t^2),$$

where A and B are square matrices, yielding the following approximation

$$e^{-\mathbf{H}t} \approx \prod_{k=-\infty}^{+\infty} e^{-\mathbf{H}_k t}, \tag{5.1}$$

which may be accurate enough if t is sufficiently small. In the physics literature, t is viewed as an imaginary time variable, hence, $e^{-\mathbf{H}t}$ is often referred to as imaginary time evolution.

First, assume the matrices $\mathbf{H}_k$ to be completely local, i.e.,

$$\mathbf{H}_k = \cdots \otimes I_d \otimes I_d \otimes A_k \otimes I_d \otimes I_d \otimes \cdots, \qquad A_k = A \in \mathbb{R}^{d\times d},$$

so that, by making use of the identities

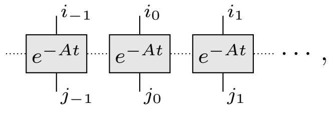
$$e^{A\otimes I} = e^A \otimes I, \qquad \text{and} \qquad e^{I\otimes A} = I \otimes e^A,$$

the matrix exponentials $e^{-\mathbf{H}_k t}$ can be simplified as follows

$$e^{-\mathbf{H}_k t} = \cdots \otimes I \otimes I \otimes e^{-At} \otimes I \otimes I \otimes \cdots,$$

and the infinite product in (5.1) corresponds to the rank-1 iTR matrix

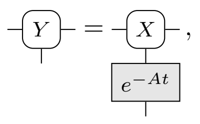
$$\prod_{k=-\infty}^{+\infty} e^{-\mathbf{H}_k t} = \cdots \otimes e^{-At} \otimes e^{-At} \otimes e^{-At} \otimes \cdots = $$



where the dotted lines correspond to an iTR matrix rank $r = 1$. Hence, if $\mathbf{x}$ is an iTR, the product

$$\mathbf{y} = \left(\prod_{k=-\infty}^{+\infty} e^{-\mathbf{H}_k t} \right) \mathbf{x}, \tag{5.2}$$

is again an iTR of the same rank, and (5.2) can efficiently be computed by left multiplying the slices of $\mathbf{x}$ with the $d \times d$ matrix e^{-At}



where X and Y are the cores of $\mathbf{x}$ and $\mathbf{y}$, respectively.

If $\mathbf{H}_k$ represent nearest neighbor interactions, i.e.,

$$\mathbf{H}_k = \cdots \otimes I_d \otimes I_d \otimes M_{k,k+1} \otimes I_d \otimes I_d \otimes \cdots, \qquad M_{k,k+1} = M \in \mathbb{R}^{d^2 \times d^2}, \tag{5.3}$$

so that each $\mathbf{H}_k$ will interact with the kth and $(k+1)$st cores of $\mathbf{x}$. Because $\mathbf{H}_k$ and $\mathbf{H}_{k+1}$ both interact with the $(k+1)$st core of $\mathbf{x}$, $e^{-\mathbf{H}t}\mathbf{x}$ cannot be directly obtained as in (5.2). Therefore, we first reorder the summation and group the even and odd terms together

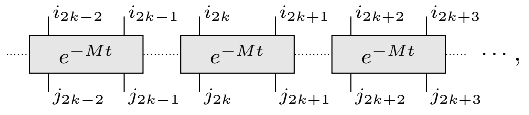
$$\mathbf{H} = \left(\sum_{k=-\infty}^{+\infty} \mathbf{H}_{2k} \right) + \left(\sum_{k=-\infty}^{+\infty} \mathbf{H}_{2k+1} \right) =: \mathbf{H}_\mathrm{e} + \mathbf{H}_\mathrm{o}.$$

Using the Suzuki–Trotter splitting twice, yields

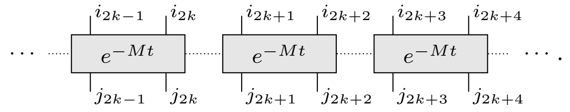
$$e^{-\mathbf{H}t} = e^{-(\mathbf{H}_\mathrm{e}+\mathbf{H}_\mathrm{o})t} \approx \left(\prod_{k=-\infty}^{+\infty} e^{-\mathbf{H}_{2k} t} \right) \left(\prod_{k=-\infty}^{+\infty} e^{-\mathbf{H}_{2k+1} t} \right), \tag{5.4}$$

for sufficiently small t, where

$$\prod_{k=-\infty}^{+\infty} e^{-\mathbf{H}_{2k} t} = \cdots \tag{5.5}$$

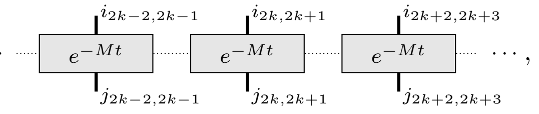


$$\prod_{k=-\infty}^{+\infty} e^{-\mathbf{H}_{2k+1} t} = \tag{5.6}$$

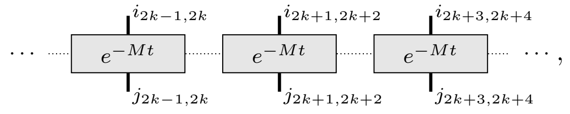


Remark that both (5.5) and (5.6) can also be considered as local operators when we combine the corresponding neighboring indices. Hence, both the even and odd infinite products in (5.4) result again in rank-1 iTR matrices

(5.7) $$\prod_{k=-\infty}^{+\infty} e^{-\mathbf{H}_{2k}t} =$$



(5.8) $$\prod_{k=-\infty}^{+\infty} e^{-\mathbf{H}_{2k+1}t} =$$

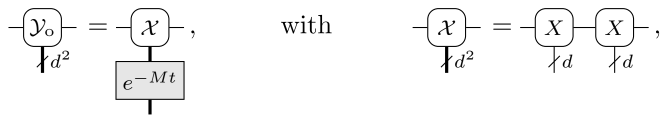


where the neighboring indices $i_{2k,2k+1} = (i_{2k}, i_{2k+1})$ and $i_{2k+1,2k+2} = (i_{2k+1}, i_{2k+2})$ are combined, respectively.

Then, in order to approximate $e^{-\mathbf{H}t}\mathbf{x}$, we first focus on the odd terms, i.e.,

(5.9) $$\mathbf{y}_\text{o} = \left(\prod_{k=-\infty}^{+\infty} e^{-\mathbf{H}_{2k+1}t} \right) \mathbf{x},$$

which can be computed in the same way as (5.2) by reformulating $\mathbf{x}$ as an iTR2 with 2 cores X, i.e., combining the $(2k+1)$st and $(2k+2)$nd cores into a *supercores* $\mathcal{X}$ of dimension $r \times d^2 \times r$. Hence, (5.9) only requires the left multiplication of the slices of the new supercore with the $d^2 \times d^2$ matrix e^{-Mt}



where d and d^2 correspond here to the dimensions of the (super)cores. Thereafter, we only need to multiply the even terms, however, directly applying

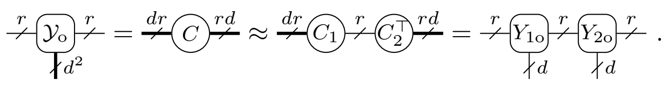
(5.10) $$\mathbf{y} = \left(\prod_{k=-\infty}^{+\infty} e^{-\mathbf{H}_{2k}t} \right) \mathbf{y}_\text{o}$$

is not possible because the supercores $\mathcal{Y}_\text{o}$ are formed by combining the indices $(2k+1, 2k+2)$, while (5.7) requires the indices $(2k, 2k+1)$ to be combined. Therefore, we first need to split $\mathbf{y}_\text{o}$ back into 2 normal cores, followed by a new and different recombination of the cores in order to carry out (5.10) efficiently.

The dimension of $\mathcal{Y}_\text{o}$ is $r \times d^2 \times r$. However, there is no guarantee that this supercore can be split into 2 cores, of dimension $r \times d \times r$, without loosing information. A possible way to split is by first reshaping $\mathcal{Y}_\text{o}$ into an $rd \times rd$ matrix C, next approximating C by a rank-r factorization

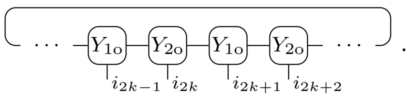
$$C \approx C_1 C_2^\top,$$

where $C_1, C_2 \in \mathbb{R}^{dr \times r}$, and finally reshaping the factors C_1 and C_2 into 3rd-order tensors

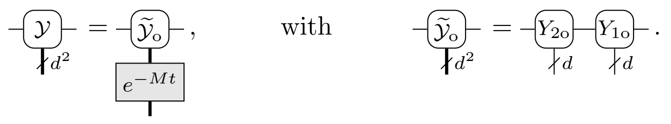


In order words, $\mathbf{y}_\text{o}$ is approximated by

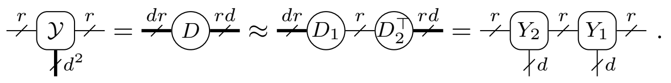
$$\tilde{\mathbf{y}}_\text{o} =$$



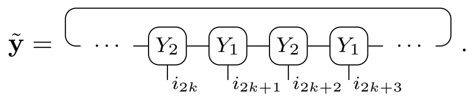
Consequently, we can now approximate (5.10) by replacing $\mathbf{y}_\text{o}$ with $\tilde{\mathbf{y}}_\text{o}$



In the final step, we split the supercore $\mathcal{Y}$ into 2 cores Y_2 and Y_1



or in order words, $\mathbf{y}$ is approximated by



Remark that, although we started with an iTR $\mathbf{x}$, the result of applying $e^{-\mathbf{H}t}$ to it is an iTR2. Therefore, in Figure 3 we graphically illustrate the application of (5.1) to an iTR2 with cores X_1 and X_2. Due to the translational invariance, we only need to apply e^{-Mt} twice in order to compute $\tilde{\mathbf{y}}$, which is indicated in black. The computation indicated in grey can completely be ignored, so that the multiplication $\mathbf{y} = e^{-\mathbf{H}t}\mathbf{x}$ can be computed in an efficient way.

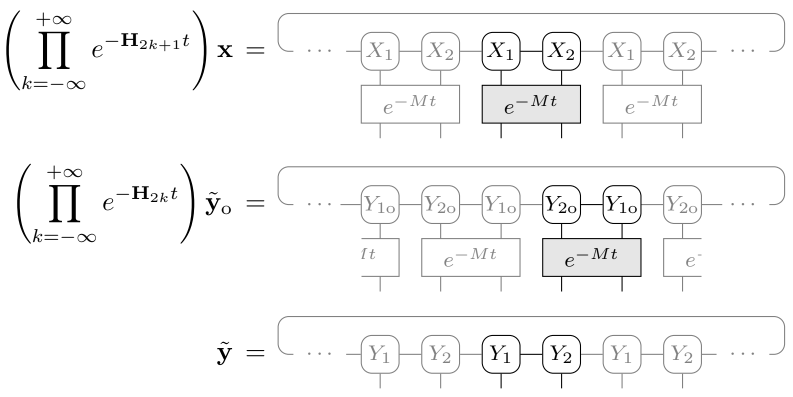
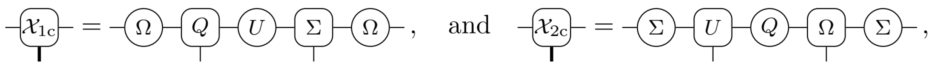


FIG. 3. *The matrix–vector operation* $\mathbf{y} = e^{-\mathbf{H}t}\mathbf{x}$.

5.2. Algorithm. As illustrated in Figure 3, we only need to operate on 1 supercore at the time. Instead of working with supercores involving X_1, X_2 and $Y_{2\text{o}}, Y_{1\text{o}}$, respectively, we will operate on the following canonical center supercores $\mathcal{X}_{1\text{c}}$ and $\mathcal{X}_{2\text{c}}$



respectively, so that their corresponding frame matrices are orthogonal.

We only describe the application of the odd part of $e^{-\mathbf{H}t}$ onto $\mathcal{X}_\text{c}$, since the even part can be performed in exactly the same way by interchanging Q, U and Σ, Ω, respectively. The computation consists of 5 steps as illustrated in Figure 4. In step 1, the canonical center supercore $\mathcal{X}_{1\text{c}}$ is formed. Next, we apply e^{-Mt} to it in step 2, followed by reshaping the resulting supercore $\mathcal{Y}_{1c}$ into a matrix. In step 3, we approximate the matrix representation of $\mathcal{Y}_{1c}$ by a truncated SVD, with the diagonal matrix S_r containing only the r largest singular values, followed by reshaping the left and right singular vectors W and V into 3rd-order tensors, i.e., cores. Then in step 4, in order to update the cores Q and U by W_Ω and $V_\Omega^\top$, respectively, and the diagonal matrix Σ by S_r, we transform the result of step 3 into the form of step 1 by left and right multiplying W and $V^\top$, respectively, by Ω^{-1}. Note that, although the resulting form resembles the form of step 1, the former is not in canonical form yet. Therefore, we update in step 5 both cores $Q_\star$ and $U_\star$, and both diagonal matrices $\Sigma_\star$ and $\Omega_\star$ to bring the updated iTR2 again in canonical form.

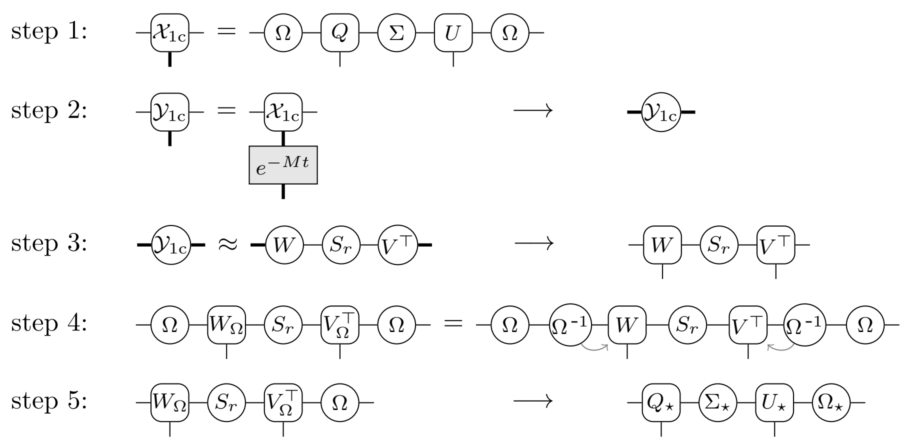


Fig. 4. The odd part of the matrix–vector operation $\mathbf{y} = e^{-\mathbf{H}t}\mathbf{x}$, where $\mathbf{x}$ is given in canonical iTR2 form. Note that the even part corresponds to interchanging the cores Q and U, and the diagonal matrices Σ and Ω, respectively. (1) Forming the canonical center supercore. (2) Applying the matrix exponential followed by reshaping the result into a matrix. (3) Performing a truncated SVD and reshaping the singular vectors into cores. (4) Updating the cores and the central diagonal matrix, while keeping Ω unchanged. (5) Computing the canonical iTR2 form.

We now have all components to state the method for computing the smallest eigenvalue of (1.2). Due to the approximation made by the Suzuki–Trotter splitting of $e^{-\mathbf{H}t}$ and the rank truncation for splitting a supercore into 2 normal cores, the multiplication $\mathbf{y} = e^{-\mathbf{H}t}\mathbf{x}$ is inexact. Even when t is fixed, the error introduced by the rank truncation is different in each iteration and can be viewed as every time applying a slightly different operator. Therefore, we call the algorithm for computing the smallest eigenvalue of $\mathbf{H}$, based on a power method of $e^{-\mathbf{H}}$, the *flexible* power method, similar to the flexible GMRES algorithm used for computing solutions of linear systems.

Algorithm 3 shows the outline of the flexible power method for computing the smallest eigenvalue of (1.2). This algorithm takes a canonical iTR2 as starting vector and returns the Rayleigh quotient and its corresponding eigenvector in canonical iTR2 form. In every iteration of Algorithm 3, we apply the matrix exponential e^{-Mt} twice, according to Figure 4, followed by the computation of the Rayleigh quotient and residual.

As we will show in the numerical experiments, the timestep t can have a significant effect on the convergence and accuracy of the Rayleigh quotient. Since Algorithm 3 applies approximate powers of the matrix exponential $e^{-\mathbf{H}}$, choosing a smaller timestep

Algorithm 3: Flexible power method

```
Input  : Canonical iTR2 with cores $Q, U \in \mathbb{R}^{r \times d \times r}$ and matrices
         $\Sigma, \Omega \in \mathbb{R}^{r \times r}$
         Nearest neighbor interaction matrix $M$ and initial timestep $t$
Output: Rayleigh quotient $\theta$ and corresponding eigenvector in canonical
         iTR2
while not converged do
1   Apply $e^{-Mt}$ to supercore $\mathcal{X}_{1c}$ according to Figure 4.
2   Apply $e^{-Mt}$ to supercore $\mathcal{X}_{2c}$ according to Figure 4.
3   Compute Rayleigh quotient $\theta$ as in (4.10).
4   Check convergence based on residual (4.24).
5   Update $t$.
end
```

will require more iterations for applying the same power. However, if the timestep is chosen too large, the convergence of Algorithm 3 stagnates due to the dominating Suzuki–Trotter splitting error. It is therefore recommended to adapt and decrease the timestep t in the course of the algorithm. By starting with a relative large t, the power method will stagnate in a small number of iterations. We can use the iTR2 residual for detecting stagnation. Next, we decrease t and continue the power method with the already obtained approximate eigenvector. By updating t and using more accurate approximate eigenvectors for smaller t, we will be able to reduce the total number of iterations.

5.3. Computational considerations. Most operations of Algorithm 3 can be efficiently implemented involving only 3rd-order tensors of size $r \times d \times r$, such as steps 1–4 in Figure 4 and the Rayleigh quotient. On the other hand, step 5 in Figure 4 and the computation of the residual involve transfer matrices of size $r^2 \times r^2$. The former requires computing dominant eigenvectors of the left and right transfer matrices and the latter needs linear systems solves involving the same matrices. In both cases, iterative methods can be used, however, even for low values of $r \approx 10$, more than 90% of the wall clock time is spend on these operations involving the transfer matrices. We could choose not to compute the residual in every iteration but the cost for transforming the updated iTR2 to canonical form remains.

Therefore, we also propose a second variant of the flexible power method that only works with standard iTR2s, i.e., ignoring step 5 in Figure 4 such that the updated cores $Q_\star := W_\Omega$ and $U_\star := V_\Omega^\top$, and the updated diagonal matrices $\Sigma_\star := S_r$ and $\Omega_\star := \Omega$. Note that in this case only 1 of the 2 diagonal matrices is updated, hence the resulting iTR2 will, in general, not satisfy all the orthogonality conditions of Definition 3.15. However, due to the alternating singular value decompositions on $\mathcal{Y}_{1c}$ and $\mathcal{Y}_{2c}$, we observe that the resulting iTR2 converges to canonical form. As a result, we can still use the relatively cheap canonical Rayleigh quotient (4.10) compared to the standard Rayleigh quotient (4.9) which would again require the computation of dominant eigenvectors of the transfer matrices.

Note that in the iTEBD algorithm described in [15, 9] the matrix-vector operation $\mathbf{y} = e^{-\mathbf{H}t}\mathbf{x}$ is implemented in a slightly different way. Instead of bringing the updated iTR2 to canonical form, as done in step 5 in Figure 4, the iTR2 is directly transformed to canonical form right after step 2 and before the truncation is applied in step 3. Hence, the resulting iTR2 is only approximately in canonical form.

6. Numerical experiments.

In this section we illustrate the flexible power method, outlined in Algorithm 3, for 3 examples: the one-dimensional Ising model with a transverse field, the spin $S = 1$ Heisenberg isotropic antiferromagnetic model, and the spin $S = 1/2$ Heisenberg model.

All numerical experiments are performed in MATLAB version 9.6.0 (R2019a) on a MacBook Pro running an Intel(R) Core(TM) i7-5557U CPU @ 3.1 GHz dual core processor with 16 GB RAM. The dominant eigenvectors of the transfer matrices are computed via `eigs` and the linear systems for computing the residual in (4.25) are solved via `gmres` with tolerance 1e−8.

6.1. One-dimensional transverse field Ising model.

We start with the one-dimensional transverse field Ising model that has the corresponding infinite dimensional Hamiltonian

$$\mathbf{H} = -\sum_{k=-\infty}^{+\infty} \cdots \otimes I \otimes \sigma_z \otimes \sigma_z \otimes I \otimes \cdots - g \sum_{k=-\infty}^{+\infty} \cdots \otimes I \otimes \sigma_x \otimes I \otimes \cdots \tag{6.1}$$

where σ_x and σ_z are the 1st and 3rd Pauli matrices, respectively, and the following 4×4 symmetric matrix

$$M = \begin{bmatrix} -1 & -g & 0 & 0 \\ -g & 1 & 0 & 0 \\ 0 & 0 & 1 & -g \\ 0 & 0 & -g & -1 \end{bmatrix}$$

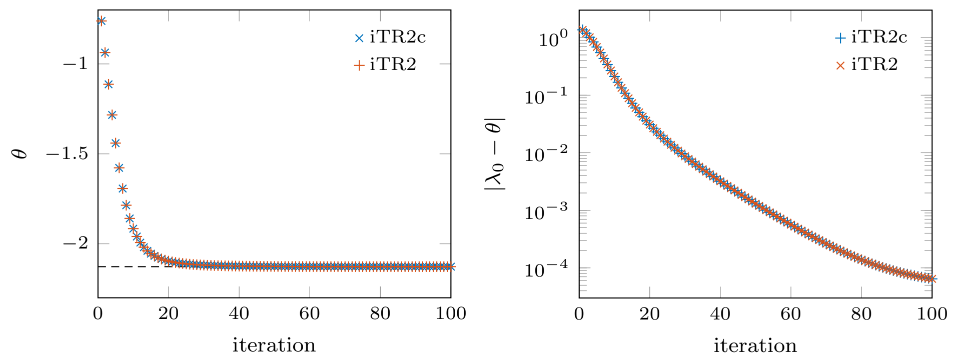
as nearest neighbor interaction matrix in the notation of (1.2). We choose the parameter $g = 2$. The exact solution of the smallest eigenvalue of (6.1) is [13]

$$\lambda_0(g) = -\frac{1}{2\pi}\int_{-\pi}^{\pi} \sqrt{1 + g^2 - 2g\cos x}\, dx, \qquad \lambda_0(2) \approx -2.127\,088\,819\,946\,730.$$

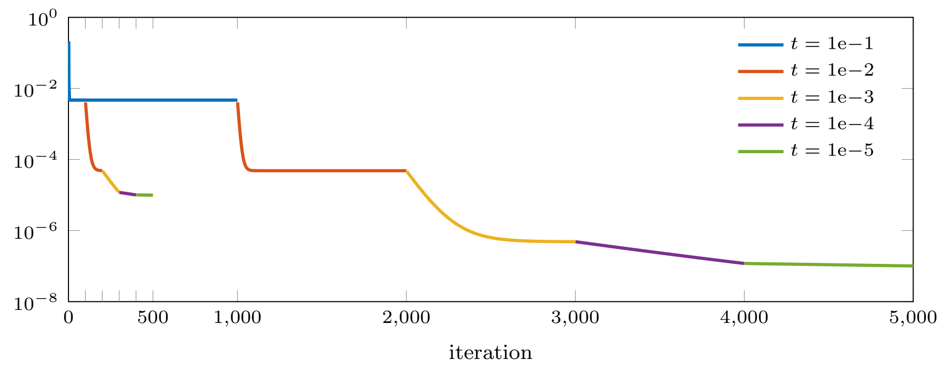
In the first experiment, we compare the 2 variants of Algorithm 3, i.e., the canoncial variant where we always convert the result of applying e^{-Mt} back into canonical iTR2 form and the non-canonical variant where we skip step 5 in Figure 4. In order to improve the numerical stability in the latter one, we normalize the diagonal matrices to have unit Frobenius norm. Figure 5a shows the Rayleigh quotient θ and its difference with the exact solution as a function of the iteration count. Note that the results for both variants of Algorithm 3 are indistinguishable, hence, we will only use the faster non-canonical variant in the remainder of this section.

Next, we analyze the effect of the timestep t on the convergence of Algorithm 3. Figure 5b depicts the convergence of 2 different runs where we respectively decreased the timestep t after every 100 or 1000 iterations. Note that different timesteps are indicated by different colors. From this figure it is clear that the timestep control can have a significant effect on the overall attained accuracy. When t is decreased too soon, as in the first run, the flexible power method tend to stagnate and further decreasing t does not help. On the other hand, we do not want to run too many iteration and therefore we will use the iTR residual, introduced in section 4.2, to detect stagnation.

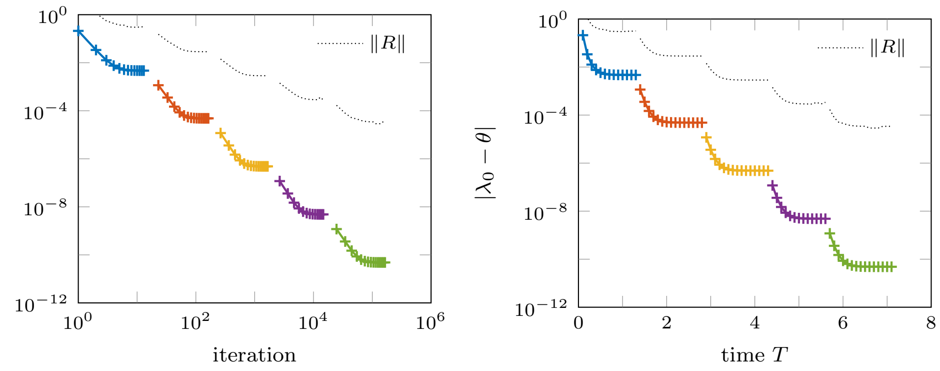
As discussed in section 5.2, the total number of iterations can be reduced by gradually decreasing the timestep t. Table 1 shows that running Algorithm 3 with



(a) Comparison of the canonical and non-canonical variants of Algorithm 3 for timestep $t = 0.01$.



(b) Influence of the timestep t on the convergence of Algorithm 3.



(c) Convergence as a function of iteration count and total time.

FIG. 5. *One-dimensional transverse field Ising model with $g = 2$ and iTR rank $r = 10$.*

a fixed timestep t = 1e−5 requires at least twice the number of iterations and wall clock time than the adaptive timestep approach, where t was divided by 10 when the first 3 digits of the residual norm $\|\text{Res}\|$ were not changing anymore in 3 consecutive convergence checks or in case the residual increased. The left plot in Figure 5c shows the corresponding convergence as a function of the iteration count. Note that by using the automatic timestep adaption approach, the accuracy of the Ritz value is multiple orders of magnitude higher than achieved in both runs shown in Figure 5b.

TABLE 1
Wall clock times for one-dimensional transverse field Ising model with $g = 2$ and iTR rank $r = 10$.

t	iter	time [s]	iter	time [s]
1e−1			13	0.87
1e−2			150	0.72
1e−3			1,500	1.40
1e−4			13,000	4.80
1e−5	370,000	122.52	150,000	44.87
Total	370,000	122.52	164,663	52.67

The right plot in Figure 5c shows the same results as the left one but in function of the total time $T = Nt$, with N the number of iterations of Algorithm 3 for each t. Remark that the stagnation happens for every timestep more or less at the same total time T, corresponding to roughly speaking the same power of $e^{-\mathbf{H}}$. We will use this observation in the next numerical experiments to change the frequency of the convergence check based on the current timestep.

6.2. Spin $S = 1$ Heisenberg isotropic antiferromagnetic model. As a second example, we consider the isotropic antiferromagnetic case of the spin $S = 1$ Heisenberg model. The resulting Hamiltonian has the form of (1.2), with the following nearest neighbor interaction matrix

$$M_{k,k+1} = X_k \otimes X_{k+1} + Y_k \otimes Y_{k+1} + \Delta Z_k \otimes Z_{k+1},$$

where $\Delta = 1$, and X_k, Y_k, and Z_k the spin-1 generators of SU(2)

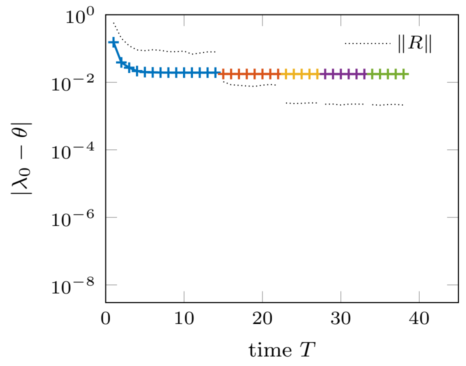
$$X_k = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \qquad Y_k = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & -\imath & 0 \\ \imath & 0 & -\imath \\ 0 & \imath & 0 \end{bmatrix}, \qquad Z_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}.$$

The smallest eigenvalue is $\lambda_0 \approx -1.401\,484\,038\,971\,2(2)$ [5].

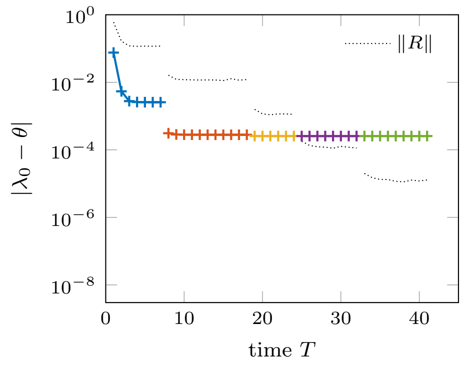
Figures 6a to 6d show the convergence of Algorithm 3 for different iTR ranks $r = 10$, 20, 30, and 60, respectively. For every case, we started with a timestep $t = 0.1$ and the time step is divided by a factor of 10 whenever residual stagnation is detected with the criteria stated in the previous example. The frequency of the convergence check was set to $1/t$ in order to avoid computing to many (expensive) residual calculations. The corresponding wall clock time for the different runs are given in Table 2. This table illustrates again that the smaller the timestep t the more iterations are required, hence, the larger the wall clock time.

When the rank of the iTR is too small, we see from Figure 6 that the Rayleigh quotient does not further improve even when the timestep is reduced because the SVD truncation error is too large. On the other hand, we observe from Figures 6b to 6d that the residual norms in all these runs behave in a similar fashion. Since the iTR2 residual (4.24) is a projected residual, we cannot use it as the only metric for deciding when to terminate the flexible power iteration. As illustrated in Figure 6e, the singular values given by the diagonal matrices of Definition 3.15 and in particular the smallest singular value together with the residual are good indicators for the actual accuracy of the approximate eigenpair of $\mathbf{H}$ because the overall accuracy is bounded by both the Suzuki–Trotter splitting and SVD truncation error.

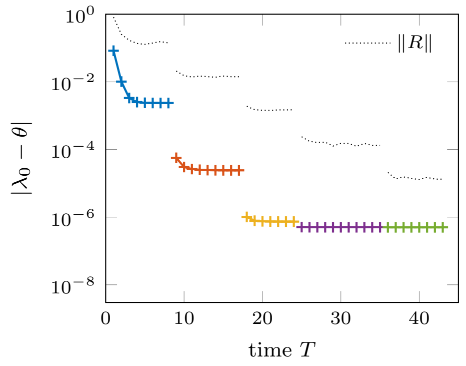
6.3. Spin $S = 1/2$ Heisenberg model. As a last example, we consider the spin $S = 1/2$ Heisenberg model. The resulting Hamiltonian has the form of (1.2),



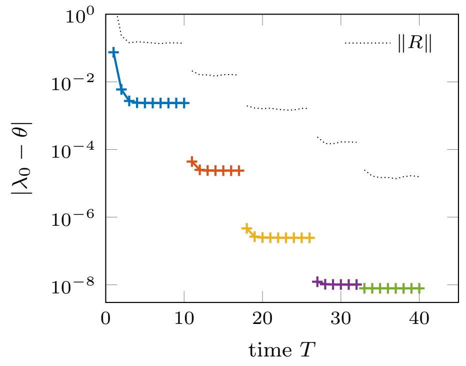
(a) $r = 10$



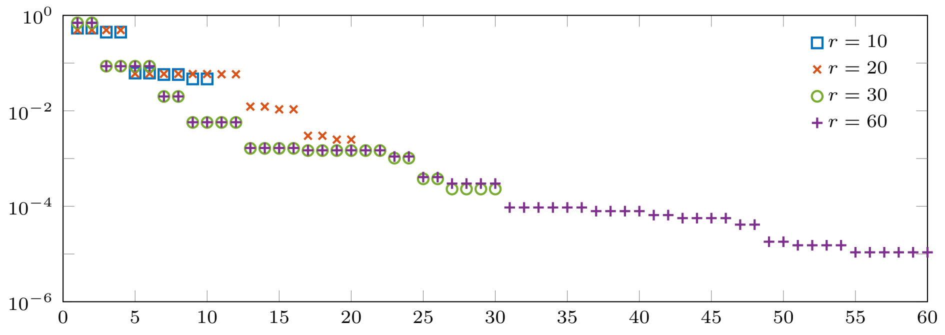
(b) $r = 20$



(c) $r = 30$



(d) $r = 60$



(e) Singular values

FIG. 6. *Spin $S = 1$ Heisenberg isotropic antiferromagnetic model.*

with the following nearest neighbor interaction matrix

$$M_{k,k+1} = X_k \otimes X_{k+1} + Y_k \otimes Y_{k+1} + Z_k \otimes Z_{k+1},$$

where $X_k = \frac{1}{2}\sigma_x$, $Y_k = \frac{1}{2}\sigma_y$, $Z_k = \frac{1}{2}\sigma_z$, with σ_x, σ_y, σ_z the Pauli matrices. The exact solution of the smallest eigenvalue is $\lambda_0 = -\ln(2) + \frac{1}{4} \approx -0.443\,147\,180\,559\,945$.

In Figure 7a, we plot the convergence history of the Rayleigh quotient θ to the exact eigenvalue λ_0 as well as the changes in the first (θ_1) and second (θ_2) terms of

Table 2
Wall clock times corresponding to Figures 6a to 6d.

t	$r = 10$		$r = 20$		$r = 30$		$r = 60$	
	iter	time [s]	iter	time [s]	iter	time [s]	iter	time [s]
1e−1	140	0.60	70	0.71	80	1.04	100	3.83
1e−2	800	0.75	1,100	2.26	900	3.28	700	9.08
1e−3	5,000	3.27	6,000	8.63	7,000	19.60	9,000	96.78
1e−4	60,000	33.75	80,000	107.57	110,000	314.33	60,000	590.71
1e−5	500,000	288.10	900,000	1,190.97	800,000	2,137.29	800,000	7,521.75

(4.9) (excluding the 1/2 factor) with respect to the total time T. In this calculation, we limit the rank of the iTR to 10 and used a fixed timestep $t = 0.001$. As we can see, θ_1 approaches to λ_0 from above, but θ_2 can fall below λ_0 and eventually approaches λ_0 from below. Although θ_1 and θ_2 both converge towards λ_0, the convergence is much slower than θ which is the average of θ_1 and θ_2.

In addition to approximating the desired eigenvalue by the Rayleigh quotient, we can obtain another type of approximation by computing the eigenvalue of the projected Hamiltonian matrix $H_{\neq 0}$ defined by (4.12). Because the summation in $H_{\neq 0}$ diverges, the eigenvalue of $H_{\neq 0}$ is not finite. However, the eigenvalue of the average of $H_{\neq 0}$ is finite and serves as an approximation to the average eigenvalue of $\mathbf{H}$ defined in (1.2). To compute the average of $H_{\neq 0}$, we can first subtract θI from each of the terms $H_{-2-\ell}$ and $H_{1+\ell}$ in (4.13), for $\ell \geq 1$, where θ is the Rayleigh quotient. We then compute the geometric sums $H_L = H_{-2} + \sum_{\ell=1}^{\infty}(H_{-2-\ell} - \theta I)$ and $H_R = H_1 + \sum_{\ell=1}^{\infty}(H_{1+\ell} - \theta I)$, respectively, using the same techniques as discussed in the proof of Theorem 4.4 for the average residual calculation. We can show that both geometric sums converge. Next, we sum up H_L, H_{-1}, H_0, and H_R after θI is subtracted from each of these terms. The subtraction of θI from H_L and H_R are made to account for the θI's not subtracted from H_{-2} and H_1, respectively, before the geometric sums were performed to yield H_L and H_R. After dividing the sum of all four terms by 4, we add θI back to form the total average of $H_{\neq 0}$, i.e.,

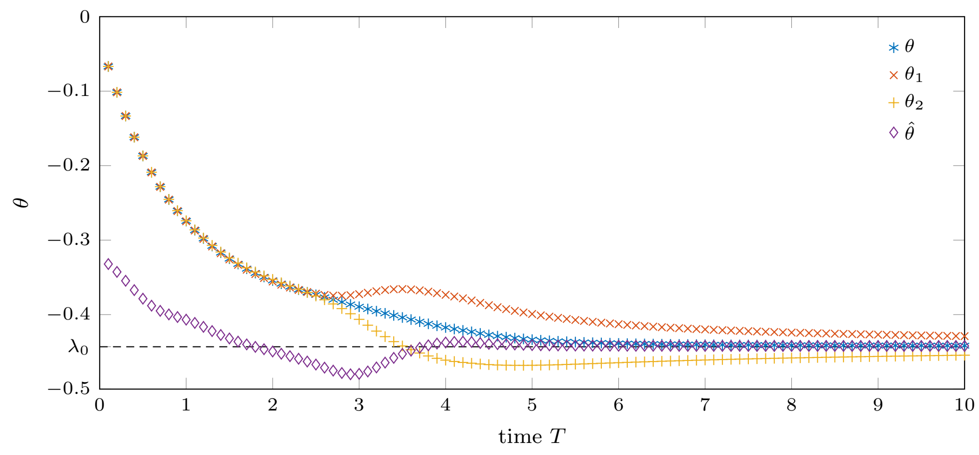
$$\hat{H}_{\neq 0} = (H_L + H_{-1} + H_0 + H_R - 4\theta I)/4 + \theta I = (H_L + H_{-1} + H_0 + H_R)/4.$$

In Figure 7a, the curve marked by the diamonds show that the eigenvalue $\hat{\theta}$ of the average $\hat{H}_{\neq 0}$ converges faster to λ_0. However, computing the average $\hat{H}_{\neq 0}$ can be expensive when the rank of $\mathbf{x}$ becomes large.

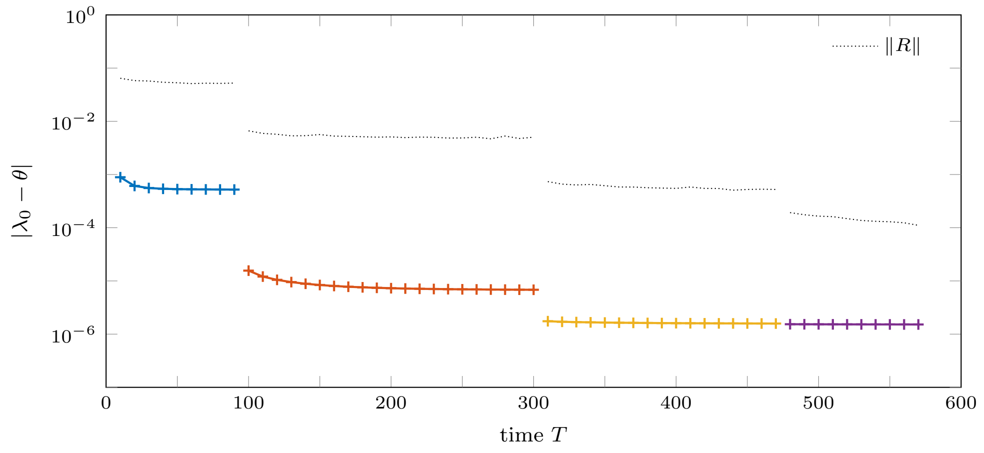
Figure 7b shows how the norm of the average residual and the difference between the approximate and exact eigenvalue change with respect to the total time T when the rank of the iTR is allowed to increased to 120. Even with this much larger rank, it is difficult to reduce the error in the approximate eigenvalue below 10^{-6}.

7. Conclusions.

In this paper, we described a flexible power method for solving a class of infinite dimensional tensor eigenvalue problem in which the infinite dimensional matrix to be partially diagonalized has a special sum of Kronecker product structure shown in (1.2), and the approximate eigenvector can be represented by an infinite translational invariant tensor ring (iTR). We presented some algebraic properties of the iTR in terms of its transfer matrix and showed how to obtain a canonical form which is convenient for manipulating an iTR in several calculations. We showed how the Rayleigh-quotient of an iTR $\mathbf{x}$ with respect to the matrix (1.2) can be properly computed in an average sense. This is the quantity computed in each step of



(a) Convergence of the iTR Rayleigh quotient and its 2 comprising terms ($r = 10$ and $t = 0.001$).



(b) Convergence as a function of total time for $r = 120$.

FIG. 7. *Spin $S = 1/2$ Heisenberg model.*

the flexible power iteration in which the matrix exponential $e^{-\mathbf{H}t}$ is multiplied with $\mathbf{x}$ for some small adjustable parameter t approximately. The flexibility of the power method results from approximations made in each iteration to compute $e^{-\mathbf{H}t}\mathbf{x}$. The approximation involves splitting $e^{-\mathbf{H}t}$ into a product of simpler terms and reducing the rank of the iTR after $e^{-\mathbf{H}t}$ is applied to $\mathbf{x}$.

Although the flexible power method itself is not new, we gave more algorithmic and implementation details in this paper on how to carry out this procedure in a practical setting. In particular, we presented a better way to compute the Rayleigh quotient, and pointed out an alternative way to extract possibly better approximation to the desired eigenvalue. We also developed a practical and effective way to monitor the convergence of the flexible power iteration by computing an average (finite) residual estimate. Such a residual estimate is used to determine when to decrease t to improve convergence and when to terminate the power iteration.

We should note that the flexible power iteration is not the only method for solving this type of infinite dimensional tensor eigenvalue problems. Alternative methods include the infinite density matrix renormalization (iDMRG) method and the variational uniform matrix product states (vUMPS) method. The purpose of our paper is to clearly formulate this type of problem from a numerical linear algebra point of view and describe how a standard numerical algorithm such as the power method can be modified and applied to solve this type of problem. Due to the infinite dimensionality of the eigenvalue problem, special care must be exercised to compute quantities such as the Rayleigh quotient and the residual of an approximate eigenpair. We also need to take into account the approximation that must be made in the multiplication of the matrix exponential $e^{-\mathbf{H}t}$ with an iTR $\mathbf{x}$. A comparison of the flexible power method with other algorithms will be made in future study.

REFERENCES

[1] M. T. Batchelor, *The Bethe ansatz after 75 years*, Phys. Today, 60 (2007), pp. 36–40, https://doi.org/10.1063/1.2709557.

[2] H. Bethe, *Zur Theorie der Metalle*, Z. Phys., 71 (1931), pp. 205–226, https://doi.org/10.1007/BF01341708.

[3] J. C. Bonner, H. W. J. Blöte, H. Beck, and G. Müller, *Quantum Spin Chains*, in Physics in One Dimension, J. Bernasconi and T. Schneider, eds., Berlin, Heidelberg, 1981, Springer Berlin Heidelberg, pp. 115–128.

[4] 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. Commun., 185 (2014), pp. 1207–1216, https://doi.org/10.1016/j.cpc.2013.12.017.

[5] J. Haegeman, J. I. Cirac, T. J. Osborne, I. Pižorn, H. Verschelde, and F. Verstraete, *Time-dependent variational principle for quantum lattices*, Phys. Rev. Lett., 107 (2011), p. 070601, https://doi.org/10.1103/PhysRevLett.107.070601.

[6] L. Hulthén, *Über das Austauschproblem eines Kristalles*, Ark. Mat. Astr. Fys., 26A (1938), pp. 1–106.

[7] D. Kressner, M. Steinlechner, and A. Uschmajew, *Low-rank tensor methods with subspace correction for symmetric eigenvalue problems*, SIAM J. Sci. Comput., 36 (2014), pp. A2346–A2368, https://doi.org/10.1137/130949919.

[8] O. Mickelin and S. Karaman, *On Algorithms for and Computing with the Tensor Ring Decomposition*, 2018, https://arxiv.org/abs/1807.02513.

[9] R. Orús and G. Vidal, *Infinite time-evolving block decimation algorithm beyond unitary evolution*, Phys. Rev. B, 78 (2008), p. 155117, https://doi.org/10.1103/PhysRevB.78.155117.

[10] I. V. Oseledets, *Tensor-train decomposition*, SIAM J. Sci. Comput., 33 (2011), pp. 2295–2317, https://doi.org/10.1137/090752286.

[11] R. Penrose, *Applications of negative dimensional tensors*, in Combinatorial Mathematics and its Applications, D. J. A. Welsh, ed., vol. 1, New York, 1971, Academic Press, pp. 221–244.

[12] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac, *Matrix product state representations*, Quantum Inf. Comput., 7 (2007), pp. 401–430.

[13] P. Pfeuty, *The one-dimensional Ising model with a transverse field*, Ann. Physics, 57 (1970), pp. 79–90, https://doi.org/10.1016/0003-4916(70)90270-8.

[14] S. Rommer and S. Östlund, *Thermodynamic limit and matrix-product states*, in Lecture Notes in Physics, I. Pesschel, M. Kaulke, X. Wang, and K. Hallberg, eds., vol. 528, Berlin, Heidelberg, 1999, Spinger, pp. 67–89.

[15] G. Vidal, *Classical simulation of infinite-size quantum lattice systems in one spatial dimension*, Phys. Rev. Lett., 98 (2007), p. 70201, https://doi.org/10.1103/PhysRevLett.98.070201.

[16] V. Zauner-Stauber, L. Vanderstraeten, M. T. Fishman, F. Verstraete, and J. Haegeman, *Variational optimization algorithms for uniform matrix product states*, Phys. Rev. B, 97 (2018), p. 45145, https://doi.org/10.1103/PhysRevB.97.045145.

[17] Q. Zhao, G. Zhou, S. Xie, L. Zhang, and A. Cichocki, *Tensor ring decomposition*, 2016, https://arxiv.org/abs/1606.05535.