Approximate quantum circuit synthesis using block encodings

Daan Camps [*] and Roel Van Beeumen [†]
Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA

(Received 6 July 2020; accepted 22 October 2020; published 11 November 2020)

One of the challenges in quantum computing is the synthesis of unitary operators into quantum circuits with polylogarithmic gate complexity. Exact synthesis of generic unitaries requires an exponential number of gates in general. We propose a novel approximate quantum circuit synthesis technique by relaxing the unitary constraints and interchanging them for ancilla qubits via block encodings. This approach combines smaller block encodings, which are easier to synthesize, into quantum circuits for larger operators. Due to the use of block encodings, our technique is not limited to unitary operators and can be applied for the synthesis of arbitrary operators. We show that operators which can be approximated by a canonical polyadic expression with a polylogarithmic number of terms can be synthesized with polylogarithmic gate complexity with respect to the matrix dimension.

DOI: 10.1103/PhysRevA.102.052411

I. INTRODUCTION

Quantum computing holds the promise of speeding up computations in a wide variety of fields [1]. After early breakthroughs, such as Shor's algorithm [2] for factoring and Grover's algorithm [3] for searching, there have been substantial developments in various quantum algorithms over the past two decades. Noteworthy are the quantum walk algorithm of Szegedy [4,5], and the quantum linear systems algorithm by Harrow *et al.* [6]. These developments have lead to quantum linear systems [7] and Hamiltonian simulation [8] algorithms inspired by quantum walks. A unifying framework called the quantum singular value transformation, which combines the notion of qubitization [9] and quantum signal processing [10] by Low and Chuang, was recently proposed by Gilyén *et al.* [11]. The quantum singular value transformation can describe all aforementioned quantum algorithms except factoring. Besides that, it has sparked an interest in the use of block encodings since they can directly be used as input for a quantum singular value transformation. A block encoding is the embedding of a—not necessarily unitary—operator as the leading principal block in a larger unitary,

$$U = \begin{bmatrix} A/\alpha & * \\ * & * \end{bmatrix} \Longleftrightarrow A = \alpha(\langle 0| \otimes I)U(|0\rangle \otimes I), \qquad (1)$$

where $*$'s indicate arbitrary matrix elements.

In this paper, we propose the use of block encodings not as a building block for quantum algorithms but as a technique for *approximate* quantum circuit synthesis and, more generally, the synthesis of arbitrary operators into quantum circuits. One of the major challenges on noisy intermediate-scale quantum (NISQ) devices is the limited circuit depth [12]. In general, exact synthesis of generic unitary operators requires exponentially many quantum gates [13–15]. The noise in NISQ devices limits the circuit depth but also relaxes the need for exact synthesis. In other words, we only need to approximate the action of some n-qubit operator up to an error proportional to the noise level. A polynomial dependence of the circuit depth on n is necessary to obtain efficient quantum circuits. Examples of other approximate synthesis approaches have been proposed in Refs. [16–20].

We show that, under certain assumptions, an efficient quantum circuit can be devised if the operator can be ϵ approximated by a canonical polyadic (CP) expression [21,22] with a number of terms that depends polylogarithmically on the operator dimension. We denote these by *PLTCP matrices*. CP decompositions have found applications in many scientific disciplines because they can often be computed approximately using optimization algorithms. However, their calculation is an NP-hard problem in general. We also demonstrate that the class of operators that we can efficiently synthesize is a linear combination of terms with Kronecker product structure, which is more general than standard CP decompositions. We call these *CP-like* decompositions.

The proposed technique uses two operations to efficiently combine block encodings: the Kronecker product of block encodings and a linear combination of block encodings. This allows us to combine block encodings of small matrices into quantum circuits for larger operators. We show that in practice the scheme requires at most a logarithmic number of ancilla qubits, study the relation between the errors on the individual encodings and the overall circuit, and analyze the CNOT complexity of the circuits. Finally, we show three examples of nonunitary operators that naturally have a CP-like structure and can efficiently be encoded using the proposed technique.

II. BLOCK ENCODINGS

Since a n-qubit quantum circuit performs a unitary operation, nonunitary operations cannot directly be handled by quantum computers. One way to overcome this limitation is by encoding the nonunitary matrix into a larger unitary

[*] dcamps@lbl.gov
[†] rvanbeeumen@lbl.gov

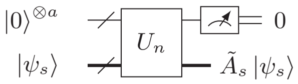

FIG. 1. Quantum circuit for U_n. The thick quantum wire carries the *signal* qubits, the others are the *ancilla* qubits. If the ancilla register is measured in the zero state, the signal register is in the desired state $\tilde{A}_s\,|\psi_s\rangle$.

one, so-called *block encoding* [11]. We define an *approximate* block encoding of an operator on s signal qubits A_s in a unitary U_n on n qubits as follows.

Definition 1. Let $a, s, n \in N$ such that $n = a + s$, and $\epsilon \in R^+$. Then an n-qubit unitary U_n is an (α, a, ϵ)-block encoding of an s-qubit operator A_s if

$$\tilde{A}_s = (\langle 0|^{\otimes a} \otimes I_s)U_n(|0\rangle^{\otimes a} \otimes I_s), \tag{2}$$

and $\|A_s - \alpha\tilde{A}_s\|_2 \leqslant \epsilon$.

The parameters (α, a, ϵ) of the block encoding are, respectively, the *subnormalization factor* to encode matrices of arbitrary norm, the number of *ancilla* qubits, and the *error* of the block encoding. Since $\|U_n\|_2 = 1$, we have that $\|\tilde{A}_s\|_2 \leqslant 1$ and $\|A_s\|_2 \leqslant \alpha + \epsilon$. Note that every unitary U_s is already a (1,0,0)-block encoding of itself and every nonunitary matrix A_s can be embedded in a $(\|A_s\|_2, 1, 0)$-block encoding [23]. This does not guarantee the existence of an efficient quantum circuit.

An equivalent interpretation of Definition 1 is that $\tilde{A}_s$ is the partial trace of U_n over the zero state of the ancilla space. This naturally partitions the Hilbert space $\mathcal{H}_n$ into $\mathcal{H}_a \otimes \mathcal{H}_s$. Given an s-qubit signal state $|\psi_s\rangle \in \mathcal{H}_s$, the action of U_n on $|\psi_n\rangle = |0\rangle^{\otimes a} \otimes |\psi_s\rangle \in \mathcal{H}_n$ becomes

$$U_n\,|\psi_n\rangle = |0\rangle^{\otimes a} \otimes \tilde{A}_s\,|\psi_s\rangle + \sqrt{1 - \|\tilde{A}_s\,|\psi_s\rangle\|_2^2}\,|\phi_n^\perp\rangle\,, \tag{3}$$

with

$$(\langle 0|^{\otimes a} \otimes I_s)\,|\phi_n^\perp\rangle = 0, \quad \||\phi_n^\perp\rangle\|_2 = 1, \tag{4}$$

and $|\phi_n^\perp\rangle$ as the normalized state for which the ancilla register has a state orthogonal to $|0\rangle^{\otimes a}$. By construction, we see that a partial measurement of the ancilla register projects out $|\phi_n^\perp\rangle$ and results in $(|0\rangle^{\otimes a} \otimes \tilde{A}_s\,|\psi_s\rangle)/\|\tilde{A}_s\,|\psi_s\rangle\|_2$ with probability $\|\tilde{A}_s\,|\psi_s\rangle\|_2^2$. In this case, the ancilla register is measured in the zero state, and the signal register is in the target state $\tilde{A}_s\,|\psi_s\rangle$, see Fig. 1. An inadmissible state orthogonal to the desired outcome is obtained with probability $1 - \|\tilde{A}_s\,|\psi_s\rangle\|_2^2$.

Using amplitude amplification, the process must be repeated $1/\|\tilde{A}_s\,|\psi_s\rangle\|_2$ times for success on average. This makes our proposed synthesis technique probabilistic.

III. COMBINING BLOCK ENCODINGS

We introduce two operations on block encodings that in combination allow us to build encodings of larger operators from encodings of small operators. The first operation creates a block encoding of a Kronecker product of two matrices from the block encodings of the individual matrices. We denote a SWAP gate on the ith and jth qubits as SWAP_j^i.

Lemma 1. Let U_n and U_m be (α, a, ϵ_1)- and (β, b, ϵ_2)-block encodings of A_s and A_t, respectively, and define $S_{n+m} = \prod_{i=1}^{s} \text{SWAP}_{a+b+i}^{a+i}$. Then,

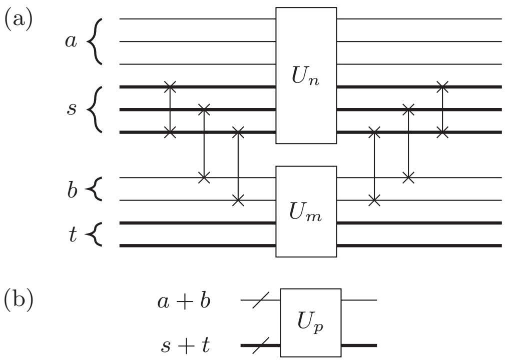

FIG. 2. Block encoding of the Kronecker product of 2 block-encoded matrices: (a) quantum circuit for $a = 3,\ s = 3,\ b = 2,\ t = 2$, and (b) equivalent multiqubit gate U_p with $p = n + m$.

$$S_{n+m}(U_n \otimes U_m)S_{n+m}^\dagger \tag{5}$$

is a $(\alpha\beta, a + b, \alpha\epsilon_2 + \beta\epsilon_1 + \epsilon_1\epsilon_2)$-block encoding of $A_s{\otimes}A_t$.

The proof of Lemma 1 is given in Appendix A. This lemma shows how two individual block encodings can be combined to encode the Kronecker product of two matrices. The method requires no additional ancilla qubits and the approximation error scales as a weighted sum of the individual errors up to first order. The operation requires only $2s$ additional SWAP operations.

Figure 2 shows the quantum circuit for a Kronecker product of block encodings. This reveals the observation that in order to combine block encodings into Kronecker products, the signal qubits of the leading block encoding have to be swapped with the ancilla qubits of the second block encoding in such a way that the $s + t$ signal qubits become the least-significant qubits in the combined circuit and that the mutual ordering of the signal qubits is preserved.

Lemma 1 trivially extends to Kronecker products of more than two block encodings. Let U_{n_i} be $(\alpha_i, a_i, \epsilon_i)$-block encodings of A_{s_i} for $i \in \{1, \ldots, d\}$. Define $n = \sum_i n_i$, and S_n as a SWAP register that swaps all signal qubits of each block encoding U_{n_i} to the least-significant qubits of the n-qubit unitary whereas preserving the mutual ordering between the signal qubits. Then, ignoring the second-order error terms,

$$S_n(U_{n_1} \otimes U_{n_2} \otimes \cdots \otimes U_{n_d})S_n^\dagger \tag{6}$$

is an $(\prod_i \alpha_i, \sum_i a_i, \sum_i \epsilon_i \prod_{k \neq i} \alpha_k)$-block encoding of $A_{s_1} \otimes A_{s_2} \otimes \cdots \otimes A_{s_d}$. In order for the subnormalization factor and approximation error on the Kronecker product not to grow too large, the subnormalization factors of the individual block encodings should be small enough.

The second operation used in the proposed technique constructs a block encoding of a linear combination of block encodings. To this end, we review the notion of a *state preparation pair of unitaries* [11].

Definition 2. Let $y \in \mathbb{C}^m$ with $\|y\|_1 \leqslant \beta$ and define $\underline{y} = [y^T 0]^T \in \mathbb{C}^{2^b}$, where $2^b \geqslant m$. Then the pair of unitaries (P_b, Q_b) is called a (β, b, ϵ)-state-preparation pair for y if

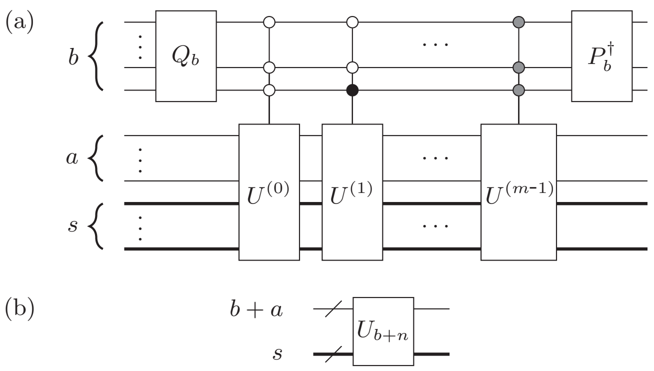

FIG. 3. Block encoding of linear combinations of block encodings: (a) quantum circuit where the white control nodes are controlled on the $|0\rangle$ state, the black control nodes on the $|1\rangle$ state, and the gray control nodes for $U^{(m-1)}$ are controlled on either the $|0\rangle$ or the $|1\rangle$ state in order to encode the bitstring for $m-1$, and (b) equivalent multiqubit gate.

$P_b\,|0\rangle^{\otimes b} = |p\rangle$ and $Q_b\,|0\rangle^{\otimes b} = |q\rangle$ such that

$$\sum_{j=0}^{2^b-1} |\beta(p_j^* q_j) - \underline{y_j}| \leqslant \epsilon. \tag{7}$$

The following lemma is a known result [24], but we provide a sharper upper bound on the approximation error compared to Ref. [11].

Lemma 2. Let $B_s = \sum_{j=0}^{m-1} y_j A_s^{(j)}$ be a s-qubit operator and assume that (P_b, Q_b) is a (β, b, ϵ_1)-state-preparation pair for y. Furthermore, let $U_n^{(j)}$ be (α, a, ϵ_2)-block encodings for $A_s^{(j)}$ for $j \in [m]$ and define the following select oracle:

$$W_{b+n} = \sum_{j=0}^{m-1} |j\rangle\langle j| \otimes U_n^{(j)} + \sum_{j=m}^{2^b-1} |j\rangle\langle j| \otimes I_n. \tag{8}$$

Then,

$$U_{b+n} = (P_b^\dagger \otimes I_a \otimes I_s)W_{b+n}(Q_b \otimes I_a \otimes I_s), \tag{9}$$

is an $(\alpha\beta, a+b, \alpha\epsilon_1 + \beta\epsilon_2)$-block encoding of B_s.

The proof is provided in Appendix B. This lemma shows that, if an efficient state preparation pair exists for the coefficient vector y, then we can efficiently implement a linear combination of block encodings from the individual block encodings. Figure 3 shows the corresponding quantum circuit. Note that this operation requires b additional ancilla qubits. The approximation error again scales as a weighted sum of the (maximum) error on the block encodings and the error on the state-preparation pair.

The combination of Lemma 2 and Eq. (6) shows that we can directly construct a block encoding of a s-qubit operator with the CP-like form

$$B_s = \sum_{j=0}^{m-1} y_j\, A_{s_1}^{(j)} \otimes A_{s_2}^{(j)} \otimes \cdots \otimes A_{s_{d_j}}^{(j)}, \tag{10}$$

if $\sum_{i=1}^{d_j} s_i = s$ for $j \in [m]$, i.e., all terms in the sum in Eq. (10) are of the same dimension, and if we have a block encoding $U_{n_i}^{(j)}$ for each $A_{s_i}^{(j)}$ where $j \in [m]$, and $i \in \{1, \ldots, d_j\}$.

To quantify the subnormalization factor, the number of ancilla qubits, and the approximation error in the block encoding for Eq. (10), we assume that each $U_{n_i}^{(j)}$ is an $(\alpha_i^{(j)}, a_i^{(j)}, \epsilon_i^{(j)})$-block encoding for $A_{s_i}^{(j)}$. Let

$$\alpha^{(j)} = \prod_i \alpha_i^{(j)}, \quad a^{(j)} = \sum_i a_i^{(j)}, \quad \epsilon^{(j)} = \sum_i \epsilon_i^{(j)} \prod_{k\neq i} \alpha_k^{(j)} \tag{11}$$

for $j \in [m]$. Then, using Eq. (6), we can combine these into $(\alpha^{(j)}, a^{(j)}, \epsilon^{(j)})$-block encodings for each term in Eq. (10). Note that whereas the number of signal qubits has to be the same for each term in the linear combination, we do not assume the same number of ancilla qubits here. If we define $a = \max_j a^{(j)}$, then each block encoding for $A_s^{(j)}$ can simply be extended to a ancilla qubits by adding additional ones at the top of the register. This does not change the leading block of the unitary. The properties of a block encoding for Eq. (10) under these assumptions are formalized in the following theorem.

Theorem 1. Let B_s be the s-qubit operator in Eq. (10) with $(\alpha^{(j)}, a^{(j)}, \epsilon^{(j)})$-block encodings of $A_{s_1}^{(j)} \otimes A_{s_2}^{(j)} \otimes \cdots \otimes A_{s_{d_j}}^{(j)}$ for $j \in [m]$, constructed according to Eq. (6) with parameters given by Eq. (11). Assume that all block encodings are extended to $a = \max_j a^{(j)}$ ancilla qubits, $\alpha = \max_j \alpha^{(j)}$, and $\epsilon_1 = \max_j \epsilon^{(j)}$. Then, by Lemma 2, we can construct a unitary U_{b+n} that is an $(\alpha\beta, a+b, \alpha\epsilon_2 + \beta\epsilon_1)$-block encoding of B_s.

Theorem 1 follows directly from the combination of Lemmas 1 and 2. Without loss of generality, the subnormalization factors $\alpha^{(j)} \leqslant \alpha$ can be incorporated in the vector y encoding the coefficients of the linear combination.

The circuit construction can be simplified for operators with the CP structure instead of the CP-like structure. The combination of the SWAP registers from Eq. (6) with the select oracle in Lemma 2 introduces generalized Fredkin gates [25]. Fredkin gates are difficult to realize experimentally [26] and can be avoided if every Kronecker product of the block encodings in the linear combination uses the same SWAP register. In this case, the select oracle becomes

$$W_{b+n} = (I_b \otimes S_n)\tilde{W}_{b+n}(I_b \otimes S_n^\dagger), \tag{12}$$

where

$$\tilde{W}_{b+n} = \sum_{j=0}^{m-1} |j\rangle\langle j| \otimes \tilde{U}_n^{(j)} + \sum_{j=m}^{2^b-1} |j\rangle\langle j| \otimes I_n, \tag{13}$$

with $\tilde{U}_n^{(j)} = U_{n_1}^{(j)} \otimes \cdots U_{n_d}^{(j)}$.

IV. DISCUSSION

Our technique combines block encodings of small matrices to create block encodings of larger operators that can be represented as in Eq. (10). This decomposition is closely related to the CP decomposition of a tensor [21] and allows for more generality. The sizes of the individual block encoded matrices can differ in each term of the linear combination, but they must all have the same size when combined into a Kronecker product.

Optimization algorithms, such as, for example, alternating least squares, have been successfully used to compute approximations to CP decompositions in many applications. Even

TABLE I. Asymptotic CNOT complexity for a quantum circuit that block encodes a PLTCP matrix B_s with s terms in the linear combination and every term a Kronecker product of s 2×2 matrices. The third column lists the CNOT complexity for an exact synthesis of a controlled single qubit gate, the fourth column lists the CNOT complexity for an approximate synthesis [30].

			Total CNOT complexity	
Circuit element	Number	Gates	Exact	Approximate
State preparation $(P_{\log_2(s)}, Q_{\log_2(s)})$ [28]			$\frac{23}{24}s$	
SWAP *registers* [1]	$2s$	SWAP gates	$6s$	
Select oracle	s	Controlled $2s$-qubit	$\Theta(11s^2\log_2(s)^2)$	$\Theta[11s^2\ \log_2(s)\log_2(1/\epsilon)]$
$2s$-qubit with $\log_2(s)$ controls	s	Controlled two-qubit	$\Theta(11s\log_2(s)^2)$	$\Theta[11s\log_2(s)\log_2(1/\epsilon)]$
Two-qubit with $\log_2(s)$ controls [29,30]	11	Controlled one-qubit	$\Theta(11\log_2(s)^2)$	$\Theta[(11\log_2(s)\log_2(1/\epsilon))]$
One-qubit with $\log_2(s)$ controls [30]			$\Theta(\log_2(s)^2)$	$\Theta[\log_2(s)\log_2(1/\epsilon)]$
Toffoli with $\log_2(s)+1$ controls [30]			$\Theta[(\log_2(s)+1)^2]$	$\Theta[(\log_2(s)+1)\log_2(1/\epsilon)]$

though exact CP decompositions are NP-hard to compute in general. The optimization algorithms can be extended to accommodate for the different sizes of block encodings in each of the terms and could incorporate the flexibility in size of the terms in their objective. They can be used as such for approximate quantum circuit synthesis. As NISQ devices suffer from noise [12], the approximate nature of algorithms for CP-like decompositions can be exploited to obtain shorter circuits for less precise decompositions with fewer terms. Under a given noise level, the error on the approximate CP-like decomposition can be balanced with the error on the individual block encodings to find a trade-off with short circuit depth.

One of the major challenges with using block encodings is the introduction of an ancilla register. This removes the constraint of strictly unitary approximations and allows for linear combinations, but at the same time it introduces a probabilistic nature in the synthesis process and requires that the circuit is repeatedly executed until success. This makes our strategy related to the repeat-until-success (RUS) synthesis technique for single qubit unitaries [16,17]. A RUS circuit is a block encoding of the desired operator in combination with a set of recovery operators to recover the input state if a failure state is measured. In our paper, we do not consider recovery operators and assume that the computation is repeated if a failure state is measured.

Another related work is Ref. [27], which proposes basic linear algebra subroutines for quantum computers. Their method relies on Hamiltonian simulation of embeddings of arbitrary matrices and allows to approximate the action of PLTCP-like matrices using Trotter splitting for simulating sums and Kronecker products of matrices.

A. CNOT complexity

The asymptotic gate complexity of the resulting quantum circuit synthesis technique depends on two factors: the number of terms m in the CP-like decomposition in Eq. (10) and the gate count of each individual block encoding in the select oracle. If we assume that $m = O[\mathrm{poly}(s)]$, then $b = O[\mathrm{polylog}(s)]$ and quantum circuits with $O[\mathrm{poly}(s)]$ gates for the state-preparation unitaries always exist [28]. Also the select oracle of Lemma 2 can in this case be implemented with $O[\mathrm{poly}(s)]$ gates.

We call operators that can be expressed as Eq. (10) *PLTCP-like matrices* if the linear combination consists of $O[\mathrm{poly}(s)]$ terms, a polylogarithmic number of terms in the matrix dimension. PLTCP-like matrices can be synthesized with polylogarithmic gate complexity if each term is efficiently implementable. The precise asymptotic complexity depends on the size of every block $A_{s_i}^{(j)}$ and the number of gates required for their block encoding.

The CNOT complexity for the simplest case where B_s is a PLTCP matrix with s terms and where every term is a Kronecker product of s 2×2 matrices is summarized in Table I. The CNOT complexity of the select oracle is determined from the decomposition of two-qubit unitaries [29] and the synthesis of controlled one-qubit unitaries [30].

For PLTCP-like matrices with more complicated structures we still maintain a $O[\mathrm{poly}(s)]$ CNOT complexity as long as the gate complexity for the synthesis of the individual block encodings scales at most with $O[\mathrm{poly}(s)]$. An advantage of this method is that the synthesis of the $O[\mathrm{poly}(s)]$ small block encoding unitaries requires fewer classical resources than the synthesis of larger blocks. The strength of the technique lies in the ability to combine small scale block encodings to build larger operators.

B. Examples

We stress that unitariness of B_s is not required because of the embedding as a block encoding and that even if B_s is unitary, the individual terms in Eq. (10) clearly are not unitary. One class of PLTCP matrices is the Laplace-like operators [31],

$$\sum_{j=1}^{d} M^{(1)} \otimes \cdots \otimes M^{(j-1)} \otimes L^{(j)} \otimes M^{(j+1)} \otimes \cdots \otimes M^{(d)}, \tag{14}$$

and they can directly be encoded from block encodings of the individual terms. For example, in the Laplace operator itself, all $M^{(j)}$ are identities and $L^{(j)} = L$ for $j \in \{1, \dots, d\}$. In this case we only need one block encoding of L, which is repeated d times, to encode the full operator. This is an improvement over the d^2 block encodings that are required in general.

Localized Hamiltonians are another example of PLTCP operators. The Hamiltonian of a transverse field Ising model (TFIM) on a one-dimensional chain of s spin-1/2 particles is

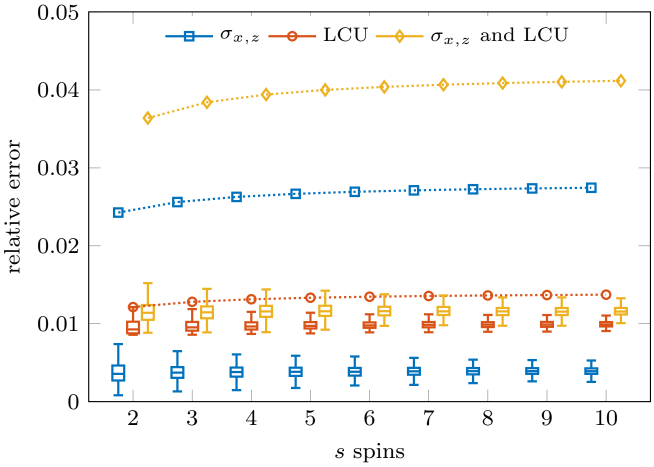

FIG. 4. Results of 1000 simulations of H_{TFIM} with two to ten spins and $h = 2$. The boxplots summarize the empirical relative errors on the block encoding of H_{TFIM} under three different error scenarios: a 1% error on the Pauli-X and Pauli-Z gates (*blue*), a 1% error on the state preparation unitaries for LCU (*red*), and a 1% error on both the Pauli gates and the LCU unitaries (*yellow*). The dotted lines show the theoretical upper bound on the error according to Theorem 1.

given by

$$H_{\mathrm{TFIM}} = -\sum_{i=1}^{s-1} \sigma_z^{(i)}\sigma_z^{(i+1)} - h\sum_{i=1}^{s} \sigma_x^{(i)}, \tag{15}$$

where σ_x and σ_z are the Pauli-X and Pauli-Z matrices. Since this Hamiltonian is a linear combination of $2s - 1$ unitaries, no ancilla qubits are required to encode the 2×2 matrices, and no SWAP operations are necessary to form the Kronecker products. The complexity of block encoding H_{TFIM} lies in forming the linear combination. We have simulated block encoding circuits for H_{TFIM} under three different error scenarios: a 1% error on the σ_x and σ_z gates, a 1% error on the state preparation for the linear combination of unitaries (LCU), and the combination of both. The results are summarized in Fig. 4 with the theoretical upper bound derived from Theorem 1 denoted by the dotted lines.

We observe that errors on the Pauli gates have a smaller effect on the accuracy of the block encoding than errors on the state preparation unitaries. The upper bound slightly overestimates the effect of the errors on the Pauli gates. This happens because the error is not uniformly distributed over the terms in the linear combination in Eq. (15). The expected number of repetitions until success lies between 1.2 and 1.4 for two to ten spins and is not sensitive to errors.

The Hamiltonian for the spin-1 Heisenberg model is equal to

$$H_{\mathrm{XYZ}} = \sum_{i=1}^{s-1} X^{(i)}X^{(i+1)} + Y^{(i)}Y^{(i+1)} + Z^{(i)}Z^{(i+1)}, \tag{16}$$

where X, Y, and Z are the spin-1 generators of SU(2). These 3×3 matrices can be embedded in 4×4 matrices by zero padding and block encoded in two signal qubits and one ancilla qubit. In order to compress the CP rank, we have *tensorized* H_{XYZ} to a s-way $9\times9\times\cdots\times9$ array and numerically computed an approximate CP decomposition using the alternating least-squares algorithm from the TENSOR toolbox [32]. The results for three to six spins are shown in Fig. 5.

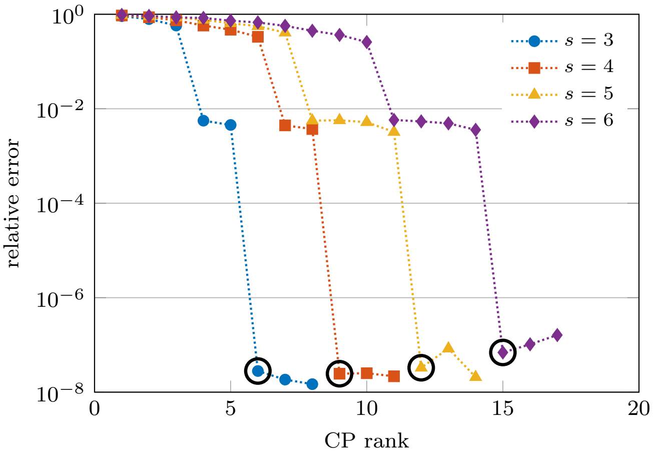

FIG. 5. Compression of the CP rank with the TENSOR toolbox [32] of the Heisenberg isotropic antiferromagnetic Hamiltonian H_{XYZ} for $s = 3,\ldots,6$ spins. The CP rank of the exact decomposition Eq. (16) is circled.

We observe that the relative error on the approximation of the Hamiltonian decreases with increasing CP rank. A stagnation occurs at the exact CP rank of the operator, signaling convergence. If an approximation with a relative error of 1% is sufficient, a CP rank reduction of 20%–30% can be achieved. This directly translates to shorter quantum circuits as each term appears in the select oracle. For example, in the case of $s = 4$ it also leads to a reduction in ancilla qubits: The exact expression is a linear combination of nine terms, requiring four ancilla qubits for encoding the linear combination, and this can be compressed to seven terms, or only three ancilla qubits.

V. CONCLUSIONS

In this paper we showed how block encodings of small matrices, which are easier to synthesize, can be combined together to create block encodings of larger operators with CP-like structure. Under the assumption of $O[\mathrm{poly}(s)]$ terms in the decomposition and small individual block encodings, this scheme has a polynomial dependence on the number of signal qubits both for gate complexity and ancilla qubits. We reviewed three examples of PLTCP matrices, showed that the CP rank can be compressed if a larger approximation error is acceptable, and found that the circuits behave well under errors.

Further research is required to study the class of operators with PLTCP-like structure and operators that can be well approximated in this form. The modification of optimization algorithms for CP decompositions [21] to admit decompositions, such as Eq. (10) is another interesting research direction.

ACKNOWLEDGMENT

This work was supported by the Laboratory Directed Research and Development Program of Lawrence Berkeley National Laboratory under US Department of Energy Contract No. DE-AC02-05CH11231.

APPENDIX A: PROOF OF LEMMA 1

Proof. From Definition 1 and the mixed-product property of the Kronecker product $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$, we obtain

$$\tilde{A}_s \otimes \tilde{A}_t = (\langle 0|^{\otimes a} \otimes I_s \otimes \langle 0|^{\otimes b} \otimes I_t)(U_n \otimes U_m)(|0\rangle^{\otimes a} \otimes I_s \otimes |0\rangle^{\otimes b} \otimes I_t). \tag{A1}$$

The Kronecker product $\tilde{A}_s \otimes \tilde{A}_t$ is encoded in $U_n \otimes U_m$, but not as the leading principal block. We use the property,

$$\text{SWAP}_2^1(I_1 \otimes |0\rangle) = |0\rangle \otimes I_1,$$

to show that S_{n+m} recovers the correct order by swapping the s signal qubits,

$$\begin{aligned} S_{n+m}(|0\rangle^{\otimes a} \otimes I_s \otimes |0\rangle^{\otimes b} \otimes I_t) &= \prod_{i=1}^{s} \text{SWAP}_{a+b+i}^{a+i}(|0\rangle^{\otimes a} \otimes I_s \otimes |0\rangle^{\otimes b} \otimes I_t), \\ &= \prod_{i=1}^{s-1} \text{SWAP}_{a+b+i}^{a+i}\text{SWAP}_{a+b+s}^{a+s}(|0\rangle^{\otimes a} \otimes I_s \otimes |0\rangle^{\otimes b} \otimes I_t), \\ &= \prod_{i=1}^{s-1} \text{SWAP}_{a+b+i}^{a+i}(|0\rangle^{\otimes a} \otimes I_{s-1} \otimes |0\rangle^{\otimes b} \otimes I_1 \otimes I_t), \\ &= \cdots \\ &= |0\rangle^{\otimes a} \otimes |0\rangle^{\otimes b} \otimes I_s \otimes I_t. \end{aligned}$$

Taking the Hermitian conjugate yields

$$(\langle 0|^{\otimes a} \otimes I_s \otimes \langle 0|^{\otimes b} \otimes I_t)S_{n+m}^\dagger = \langle 0|^{\otimes a} \otimes \langle 0|^{\otimes b} \otimes I_s \otimes I_t.$$

Combining this with Eq. (A1) shows

$$\begin{aligned} \tilde{A}_s \otimes \tilde{A}_t &= (\langle 0|^{\otimes a} \otimes I_s \otimes \langle 0|^{\otimes b} \otimes I_t)S_{n+m}^\dagger S_{n+m}(U_n \otimes U_m)S_{n+m}^\dagger S_{n+m}(|0\rangle^{\otimes a} \otimes I_s \otimes |0\rangle^{\otimes b} \otimes I_t), \\ &= (\langle 0|^{\otimes a} \otimes \langle 0|^{\otimes b} \otimes I_s \otimes I_t)S_{n+m}(U_n \otimes U_m)S_{n+m}^\dagger(|0\rangle^{\otimes a} \otimes |0\rangle^{\otimes b} \otimes I_s \otimes I_t), \end{aligned}$$

such that (5) has $\tilde{A}_s \otimes \tilde{A}_t$ as the principal leading block. The subnormalization and approximation error of $\tilde{A}_s \otimes \tilde{A}_t$ satisfy

$$\begin{aligned} &\|A_s \otimes A_t - \alpha\beta\tilde{A}_s \otimes \tilde{A}_t\|_2 \\ &\quad \leqslant \|(\alpha\tilde{A}_s + \epsilon_1 I_s) \otimes (\beta\tilde{A}_t + \epsilon_2 I_t) - \alpha\tilde{A}_s \otimes \beta\tilde{A}_t\|_2, \\ &\quad = \|\alpha\tilde{A}_s \otimes \epsilon_2 I_t + \epsilon_1 I_s \otimes \beta\tilde{A}_t + \epsilon_1 I_s \otimes \epsilon_2 I_t\|_2, \\ &\quad \leqslant \alpha\epsilon_2\|\tilde{A}_s\|_2 + \beta\epsilon_2\|\tilde{A}_t\|_2 + \epsilon_1\epsilon_2, \\ &\quad \leqslant \alpha\epsilon_2 + \beta\epsilon_1 + \epsilon_1\epsilon_2, \end{aligned}$$

where we used that $\|A_s\|_2 \leqslant \alpha\|\tilde{A}_s\|_2 + \epsilon_1$ and $\|\tilde{A}_s\|_2 \leqslant 1$ and analogous results for $\tilde{A}_t$. This completes the proof. ■

APPENDIX B: PROOF OF LEMMA 2

Proof. We have that the leading s-qubit block of U_{b+n} is given by

$$\begin{aligned} \tilde{B}_s &= (\langle 0|^{\otimes b} \otimes \langle 0|^{\otimes a} \otimes I_s)U_{b+n}(|0\rangle^{\otimes b} \otimes |0\rangle^{\otimes a} \otimes I_s), \\ &= (\langle 0|^{\otimes b} \otimes \langle 0|^{\otimes a} \otimes I_s)(P_b^\dagger \otimes I_a \otimes I_s)W_{b+n}(Q_b \otimes I_a \otimes I_s)(|0\rangle^{\otimes b} \otimes |0\rangle^{\otimes a} \otimes I_s), \\ &= (\langle 0|^{\otimes b} P_b^\dagger \otimes \langle 0|^{\otimes a} \otimes I_s)W_{b+n}(Q_b\,|0\rangle^{\otimes b} \otimes |0\rangle^{\otimes a} \otimes I_s), \\ &= (\langle p| \otimes \langle 0|^{\otimes a} \otimes I_s)W_{b+n}(|q\rangle \otimes |0\rangle^{\otimes a} \otimes I_s). \end{aligned}$$

Plugging in the expression for the select oracle Eq. (8), this yields

$$\begin{aligned} \tilde{B}_s &= \sum_{j=0}^{m-1} \langle p|j\rangle\,\langle j|q\rangle \otimes ((\langle 0|^{\otimes a} \otimes I_s)U_n^{(j)}(|0\rangle^{\otimes a} \otimes I_s) + \sum_{j=m}^{2^b-1} \langle p|j\rangle\,\langle j|q\rangle \otimes \langle 0|^{\otimes a}\,|0\rangle^{\otimes a} \otimes I_s, \\ &= \sum_{j=0}^{m-1} p_j^* q_j \tilde{A}_s^{(j)} + \sum_{j=m}^{2^b-1} p_j^* q_j I_s. \end{aligned}$$

By Definitions 1 and 2, we get that

$$\begin{aligned}
\|B_s - \alpha\beta\tilde{B}_s\|_2 &= \left\| \sum_{j=0}^{m-1} y_j A_s^{(j)} - \alpha\beta \sum_{j=0}^{m-1} p_j^* q_j \tilde{A}_s^{(j)} - \alpha\beta \sum_{j=m}^{2^b-1} p_j^* q_j I_s \right\|_2, \\
&= \left\| \sum_{j=0}^{m-1} y_j A_s^{(j)} - \alpha\beta p_j^* q_j \tilde{A}_s^{(j)} - \alpha \sum_{j=m}^{2^b-1} \beta p_j^* q_j I_s \right\|_2, \\
&\leqslant \alpha\epsilon_1 + \left\| \sum_{j=0}^{m-1} \underline{y_j}\big(A_s^{(j)} - \alpha\tilde{A}_s^{(j)}\big) \right\|_2 + \alpha \left\| \sum_{j=m}^{2^b-1} \underline{y_j} I_s \right\|_2, \\
&\leqslant \alpha\epsilon_1 + \beta\epsilon_2.
\end{aligned}$$

The penultimate inequality approximates all $\beta p_j^* q_j$ terms by $\underline{y_j}$ in the two sums. The error of each individual approximation is bounded by ϵ_1 such that the total error is bounded from above by $\alpha\epsilon_1$ as $\|\tilde{A}_s^{(j)}\|_2 \leqslant 1$ and $\|I_s\|_2 = 1$. The last term in the penultimate line is equal to zero by Definition 2. The final equality directly follows from the block encoding property and $\|y\|_1 \leqslant \beta$. ■

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th ed. (Cambridge University Press, New York, 2010).

[2] P. W. Shor, in *Proceedings 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, 1994* (IEEE, Los Alamitos, CA, 1994), pp. 124–134.

[3] L. K. Grover, in *Proceedings 28th Annual ACM Symposium on the Theory of Computing, Philadelphia, 1996* (ACM Press, New York, 1996), pp. 212–219.

[4] M. Szegedy, arXiv:quant-ph/0401053.

[5] M. Szegedy, in *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, Rome, Italy, 2004* (IEEE, Piscataway, NJ, 2004), pp. 32–41.

[6] A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009).

[7] A. M. Childs, R. Kothari, and R. D. Somma, SIAM J. Comput. **46**, 1920 (2017).

[8] D. W. Berry, A. M. Childs, and R. Kothari, in *Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science, Berkeley, CA, 2015* (IEEE, Piscataway, NJ, 2015), pp. 792–809.

[9] G. H. Low and I. L. Chuang, Quantum **3**, 163 (2019).

[10] G. H. Low and I. L. Chuang, Phys. Rev. Lett. **118**, 010501 (2017).

[11] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, 2019* (ACM Press, New York, 2019), pp. 193–204.

[12] J. Preskill, Quantum **2**, 79 (2018).

[13] A. Y. Kitaev, A. H. Shen, and M. N. Vyalyi, *Classical and Quantum Computation* (AMS, Boston, 2002).

[14] V. V. Shende, S. S. Bullock, and I. L. Markov, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **25**, 1000 (2006).

[15] C. M. Dawson and M. A. Nielsen, Quantum Inf. Comput. **6**, 81 (2006).

[16] A. Paetznick and K. M. Svore, Quantum Inf. Comput. **14**, 1277 (2014).

[17] A. Bocharov, M. Roetteler, and K. M. Svore, Phys. Rev. Lett. **114**, 080502 (2015).

[18] E. A. Martinez, T. Monz, D. Nigg, P. Schindler, and R. Blatt, New J. Phys. **18**, 063029 (2016).

[19] S. Khatri, R. LaRose, A. Poremba, L. Cincio, A. T. Sornborger, and P. J. Coles, Quantum **3**, 140 (2019).

[20] E. Younis, K. Sen, K. Yelick, and C. Iancu, arXiv:2003.04462.

[21] T. G. Kolda and B. W. Bader, SIAM Rev. **51**, 455 (2009).

[22] F. L. Hitchcock, J. Math. Phys. **6**, 164 (1927).

[23] G. Alber, T. Beth, M. Horodecki, P. Horodecki, R. Horodecki, M. Rötteler, H. Weinfurter, R. Werner, and A. Zeilinger, *Quantum Information* (Springer-Verlag, Berlin/Heidelberg, 2001).

[24] A. M. Childs and N. Wiebe, Quantum Inf. Comput. **12**, 901 (2012).

[25] E. Fredkin and T. Toffoli, Internat. J. Theoret. Phys. **21**, 219 (1982).

[26] T. Ono, R. Okamoto, M. Tanida, H. F. Hofmann, and S. Takeuchi, Sci. Rep. **7**, 45353 (2017).

[27] L. Zhao, Z. Zhao, P. Rebentrost, and J. Fitzsimons, arXiv:1902.10394.

[28] M. Plesch and Č. Brukner, Phys. Rev. A **83**, 032302 (2011).

[29] G. Vidal and C. M. Dawson, Phys. Rev. A **69**, 010301(R) (2004).

[30] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, Phys. Rev. A **52**, 3457 (1995).

[31] D. Kressner, M. Steinlechner, and A. Uschmajew, SIAM J. Sci. Comput. **36**, A2346 (2014).

[32] B. W. Bader, T. G. Kolda, and Others, MATLABTRNSOR Toolbox Version 3.1, available online (2019).