

An Ancilla Based Quantum Simulation Framework for Non-Unitary Matrices

Ammar Daskin*

Department of Computer Engineering, Istanbul Medeniyet University, Kadikoy, Istanbul, Turkey

Sabre Kais

*Department of Chemistry, Department of Physics and Birck Nanotechnology Center,
Purdue University, West Lafayette, IN, USA and
Qatar Environment and Energy Research Institute, HBKU, Doha, Qatar*

The success probability in an ancilla based circuit generally decreases exponentially in the number of qubits consisted in the ancilla. Although the probability can be amplified through the amplitude amplification process, the input dependence of the amplification makes difficult to sequentially combine two or more ancilla based circuits. A new version of the amplitude amplification known as the oblivious amplitude amplification runs independently of the input to the system register. This allows us to sequentially combine two or more ancilla based circuits. However, this type of the amplification only works when the considered system is unitary or non-unitary but somehow close to a unitary.

In this paper, we present a general framework to simulate non-unitary processes on ancilla based quantum circuits in which the success probability is maximized by using the oblivious amplitude amplification. In particular, we show how to extend a non-unitary matrix to an almost unitary matrix. We then employ the extended matrix by using an ancilla based circuit design along with the oblivious amplitude amplification. Measuring the distance of the produced matrix to the closest unitary matrix, a lower bound for the fidelity of the final state obtained from the oblivious amplitude amplification process is presented. Numerical simulations for random matrices of different sizes show that independent of the system size, the final amplified probabilities are generally around 0.75 and the fidelity of the final state is mostly high and around 0.95. Furthermore, we discuss the complexity analysis and show that combining two such ancilla based circuits, a matrix product can be implemented. This may lead us to efficiently implement matrix functions represented as infinite matrix products on quantum computers.

I. INTRODUCTION AND BACKGROUND

Quantum computers function by using the unitary time evolution operators, i.e. quantum gates, described in the formalism of the standard quantum mechanics. Although non-unitary dynamics can be emulated in the scope of unitary dynamics by using ancilla or measurement based quantum circuits (e.g., [1–3]), the success probability in these circuits diminish substantially with the size of the ancilla register. In most cases, the small success probability can be remedied by using the standard amplitude amplification [4–6]. However, the input dependence of the amplitude amplification makes difficult to sequentially combine two or more ancilla based circuits. In certain types of ancilla based quantum circuits, it is shown that the amplitude amplification can be done on the ancilla system independently of the input to the system register. This type of the amplification process is named as the oblivious amplitude amplification [7, 8]. In different contexts, it is also named as repeat until success algorithm [9, 10] (please see Ref. [11] for a broad overview of the topic). A slightly modified version of the oblivious amplitude amplification is proven to be working also for matrices which are almost unitary: i.e., a non-unitary matrix but somehow close to a unitary ma-

trix [12].

A Hermitian matrix, A , can be used along with $\sqrt{I - A}$ (I is an identity matrix) to form a unitary matrix which can then be processed on an ancilla based quantum circuit. However, finding the square root of a matrix is numerically a difficult problem when the size of the matrix is large. In this paper, we show how to approximate this unitary matrix formation in a way that the produced matrix is almost unitary. In addition, we show that this type of approximate matrices can be processed on the ancilla based circuit where the success probability is maximized by the oblivious amplitude amplification. We also present a lower bound for the fidelity of the final state obtained from the amplification process by measuring the distance of the approximate matrix to the closest unitary matrix. Numerical simulations are done for random matrices of different sizes and it is found that independent of the system size, the final amplified probabilities are generally around 0.75 and the fidelity of the final state is mostly high and around 0.95. This makes possible to implement a product of non-unitary matrices on quantum computers by sequentially combining two ancilla based circuits. Therefore, it may lead us to efficiently implement matrix functions represented as infinite matrix products.

This paper is organized as follows: In the following two subsections, the ancilla based circuits considered in this paper and the amplitude amplification are introduced for the unfamiliar readers. In the next section, first, it is described how to complete a non-unitary matrix to a uni-

* also known as Anmer Daskin. email: adaskin25@gmail.com

tary matrix and then how to approximate it. Then, the numerical examples and the circuit design used in the examples are shown. In Sec.III, the algorithmic complexity of the whole framework is analysed and its possible use in the matrix product and matrix functions is discussed. Finally, the paper is concluded with the summary of the results.

A. Ancilla Based Quantum Circuits

It is well-known that using an ancilla register eases the complexity in the implementation of quantum gates (e.g. see [13, 14]). A particular type of ancilla based circuits can be described for the matrices represented as a sum of unitary matrices [12, 15]: Consider a unitary matrix, A , which is of dimension $N = 2^n$ and represented as a sum of unitary processes, U_i s:

$$A = \sum_{i=0}^{M-1} U_i. \quad (1)$$

A quantum operation can be described as a matrix vector transformation: For instance, let $|in\rangle$ represent a general input state. The application of A on this input state can be described as $A|in\rangle$. The result of this transformation can be obtained on an ancillary based circuit consisting of an ancilla and a system registers by using U_i s. One way of doing this with an ancilla register composed of $m = \log M$ qubits can be described as follows:

- First, all U_i s are aligned on the system register composed of n qubits.
- Then, the application of each U_i is controlled by the ancilla register. Here, the control scheme is determined from the binary representation of i in a way that U_i is applied when the ancilla is in $|i\rangle$ state. Here, $|i\rangle$ is the i th vector in the standard basis. This circuit can be represented in matrix form by a block diagonal matrix, V :

$$V = \begin{pmatrix} U_0 & & \\ & \ddots & \\ & & U_{M-1} \end{pmatrix}. \quad (2)$$

- The initial state of the ancilla register is assumed to be $|0\rangle$ state. Then, it is put into the superposition state by the application of a series of Hadamard gates, $H^{\otimes m}$. Thus, the whole circuit in matrix form becomes equal to the following:

$$\mathcal{U} = (H^{\otimes m} \otimes I)V = \frac{1}{\sqrt{M}} \begin{pmatrix} \sum_{i=0}^{M-1} U_i & J_1 \\ J_2 & J_3 \end{pmatrix}, \quad (3)$$

where I is the identity matrix and J_1, J_2 , and J_3 represent the other parts of \mathcal{U} .

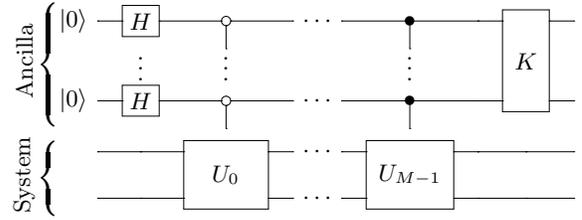


FIG. 1: The circuit representation of \mathcal{U} .

\mathcal{U} can be employed to emulate the action of A with the success probability $\frac{1}{M}$: If we apply \mathcal{U} to the input $|0\rangle|in\rangle$; in the output, for the states in which the ancilla is equal to $|0\rangle$, the circuit gives the action of A on the system register, $A|in\rangle$. Since A in \mathcal{U} is normalized by the coefficient $\frac{1}{\sqrt{M}}$, the probability to see the ancilla in $|0\rangle$ is $\frac{1}{M}$.

This can be also generalized to $A = \sum_{i=0}^{M-1} k_i U_i$ by using an operator K whose first row and column are equal to the coefficients k_0, \dots, k_{M-1} (Note that the coefficients are assumed to be normalized.):

$$\mathcal{U} = (H^{\otimes m} \otimes I)V(K \otimes I) = \frac{1}{\sqrt{M}} \begin{pmatrix} \sum_{i=0}^{M-1} k_i U_i & J_1 \\ J_2 & J_3 \end{pmatrix}. \quad (4)$$

K can be considered as a Householder transformation for the vector $\langle \mathbf{k} | = (k_0, \dots, k_{M-1})$: i.e., $K|\mathbf{k}\rangle = |0\rangle$ and $K|0\rangle = |\mathbf{k}\rangle$. The circuit representing \mathcal{U} is drawn in Fig.1.

B. Amplitude Amplification

As derived from Eq.(4), the success probability of emulating the unitary matrix A by the circuit, \mathcal{U} , is $\frac{1}{M}$. The probability depends on the size of the ancilla and decreases exponentially in the number of qubits contained in the ancilla. This can be increased upto one by the application of the amplitude amplification, which is the main part of the famous Grover's search algorithm [4] finding a solution encoded in a quantum state.

The amplitude amplification is described as follows [5, 6, 16]: Let the state $|\psi\rangle$ be defined as the linear combination of the standard basis resulted from the application of the operator B to an initial zero state:

$$|\psi\rangle = B|0\rangle = \sum_{x=0}^{N-1} b_x |\mathbf{x}\rangle, \quad (5)$$

where b_x is the amplitude of the x th vector, $|\mathbf{x}\rangle$, in the standard basis. Let also some states in $|\psi\rangle$ be considered as the "good" states represented by the set X_{good} , and the remaining states be the "bad" states represented by

X_{bad} :

$$|\psi\rangle = \sum_{x \in X_{good}} b_x |\mathbf{x}\rangle + \sum_{x \in X_{bad}} b_x |\mathbf{x}\rangle. \quad (6)$$

The re-normalizations of X_{good} and X_{bad} form a new two dimensional basis set $\{|\psi_{good}\rangle, |\psi_{bad}\rangle\}$. $|\psi\rangle$ can be rewritten in this new basis with coefficients represented by $\sin(\theta)$ and $\cos(\theta)$ with $\theta \in [0, \pi/2]$:

$$|\psi\rangle = \sin(\theta) |\psi_{good}\rangle + \cos(\theta) |\psi_{bad}\rangle. \quad (7)$$

Here, $\sin(\theta)^2 = P_{good} = \sum_{x \in X_{good}} |b_x|^2$, and $\cos(\theta)^2 = P_{bad} = \sum_{x \in X_{bad}} |b_x|^2$.

In the amplitude amplification algorithm, the amplitude of $|\psi_{good}\rangle$ is amplified by using the following search iterate:

$$Q = U_\psi^\perp U_f. \quad (8)$$

Here, when applied to $|\psi\rangle$, U_f negates the sign of $|\psi_{good}\rangle$ and does nothing to $|\psi_{bad}\rangle$:

$$U_f |\psi\rangle = -\sin(\theta) |\psi_{good}\rangle + \cos(\theta) |\psi_{bad}\rangle. \quad (9)$$

When $|\psi_{good}\rangle$ is known, U_f can be easily constructed in matrix form as follows:

$$U_f = I - 2 |\psi_{good}\rangle \langle \psi_{good}|. \quad (10)$$

The other operator in the search iterate is U_ψ^\perp described in matrix form as:

$$U_\psi^\perp = 2 |\psi\rangle \langle \psi| - I = BU_0^\perp B^{-1}, \quad (11)$$

where $U_0^\perp = (2|\mathbf{0}\rangle \langle \mathbf{0}| - I)$ is the reflection about the axis orthogonal to $|\mathbf{0}\rangle$, and B^{-1} is the inverse of the matrix B . When B is unitary, B^{-1} is equal to the conjugate transpose of B , B^\dagger .

It is easy to prove that when Q is applied k number of times to $|\psi\rangle$, the final state ends up in the following:

$$Q^k |\psi\rangle = \sin((2k+1)\theta) |\psi_{good}\rangle + \cos((2k+1)\theta) |\psi_{bad}\rangle. \quad (12)$$

From the above equation, the highest probability for the good states, $\sin((2k+1)\theta)^2$, can be obtained when $(2k+1)\theta \approx \frac{\pi}{2}$. This leads $O(\frac{\pi}{4\theta})$ as a bound for k .

1. Application to \mathcal{U} and Oblivious Amplitude Amplification

In the circuit \mathcal{U} given in Fig.1, the good and the bad states are already known: the set of good states are those where the ancilla qubit is in $|\mathbf{0}\rangle$ state and the rest is the set of the bad states. Therefore, the iteration operator used in the amplitude amplification can be determined.

Let $|\beta\rangle$ be the output state of \mathcal{U} . $|\beta\rangle$ can be written in terms of the combination of the good and the bad states:

$$|\beta\rangle = \frac{1}{\sqrt{M}} |\beta_{good}\rangle + \frac{\sqrt{M-1}}{\sqrt{M}} |\beta_{bad}\rangle. \quad (13)$$

Since the good states are the ones where the ancilla register equal to zero, the normalized $|\beta_{good}\rangle$ reads the following:

$$|\beta_{good}\rangle = |\mathbf{0}\rangle \sum_{x=0}^{N-1} \beta_x |\mathbf{x}\rangle, \quad (14)$$

where $|\mathbf{x}\rangle$ is the x th vector in the standard basis.

It is shown that the amplitude amplification can be done on the first register by using the following search iterate [7, 8, 12].

$$Q = -USUS, \quad (15)$$

where S is considered in place of U_f and has the same effect as U_0^\perp in the iteration:

$$S = \left(I^{\otimes m} - 2|\mathbf{0}\rangle \langle \mathbf{0}| \right) \otimes I^{\otimes n}. \quad (16)$$

The main difference of this iteration from the standard amplitude amplification is the independence from the input to the system register: \mathcal{U} does not include the input preparation operator which converts a zero state into the desired input.

Since the amplitude of the good states is $1/\sqrt{M}$; Q is applied exactly $\left\lfloor \frac{\pi}{4} \sqrt{M} \right\rfloor$ number of times to maximize the amplitude of $|\beta_{good}\rangle$.

II. APPLICATION IN THE CASE A IS NON-UNITARY

When A is non-unitary; the expected output from the circuit \mathcal{U} can no longer be described by the fixed amplitudes given in Eq.(13) since it changes based on the input state to the system. This necessitates the inclusion of the input preparation in the operator U_ψ^\perp of the standard amplitude amplification:

$$U_\psi^\perp = \mathcal{U}(I^{\otimes m} \otimes A_{in})U_0^\perp(I^{\otimes m} \otimes A_{in})^\dagger \mathcal{U}^\dagger, \quad (17)$$

where A_{in} converts the zero state into the desired input state. To reach the maximum amplitude, in this case, Q needs to be applied $\left\lfloor \frac{\pi}{4\eta} \sqrt{M} \right\rfloor$ number of times, where η is the norm of the expected output state produced from $\|UU_{in}|\mathbf{0}\rangle\|$. Since η is not known, one can apply quantum search algorithm [16]. In that case, the first register is partially measured: If it is zero, then the algorithm is stopped. Otherwise, the iteration operator Q is applied again.

In the following subsections, we shall show how to extend A to a unitary matrix in order to avoid input dependence in the amplitude amplification and successfully apply the oblivious amplitude amplification.

A. Extending A to a Unitary Matrix

It is known that every Hermitian positive definite matrix has a unique Hermitian positive definite square root which can be numerically computed via Schur decomposition [17, 18]. Using the square root of $(I - A^2)$, a non-unitary Hermitian matrix, A , can be extended to a unitary, U , as in the following form:

$$U = \begin{pmatrix} \frac{A}{\sqrt{I - A^2}} & \sqrt{I - A^2} \\ & -A \end{pmatrix}. \quad (18)$$

It is not difficult to see that U is a unitary matrix: $UU^\dagger = U^\dagger U = I^{\otimes n+1}$. This can be seen easier when U is written in the cosine-sine decomposition as:

$$U = \begin{pmatrix} \frac{A}{\sqrt{I - A^2}} & \sqrt{I - A^2} \\ & -A \end{pmatrix} = \begin{pmatrix} R & 0 \\ 0 & R \end{pmatrix} \begin{pmatrix} C & S \\ S & -C \end{pmatrix} \begin{pmatrix} R^\dagger & 0 \\ 0 & R^\dagger \end{pmatrix}, \quad (19)$$

where the eigendecomposition of A is described by $A = RCR^\dagger$, and C and S are the diagonal matrices with the diagonal elements cosine and sine of the eigenvalues of the matrix A . Note that this cosine-sine decomposition can also be used to generate a quantum circuit for U in terms of single and two qubit gates [19–21].

1. Simulation of U

Computation of $\sqrt{I - A^2}$ requires the eigendecomposition of $I - A^2$. Since the eigenvalues of $\sqrt{I - A^2}$ and A are related as cosine and sine of some values, the eigendecomposition of A is enough to form the matrix $\sqrt{I - A^2}$. Eigendecomposition of a matrix can be found through numerical methods such as power iterations and QR method. For a Hermitian matrix whose size is reasonable for classical computers, obtaining a kind of approximate decomposition is feasible in general. However, when the system size is huge, this problem becomes a very heavy computational task for the classical computers so that, in some cases, decomposition may be impossible (note that the computational complexity in terms of the system size is generally $O(N^3)$.) [22].

Therefore, when the computation of the eigendecomposition of A is available; one can use the matrix A on quantum computers as follows: In the circuit for U , the initial state of the first qubit is always set to $|0\rangle$. Thus, we apply U to an initial state $|input\rangle = |0\rangle|in\rangle$, where $|in\rangle$ represents the initial input desired to be applied to A . This generates the following output vector:

$$|output\rangle = \begin{pmatrix} \frac{A}{\sqrt{I - A^2}} |in\rangle \\ |in\rangle \end{pmatrix} \quad (20)$$

To the above vector, if we apply the operator $P = (|0\rangle\langle 0| \otimes I^n)$, then the system collapses to the state $A|in\rangle$. Therefore, at the end of circuit $U|input\rangle$, the probability to see the first qubit in zero state, determines the success probability of the simulation done by the following process.

When the eigendecomposition of A is not available (because of the computational complexity, this may be the case when the system size of A is large), we shall try to estimate U and process the estimated non-unitary on the ancilla based circuit described in the previous section.

2. Estimation of U

From now on, we assume that A is a real symmetric matrix. Then, we normalize the matrix A as $A = \frac{A}{\mu}$ with $\mu = \sqrt{\max_i \sum_j A_{ij}^2}$. This makes 2-norms of the columns of the matrix less than or equal to 1. Then, instead of the matrix in Eq.(18), we construct the following matrix:

$$U = \begin{pmatrix} A & D \\ D & -A \end{pmatrix}, \quad (21)$$

where D is a diagonal matrix described by the diagonal entries $D_{ii} = \sqrt{1 - \sum_j A_{ij}^2}$. The matrix U is not a unitary matrix; however, $A^2 + D^2$ and so $UU^\dagger = U^\dagger U = I^2$ have diagonal elements equal to one. In addition, since A is normalized with μ ; $\rho(A^2) \leq \rho(A) \leq 1$, where $\rho(A)$ is the spectral radius (the largest eigenvalue) of A . In particular cases such as $\rho(A) < 1$ or a diagonally dominant A , the off-diagonal entries of A^2 are likely to be small, and so one can expect the matrix U to be close to a unitary since $A^2 + D^2$ becomes close to an identity matrix.

We shall describe the closeness of U to a unitary matrix via polar decomposition which is known to produce the closest unitary matrix to a given arbitrary matrix (e.g. see [23]): In the polar decomposition, $U = \tilde{U}\tilde{H}$, while \tilde{U} is the closest unitary to U , \tilde{H} is an Hermitian positive definite matrix. We define the following as a measure of the closeness of the estimated unitary U to \tilde{U} :

$$c = \frac{\|U - \tilde{U}\|^2}{\|U\|^2} = \frac{\|\tilde{H} - I\|^2}{\|\tilde{H}\|^2}, \quad (22)$$

where $\|\cdot\|$ represents the norm of a matrix. Also note that a similar measure, $\|U - \tilde{U}\|$, is used by Berry et al. as an error bound in the modified version of the oblivious amplitude amplification [12]. To bound the value of c , we will use the Frobenius norm: $\|\cdot\|_F$, which is also invariant under unitary transformations. Thus, $\|\tilde{H}\|_F = \|U\|_F = \sqrt{\text{trace}(UU^\dagger)} = \sqrt{\text{trace}(\tilde{H}^2)} = \sqrt{2N}$. In addition, we can represent the term $\|\tilde{H} - I\|_F$ as follows:

$$\begin{aligned} \|\tilde{H} - I\|_F &= \sqrt{\text{trace}(\tilde{H}^2 - 2\tilde{H} + I)} \\ &= \sqrt{\text{trace}(\tilde{H}^2) - \text{trace}(2\tilde{H}) + \text{trace}(I)} \quad (23) \\ &= \sqrt{4N - \text{trace}(2\tilde{H})} = \sqrt{4N - \Phi}. \end{aligned}$$

From the generalized mean equality; since the trace is the sum of eigenvalues, the following inequality holds:

$\frac{1}{2N} \text{trace}(\tilde{H}) \leq \sqrt{\frac{1}{2N} \text{trace}(\tilde{H}^2)}$. As a consequence of this, $\Phi = \text{trace}(2\tilde{H}) \leq 4N$ and so $\sqrt{4N - \Phi} \leq 2N$. This makes c in terms of the Frobenius norms always less than or equal to one. When $\Phi = 4N$, c becomes zero and so the matrix can be considered a unitary matrix. Note that one can also try to maximize c for different choices of A .

In the numerical examples, we shall compute c by using 2-norms of the matrices instead of the Frobenius norms: Since the error also depends on the input state, the 2-norm obtained from the maximum eigenvalue likely gives a better approximation for the possible maximum error. Furthermore, we shall use the following as an estimate lower bound for the fidelity of the output after the oblivious amplitude amplification algorithm:

$$ef = |1 - c|^2. \quad (24)$$

The estimated fidelity would be one for the unitary matrices. And it would be small when the closeness between U and \tilde{U} is small.

When U is non-unitary, it cannot be used directly on quantum computers. Berry et al.[12] have proved that a slightly modified version of the oblivious amplitude amplification works also for any arbitrary non-unitary matrix which is close to a unitary matrix. Therefore, since the constructed matrix in Eq.21 is expected to be close to a unitary matrix, the modified oblivious amplitude amplification can be used for the simulation of this matrix in the ancilla based circuit. However, in the next section, the numerical results for the randomly generated matrices show that one can use the oblivious amplitude amplification given in Eq.(15) without modification for the matrices generated by (21). Note that the modified version of the oblivious amplitude amplification would give better results; however, it requires a projection operator.

B. Numerical Examples

In the numerical examples, we shall use the standard oblivious amplitude amplification given in Eq.(15) along with the circuit design method described in Ref.[15]. The circuit design method is only included for the completeness. One can also run simulations by generating random matrices in the form Eq.(21) and then writing each one of them as a direct sum of M unitary matrices.

1. General Circuit Designs

An ancilla-based quantum circuit design technique is introduced in Ref. [15]. A $2n$ -qubit system \mathcal{U} is employed to emulate a general n -qubit quantum operation, U . The primary intuition of the technique is based on generating the matrix-vector product result of a quantum operation inside the new system. This intuition is conveyed in two

parts: First, the ancilla register is put into the superposition state, then the following matrix V similar to the one in Eq.(2) is used:

$$V = \begin{pmatrix} \begin{matrix} u_{11} & u_{12} & \dots & u_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ \bullet & \bullet & \dots & \bullet \end{matrix} & & & \\ & \begin{matrix} u_{21} & u_{22} & \dots & u_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \bullet & \bullet & \dots & \bullet \end{matrix} & & \\ & & \ddots & \\ & & & \begin{matrix} u_{N1} & u_{N2} & \dots & u_{NN} \\ \vdots & \vdots & \ddots & \vdots \\ \bullet & \bullet & \dots & \bullet \end{matrix} \end{pmatrix}, \quad (25)$$

where u_{ij} s are the matrix elements of U , and each “•” represents a matrix element which plays no significant role in the description. If this matrix is applied to a quantum state in which the ancilla is in the superposition state, the desired output is obtained in the states where the system register is in $|0\rangle$ state:

$$V\left(\frac{1}{\sqrt{M}} \sum_{x=0}^{M-1} |x\rangle |input\rangle\right) = \frac{1}{\sqrt{M}} \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_i \\ \vdots \\ \beta_{N-1} \\ \vdots \end{pmatrix}. \quad (26)$$

Obviously, when we swap the ancilla and the system register, we obtain the same output as in Eq.(13) with the good states in Eq.(14) (Note that here $M = N$). As a consequence, the same amplitude amplification given in Eq.(15) can be applied to increase the probability upto one.

Having a non-unitary U does not change the circuit design. However, it changes the success probability given in the above equation. When U is close to a unitary matrix, we can apply the oblivious amplitude amplification in Eq.(15) (One can also use the modified version of the oblivious amplitude amplification given in Ref.[12]).

2. Numerical Examples

In the numerical examples, the oblivious amplitude amplification given in Eq.(15) is applied to the non-unitary matrices constructed in the form given in Eq.(21). The following method is used to generate random symmetric matrices:

- An $N \times N$ symmetric random matrix, A , with the elements in $[-1, 1]$ is generated.
- Then, A is normalized by $\mu = \sqrt{\max_i \sum_j A_{ij}^2}$.
- Then, the diagonal matrix D is obtained: $D_{ii} = \sqrt{1 - \sum_j A_{ij}^2}$.

- Then, the matrices A and D are used to construct $2N \times 2N$ matrix U .

In addition, we also generate random input state to U in the form: $|input\rangle = (|0\rangle \otimes |in\rangle)$, where $|in\rangle$ is a random quantum state with real coefficients.

We have run the simulations for 100 random cases with U of orders $N = 16$, $N = 32$, $N = 64$, and $N = 128$ (Note that the total system sizes with the ancilla register becomes $(2^8 \times 2^8)$, $(2^{10} \times 2^{10})$, $(2^{12} \times 2^{12})$ and $(2^{14} \times 2^{14})$, respectively.). For each random case, the estimated fidelity given in Eq.(24) has been computed as follows:

- First, to find the polar decomposition of U , the singular value decomposition of U is computed: $U = V_1 E V_2^\dagger$, where E represents the singular values and V_1 and V_2 are the right and left singular vectors.
- Then, the closest unitary is computed from $\tilde{U} = V_1 V_2^\dagger$.
- Then, c given in Eq.(22) is computed by using the 2-norms of the matrices.
- Finally, the estimated fidelity is computed from Eq.(24).

In the amplitude amplification process, the amplitudes of the first N states-i.e., the states where the first register is in $|0\rangle$ -are used to compute the success probability. Furthermore, the estimated fidelity is compared with the real fidelity which is the inner product of the collapsed quantum state (the first N amplitudes) and the vector $(A|in\rangle)$.

In Fig.2, both U and $|in\rangle$ are generated randomly. And from the last iteration of the oblivious amplitude amplification (the state in which the success probability is the highest), the fidelity and probability are computed for each case. As can be observed in the figure, the mean value of the observed fidelities is 0.9462 and the mean value of the observed success probabilities is 0.7666.

Furthermore, in Fig.3, the same matrix U is tested in the amplitude amplification with different input states in order to observe the effect of the change in the input state on the fidelity and the validity of the estimated fidelity. Fig.3 shows that the fidelity remains high and is always above the estimated fidelity (the mean of fidelities in all subfigures is 0.9426). Therefore, the proposed estimated fidelity may provide a good lower bound for the fidelity. As in Fig.2, the mean value of the success probabilities here also remain as 0.7601.

In Fig.4, we also present the change of the fidelity and the probability in the iterations of the oblivious amplitude amplification for random cases with the matrices of the same orders: $N = 16$, $N = 32$, $N = 64$, and $N = 128$. The required number of iterations for each case has been computed from $k = \lceil \frac{\pi}{4} \sqrt{N} \rceil$. For $N = 16$, $N = 32$, $N = 64$, and $N = 128$; respectively, $k = 3$, $k = 4$, $k = 6$, and $k = 8$ drawn as dashed vertical lines

on the figures. As seen in the figure: the fidelities in all figures remain high for a few iterations, but then start gradually decreasing while the probabilities are rising. When the probabilities reach their maximums either at the k th iteration or before that; the fidelities go down to around 0.95.

As a general observation, we can see that the numerical results presented in Fig.2, Fig.3, and Fig.4 remain mostly similar despite the change in the system size. This indicates the independence of the results from the system size.

III. DISCUSSION

3. The Total Circuit Complexity

Quantum circuit design methods can be broadly categorized into stochastic and non-stochastic approaches [24]. In terms of stochastic approaches, generally genetic and evolutionary algorithms are used to obtain a circuit which minimizes an optimization function measuring the closeness of the found circuit to the given operator (e.g. [25]). Note that instead of designing a quantum circuit implementing a specific matrix, one can also consider the quantum control approach using optimization to find a way to implement the operator directly, without gates (e.g. [26]). This can also be done for non-unitary processes. On the other hand, non-stochastic deterministic methods are generally based on matrix decomposition techniques: for instance, methods based on the Cartan decomposition [27], the cosine-sine decomposition [21], the QR decomposition [28] are presented. It is shown that a circuit for a general unitary matrix requires at least $(2^{2n-2} - 3n/4 - 1/4)$ number of two-qubit quantum gates [21].

The combination of the circuits for each U_i forms a binary coded network controlled by the ancilla register. These types of networks can be decomposed into single and two qubit quantum gates by using a simple decomposition technique described in Ref.[20]. The decompositions of the networks for the circuit design in Ref. [15] yield a circuit with $O(N^2 - N)$ number of CNOT gates if $M = N$ (For the complete complexity analyses, please refer to Ref.[15]). Therefore, the quantum complexity of the circuit U is $O(N^2 - N)$. The amplitude amplification requires $O(\sqrt{N})$ repetitions which makes the total complexity $O(\sqrt{M}N^2 - \sqrt{M}N)$.

If each U_i can be efficiently simulatable: i.e. each requires only $O(poly(n))$ gates; the total circuit complexity for U can be bounded by $O(poly(n)M)$: Since there would be $O(poly(n))$ number of quantum gates for each U_i , they are likely to compose $O(poly(n))$ number of binary coded networks. The decomposition of such networks produce, in total, $O(poly(n)M)$ number of quantum gates. On the other had, the amplitude amplification would require $O(\sqrt{M})$ repetitions yielding

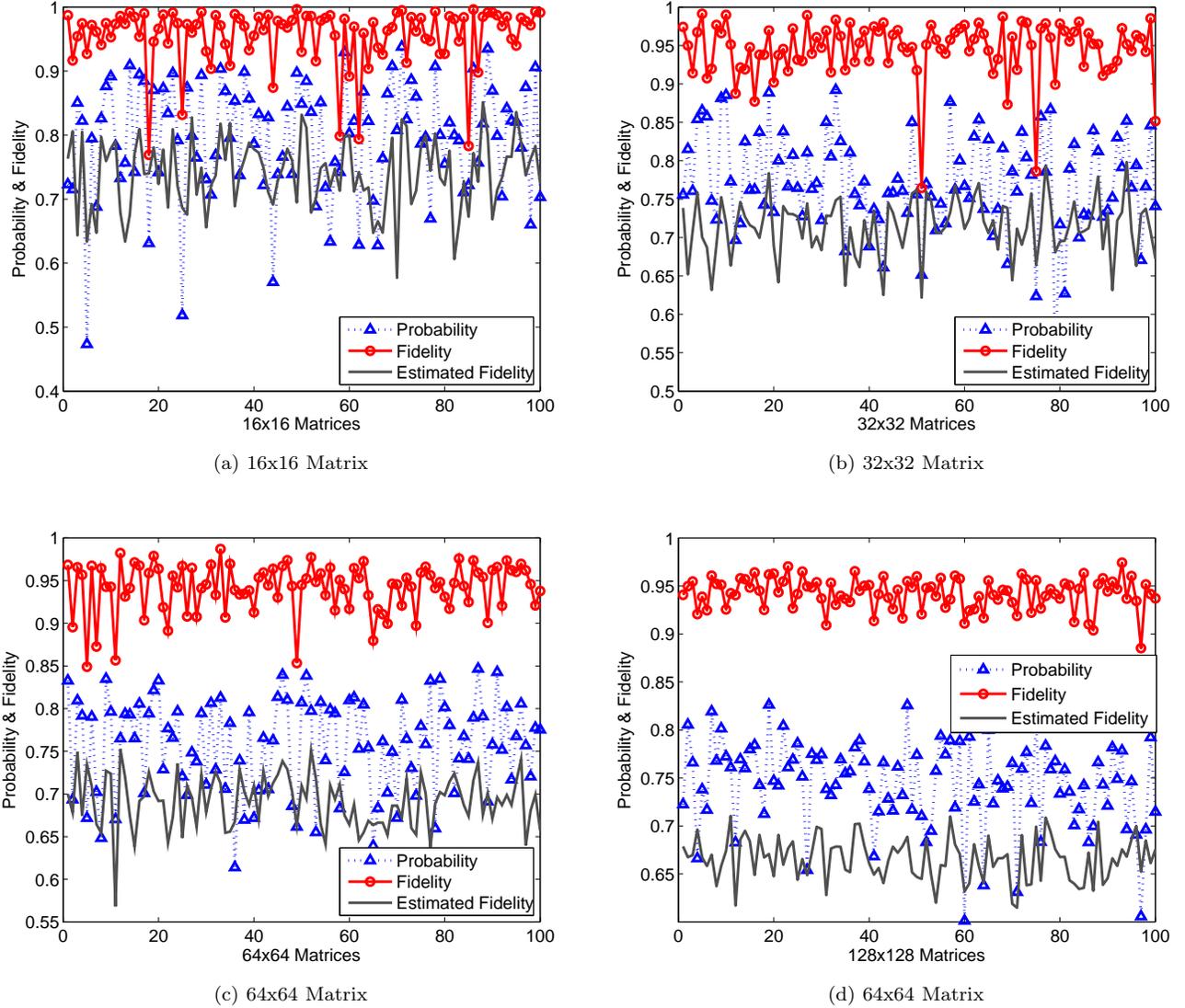


FIG. 2: The change of the fidelity, the probability, and the estimated lowest fidelity at the end of the amplitude amplification process for 100 random matrices of different sizes and random inputs.

$O(\text{poly}(n)M\sqrt{M})$ number of CNOT and single quantum gates. When $M = N$, this becomes $O(\text{poly}(n)N\sqrt{N})$.

$\prod_{j=1}^r W_j$ by the following:

$$\prod_{j=1}^r Q_j^k U_j. \quad (27)$$

A. The Circuit Simulation of a Matrix Product

Let \mathcal{U}_1 and \mathcal{U}_2 are the universal circuits for the unitary matrices, respectively, W_1 and W_2 . Then, the circuit for the product $W_1 W_2$ can be implemented as $Q_2^k \mathcal{U}_2 Q_1^k \mathcal{U}_1$, where \mathcal{U}_j represents the universal circuit, Q_j is in the form of Eq.(15) and $k = \lfloor \frac{\pi}{4} \sqrt{M} \rfloor$. This can be generalized to the product consisting of r number of matrices,

Here, if each U_j requires $O(N^2)$ quantum gates, the obtained circuit implementation for the matrix product requires $O(r\sqrt{MN}^2)$ quantum gates or $O(\sqrt{MN}^2)$ for $r \ll N$. When U_j can be efficiently implementable, e.g. $O(\text{poly}(n))$, the matrix product requires only $O(\text{poly}(n)M\sqrt{M})$ number of quantum gates.

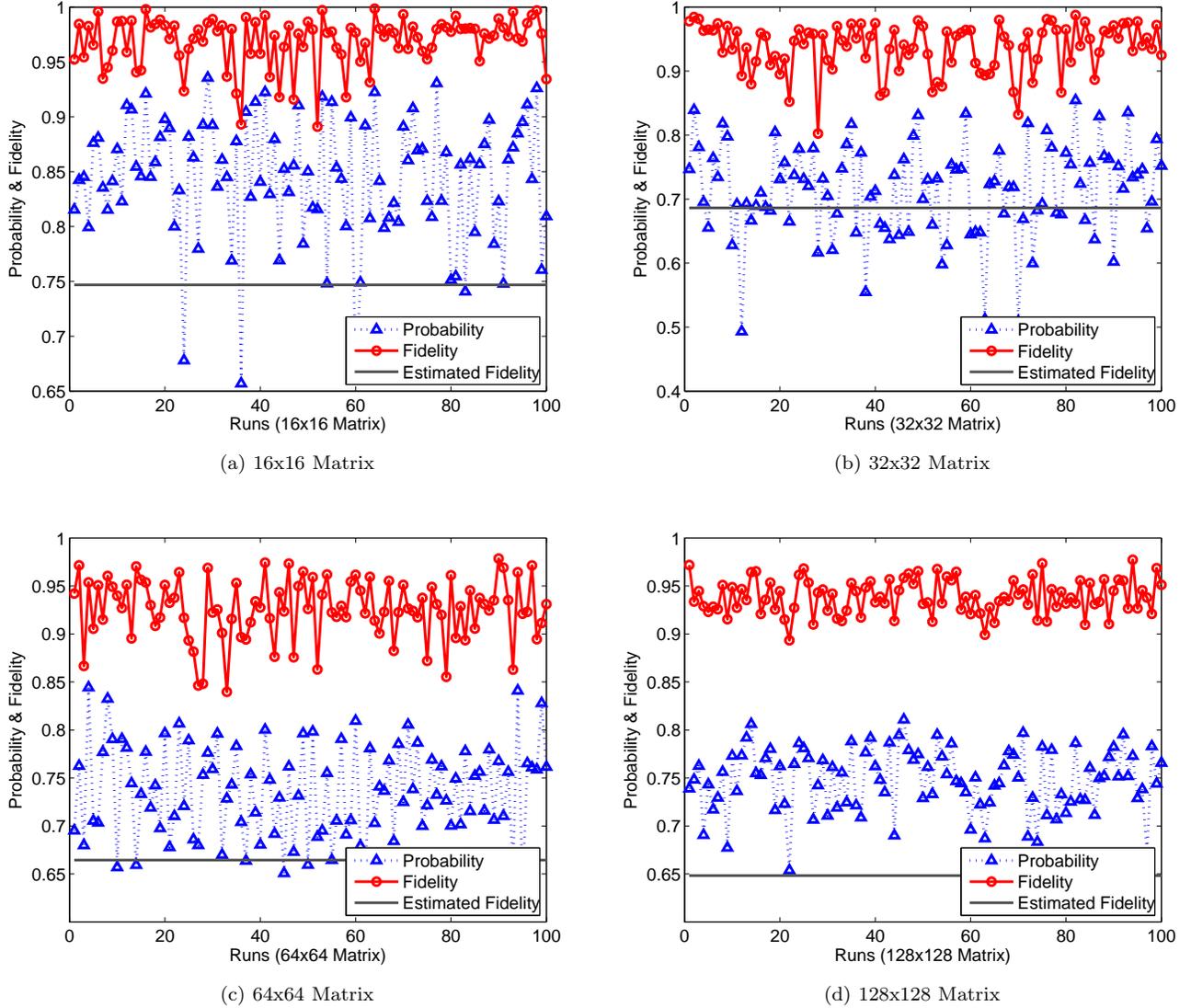


FIG. 3: The change of the fidelity, the probability, and the estimated lowest fidelity at the end of the amplitude amplification process. In each run, the matrix U is unchanged but a different random input is used.

1. Non-Unitary Matrix Product

When the standard amplitude amplification is employed; the amplitude amplification in the second circuit requires the whole amplitude amplification process in the first circuit to be applied again in every iteration. Because of this input dependence, this results in a recursion which makes the whole thing very inefficient.

However, when U is non-unitary but close to a unitary, one can apply the oblivious amplitude amplification. In that case, the complexity of the matrix product becomes the same as the unitary case.

B. Quantum Circuits for the Functions of a Matrix

Many trigonometric functions can be represented in the forms of infinite products (Please see the first chapter of Ref.[29]). Some of the infinite product formulas are also valid for matrices. For instance, formulas for the exponential and the cosines are as follows: Let $A \in \mathbb{C}^{\otimes n}$, then the product formula for the exponential of this matrix is defined as [30]:

$$e^A = \lim_{k \rightarrow \infty} \left(1 + \frac{A}{k}\right)^k = \lim_{k \rightarrow \infty} W_k^k. \quad (28)$$

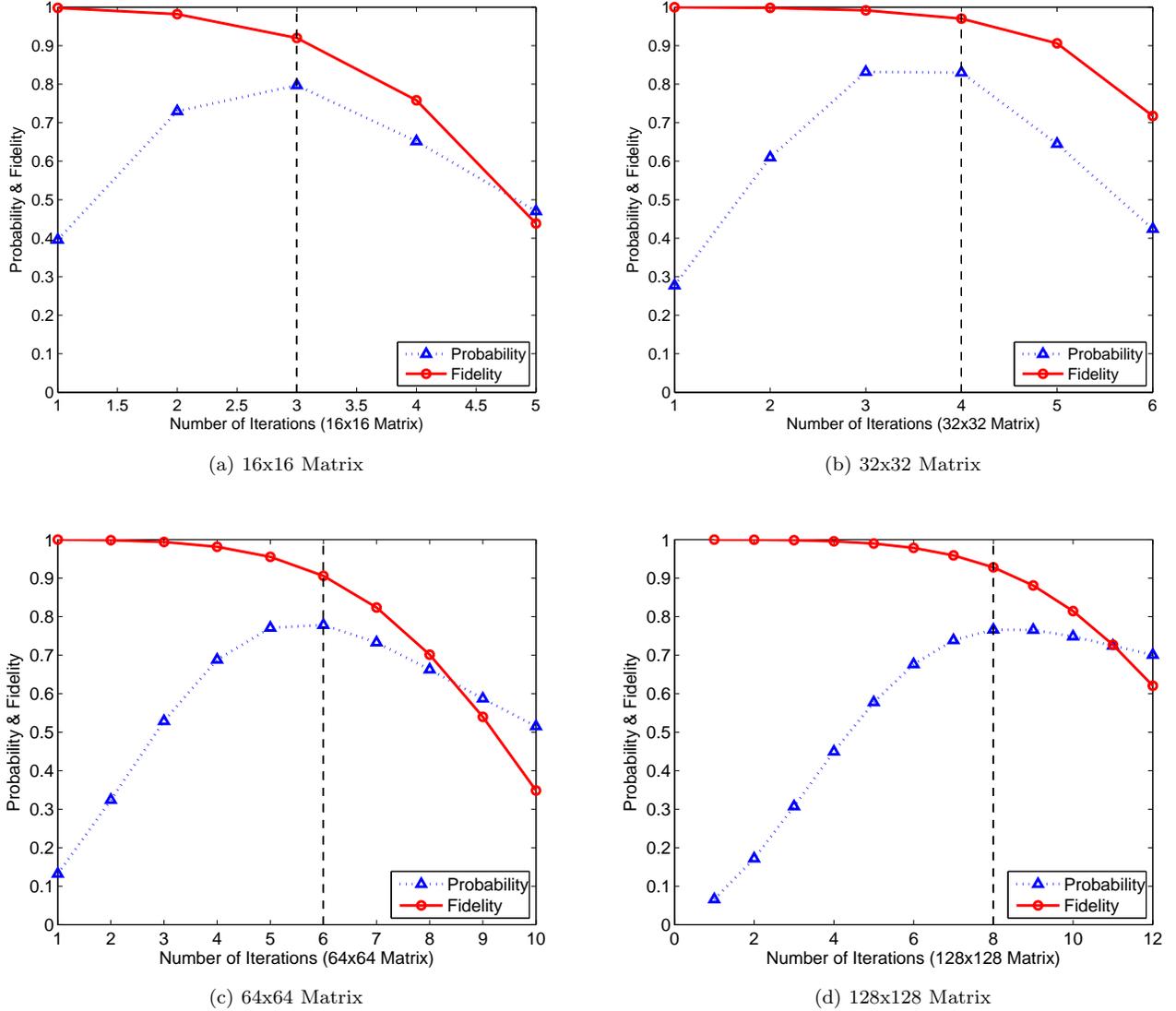


FIG. 4: The change of the fidelity and the probability in the amplitude amplification process for the random matrix U given in Eq.21 of different sizes. In each figure, the vertical dashed line shows the value of $k = \lfloor \frac{\pi}{4} \sqrt{N} \rfloor$.

This can be simply proved by the binomial expansion:

$$\left(I + \frac{A}{k}\right)^k = \sum_{j=0}^k \frac{1}{k^j} \frac{k!}{j!(k-j)!} A^j. \quad (29)$$

If we plug this into Eq.(28), then we attain the Taylor expansion of the matrix exponential and complete the proof:

$$\lim_{k \rightarrow \infty} \left(I + \frac{A}{k}\right)^k = \lim_{k \rightarrow \infty} \sum_{j=0}^k \frac{1}{k^j} \frac{k!}{j!(k-j)!} A^j = \sum_{j=0}^{\infty} \frac{1}{j!} A^j = e^A. \quad (30)$$

In a similar fashion, the cosine representation of A can be defined in the product form as:

$$\begin{aligned} \cos(\pi A) &= \prod_{j=1}^{\infty} \left(I - \frac{4A^2}{(2j+1)^2} \right) \\ &= \prod_{j=1}^{\infty} \left(I - \frac{2A}{2j+1} \right) \left(I + \frac{2A}{2j+1} \right) = \prod_{j=1}^{\infty} W_{j1} W_{j2} \end{aligned} \quad (31)$$

As in the matrix product case discussed in the previous section, first each W_j in the product formulas is extended to the form of U given in Eq.(21) where W_j replaces A . Then, we generate the circuit designs \mathcal{U}_i for each W_j by either writing U as a sum of simple unitary matrices or

following the circuit design in Ref.[15] used in the numerical examples. Then, the same product formula given in Eq.(27) is applied.

IV. CONCLUSION

In this paper, after extending a non-unitary matrix to a unitary matrix, we have described a method to approximate this unitary extension. It is shown that in most cases the approximation gives a non-unitary matrix sufficiently close to a unitary matrix. Therefore, this matrix can be simulated on an acilla based circuit where the success probability is increased by the oblivious amplitude amplification. Since the oblivious amplitude amplifica-

tion does not depend on the input to the system register, using the proposed framework one can implement a matrix product for non-unitary matrices by combining two ancilla based quantum circuits. This allows quantum computers to implement the approximation of the matrix functions represented as infinite matrix products. A non-unitary matrix can be written in terms of sums of unitary matrices: when each unitary matrix in the summation can be efficiently implementable, then one may also implement the non-unitary matrix efficiently. Therefore, one may also implement the matrix product and so the approximations of matrix functions efficiently. We have run numerical simulations for random matrices and shown how the fidelities and the probabilities change in each random case.

-
- [1] Hefeng Wang, Lian-Ao Wu, Yu-xi Liu, and Franco Nori, “Measurement-based quantum phase estimation algorithm for finding eigenvalues of non-unitary matrices,” *Phys. Rev. A* **82**, 062303 (2010).
- [2] Hiroaki Terashima and Masahito Ueda, “Nonunitary quantum circuit,” *International Journal of Quantum Information* **3**, 633–647 (2005).
- [3] Anmer Daskin, Ananth Grama, and Sabre Kais, “A universal quantum circuit scheme for finding complex eigenvalues,” *Quantum information processing* **13**, 333–353 (2014).
- [4] Lov K Grover, “Quantum computers can search rapidly by using almost any transformation,” *Physical Review Letters* **80**, 4329 (1998).
- [5] Michele Mosca *et al.*, “Quantum searching, counting and amplitude amplification by eigenvector analysis,” in *MFCS98 workshop on Randomized Algorithms* (1998) pp. 90–100.
- [6] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp, “Quantum amplitude amplification and estimation,” *Contemporary Mathematics* **305**, 53–74 (2002).
- [7] Adam Paetznick and Krysta M Svore, “Repeat-until-success: Non-deterministic decomposition of single-qubit unitaries,” *Quantum Information & Computation* **14**, 1277–1301 (2014).
- [8] Dominic W Berry, Andrew M Childs, Richard Cleve, Robin Kothari, and Rolando D Somma, “Exponential improvement in precision for simulating sparse hamiltonians,” in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing* (ACM, 2014) pp. 283–292.
- [9] Yuan Liang Lim, Almut Beige, and Leong Chuan Kwek, “Repeat-until-success linear optics distributed quantum computing,” *Phys. Rev. Lett.* **95**, 030505 (2005).
- [10] Yuan Liang Lim, Sean D. Barrett, Almut Beige, Pieter Kok, and Leong Chuan Kwek, “Repeat-until-success quantum computing using stationary and flying qubits,” *Phys. Rev. A* **73**, 012304 (2006).
- [11] Robin Kothari, *Efficient algorithms in quantum query complexity*, Ph.D. thesis, University of Waterloo (2014).
- [12] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma, “Simulating hamiltonian dynamics with a truncated taylor series,” *Phys. Rev. Lett.* **114**, 090502 (2015).
- [13] Benjamin P. Lanyon, Marco Barbieri, Marcelo P. Almeida, Thomas Jennewein, Timothy C. Ralph, Kevin J. Resch, Geoff J. Pryde, Jeremy L. O’Brien, Alexei Gilchrist, and Andrew G. White, “Simplifying quantum logic using higher-dimensional Hilbert spaces,” *Nature Physics* **5**, 134–140 (2009).
- [14] T. D. Mackay, S. D. Bartlett, L. T. Stephenson, and B. C. Sanders, “Quantum walks in higher dimensions,” *Journal of Physics A: Mathematical and Theoretical* **35**, 2745 (2002).
- [15] Anmer Daskin, Ananth Grama, Giorgos Kollias, and Sabre Kais, “Universal programmable quantum circuit schemes to emulate an operator,” *The Journal of chemical physics* **137**, 234112 (2012).
- [16] Phillip Kaye, Michele Mosca, and Raymond Laflamme, *An introduction to quantum computing* (Oxford Univ. Press, UK, 2006).
- [17] Åke Björck and Sven Hammarling, “A schur method for the square root of a matrix,” *Linear algebra and its applications* **52**, 127–140 (1983).
- [18] Nicholas J Higham, “Computing real square roots of a real matrix,” *Linear Algebra and its applications* **88**, 405–430 (1987).
- [19] Robert R Tucci, “A rudimentary quantum compiler (2cnd ed.),” arXiv preprint quant-ph/9902062 (1999).
- [20] Mikko Möttönen, Juha J. Vartiainen, Ville Bergholm, and Martti M. Salomaa, “Quantum circuits for general multiqubit gates,” *Physical Review Letters* **93**, 130502 (2004).
- [21] V. V. Shende, S. S. Bullock, and I. L. Markov, “Synthesis of quantum-logic circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits*.
- [22] Beresford N Parlett, *The symmetric eigenvalue problem*, Vol. 7 (SIAM, 1980).
- [23] Nicholas J Higham, *Matrix nearness problems and applications* (University of Manchester. Department of Mathematics, 1988).
- [24] Anmer Daskin, *Quantum circuit design methods and applications*, Ph.D. thesis, PURDUE UNIVERSITY (2014).
- [25] Anmer Daskin and Sabre Kais, “Decomposition of unitary matrices for finding quantum circuits: application to molecular hamiltonians,” *The Journal of chemical*

-
- physics **134**, 144112 (2011).
- [26] Vladimir M Stojanović, A Fedorov, A Wallraff, and C Bruder, “Quantum-control approach to realizing a toffoli gate in circuit qed,” *Physical Review B* **85**, 054504 (2012).
- [27] Byron Drury and Peter Love, “Constructive quantum shannon decomposition from cartan involutions,” *Journal of Physics A: Mathematical and Theoretical* **41**, 395305 (2008).
- [28] George Cybenko, “Reducing quantum computations to elementary unitary operations,” *Computing in Science & Engineering* **3**, 27–32 (2001).
- [29] Daniel Zwillinger, *Table of integrals, series, and products* (Elsevier, 2014).
- [30] Dennis S Bernstein, *Matrix mathematics: theory, facts, and formulas* (Princeton University Press, 2009).