Asymptotically Optimal Quantum Circuits for d-Level Systems

Stephen S. Bullock,[1] Dianne P. O'Leary,[1,3] and Gavin K. Brennen[2]

[1]*Mathematical and Computational Sciences Division, National Institute of Standards and Technology, Gaithersburg, Maryland 20899-8910, USA*

[2]*Atomic Physics Division, National Institute of Standards and Technology, Gaithersburg, Maryland 20899-8420, USA*

[3]*Department of Computer Science and UMIACS, University of Maryland, College Park, Maryland 20742, USA*

(Received 23 December 2004; published 14 June 2005)

Scalability of a quantum computation requires that the information be processed on multiple subsystems. However, it is unclear how the complexity of a quantum algorithm, quantified by the number of entangling gates, depends on the subsystem size. We examine the quantum circuit complexity for exactly universal computation on many d-level systems (qudits). Both a lower bound and a constructive upper bound on the number of two-qudit gates result, proving a sharp asymptotic of $\Theta(d^{2n})$ gates. This closes the complexity question for all d-level systems (d finite). The optimal asymptotic applies to systems with locality constraints, e.g., nearest neighbor interactions.

DOI: 10.1103/PhysRevLett.94.230502 PACS numbers: 03.67.Lx, 03.65.Fd

The dominant theoretical model of quantum computation is the quantum circuit [1] acting on qubits. Multilevel quantum logics have been proposed as an alternative to qubits due to the trade off in the tensor structure. For state space dimensions $d > 2$, there is a larger space of local operations, and fewer entangling gates might be required to realize a particular quantum computation. Some quantum algorithms may be designed native to qudits, e.g., the Fourier transform [2]. Additionally, in many candidate systems for quantum computation the physical subsystems encoding the quantum information have dimension $d > 2$. Examples include charge-position states in quantum dots [3], rotational and vibrational states of a molecule [4], harmonic oscillator states [5], and hyperfine levels of alkali atoms [6].

It has been shown that a necessary condition for good resource scaling in quantum computation is the existence of an underlying tensor product structure to the Hilbert space [7]. Equivalently, quantum information should be encoded and processed in multiple subsystems. But an important question remains open: How does the complexity of a quantum algorithm change with the size of the subsystem dimension? The *exact* universality theorem for quantum computation with qudits [8] states that any unitary evolution on many qudits can be constructed to infinite precision using a finite sequence of single qudit and two-qudit unitaries or gates. The complexity of a unitary evolution U on n qudits is that number ℓ for which we have a minimum expression,

$$U = U^1_{j_1 k_1} U^2_{j_2 k_2} \cdots U^\ell_{j_\ell k_\ell}, \tag{1}$$

with each U^p_{jk} a two-qudit ($d^2 \times d^2$) operator acting exclusively on qudits j, k. Any two-qudit unitary can be constructed using fewer than $4d^4$ controlled- one-qudit phase gates $e^{i\phi|d-1\rangle\langle d-1|\otimes|d-1\rangle\langle d-1|}$ and approximately $4d^4$ single qudit Givens rotations [9]. Thus, it does not change the complexity to count only the number of two-qudit gates.

In this Letter, we close the complexity gap for the case of algorithms that compute symmetryless unitaries on n qudits. For qubits, Shende, Markov, and Bullock following Knill have shown that $\Omega(4^n)$ two-qubit gates and indeed roughly $4^n/4$ quantum controlled-NOT's (CNOT's) are required [10,11], while a recent Letter [12] provided an $O(4^n)$ gate construction. (Here, we use the complexity theory conventions that $f(n)$ is $O[g(n)]$ if $f(n) \leq Cg(n)$ for some C, $f(n)$ is $\Omega[g(n)]$ if $f(n) \geq Cg(n)$ for some other C, and $f(n)$ is $\Theta[g(n)]$ if both hold.) These results have quite recently been improved so as to no longer require ancilla qubits and even attain a CNOT count of roughly $4^n/2$ (e.g., [13]). For qudits, the best prior constructive upper bound is $O(n^2 d^{2n})$ two-qudit gates [14], leaving open a gap between a lower bound of $\Omega(d^{2n})$. The main result of our work is a constructive, ancilla-dependent proof that $\Theta(d^{2n})$ two-qudit gates are required to exactly simulate an arbitrary n-qudit evolution without symmetry. En route, we also prove that $\Theta(d^n)$ two-qudit gates suffice for n-qudit state synthesis, i.e., any unitary extension of the d^n parameter map $|0\rangle \rightarrow |\psi\rangle$. The algorithm that produces the quantum circuit is a variant of the QR matrix-decomposition, cf. [12,15,16]. Unlike an earlier qubit construction [12], it does not rely on a GRAY code, either in base two or base d.

Physical implementation of symmetryless evolutions is not practical when the number of qudits n is large, since the number of gates required scales exponentially in n. Yet circuits for generic unitaries are still of interest. First, they may improve subblocks of larger circuits through a process of peephole optimization: when many consecutive two-qudit gates act on a small collection of qudits, we compute the associated unitary evolution and substitute a circuit of the sort presented here in hopes of decreasing the total number of required operations. Second, they are useful in translating circuits from gate libraries that include multi-qudit gates to two-qudit gates when a physical system conveniently allows only for pairwise interactions. They

may also be used to translate an arbitrary gate library into a fault-tolerant library of qudit gates [17]. The symmetries that allow for polynomial-size quantum circuits are not well understood. Producing efficient symmetryless circuits may provide insights into general design principles that might also be useful in constructing computations. Also recall the sharp asymptotic for symmetryless n-qubit unitary evolution: $\Theta(4^n)$ two-qubit gates are required [12].

The lower bound argument uses Sard's theorem from smooth topology. A well-known corollary (e.g., [18]) demands that for a smooth map $f{:}M \to N$ that carries an m-dimensional manifold M into an n-dimensional manifold N for $m < n$, the set image (f) must be a measure zero subset of N. Thus, suppose that we consider an expression associated with a fixed circuit topology for two-qudit gates. Namely, suppose we factor a $U \in \mathrm{U}(d^{2n})$ as in Eq. (1). Moreover, suppose that we take ℓ and the tuples (j_q, k_q) for $1 \le q \le \ell$ to be fixed. Then by varying the $U^q_{j_q k_q}$ in $\mathrm{U}(d^2)$, we obtain a map of smooth manifolds $f{:}[\mathrm{U}(d^2)]^\ell \to \mathrm{U}(d^n)$. Now generically, $\dim[\mathrm{U}(q)] = q^2$. Hence the smooth function implicit in the product of operators in Eq. (1) carries a manifold of dimension ℓd^4 into a manifold of dimension d^{2n}. In order for the set of unitary evolutions realized by varying $d^2 \times d^2$ unitaries in a fixed circuit diagram not to measure 0, we require $\ell \ge d^{2n-4}$. As there are only finitely many circuit topologies holding fewer than d^{2n-4} factors per Eq. (1), we generically require $\Omega(d^{2n-4}) = \Omega(d^{2n})$ gates of the two-qudit library to realize symmetryless unitary evolutions within $\mathrm{U}(d^n)$. A similar argument produces a lower bound of $\Omega(d^n)$ gates for state synthesis.

Consider two emulation schemes of qudits by qubits. First, one might emulate each individual qudit with as few qubits as possible, so that the local qudit structure is respected. Second, one might rather pack the entire d^n-dimensional n-qudit state into the smallest possible qubit state space, ignoring the local (tensor) structure. We argue that the emulation circuit for the first scheme does not attain the lower bound asymptotic, while in essence the second scheme does not allow for circuit-level emulation at all.

In the first scheme, label $\beta = \lceil \log_2 d \rceil$, so that β qubits are required to emulate a qudit. Now for the qubit circuit diagram, some multiqubit gates will in fact be local to the qudit, while others are genuine two-qudit gates. Hence, if U is a $d^n \times d^n$ unitary matrix and the $O(2^{2\beta n})$ circuit is applied after splitting each qudit into β virtual qubits, we obtain an *upper* bound of $O(2^{2\beta n})$ two-qudit gates. This asymptotic is worse than both $O(d^{2n})$ and even $O(n^k d^{2n})$ unless d is a power of 2. Thus, prior art does not suffice for the upper bound asymptotic.

Regarding the second scheme, n qudits may be viewed as d^n Hilbert space dimensions. Ignoring the local structure, a unitary evolution of $\mathcal{H}(n, d) \cong \mathbb{C}^{d^n}$ may be realized as a subblock of a unitary evolution of $n\delta = \lceil n \log_2 d \rceil$ qubits rather than $n\beta = n\lceil \log_2 d \rceil$ as above. Indeed, with this form of emulation, it is true that $O(4^{n\delta}) = O(d^{2n})$ virtual two-qubit gates would suffice by earlier methods. However, in this mode of emulation a virtual two-qubit gate *need not* correspond to a two-qudit gate. Indeed, it might not even be a k-qudit gate for k small. Consider, for example, a two-qutrit gate of the form $I_3 \otimes V$ acting on $\mathcal{H}(3, 3)$, where $V \in \mathrm{U}(3^2)$. This has a 9×9 block structure, but emulating such a unitary using qubits is more or less an arbitrarily difficult five-qubit evolution. It is certainly not a two-qubit gate. Similar reasoning shows that emulation of computation over qudits of dimension d with qudits of dimension d' is usually inefficient, unless $(d')^k = d$ for some integer k.

Although the complexity bound is phrased in terms of two-qudit operators, our QR factorization algorithm produces a quantum circuit of k-controlled one-qudit operators. Our gates are controlled V operations, meaning V is applied to a target qudit based on a string of $n-1$ controls. Each control is either $*$, to denote a match with an arbitrary value (no control), or is chosen to be one of $0, 1, \ldots, d-1$, to force a specific matching value (control). We define a controlled one-qudit gate $\bigwedge(C, V)$ as follows. Let V be a $d \times d$ unitary matrix, i.e., a one-qudit operator. Let $C = [C_1 C_2 \cdots C_n]$ be a length-n *control word* composed of letters from the alphabet $\{0, 1, \ldots, d-1\} \sqcup \{*\} \sqcup \{T\}$, with exactly one letter in the word being T. By $\#C$ we mean the number of letters in the word with numeric values (i.e., the number of controls), and the set of control qudits is the corresponding subset of $\{1, 2, \ldots, n\}$ denoting the positions of numeric values in the word. A control word *matches* a string of dits if each numeric value matches. Then the controlled one-qudit operator $\bigwedge(C, V)$ is the n-qudit operator that applies V to the qudit specified by the position of T iff the control word matches the n-dit string. Alternatively, if $C_j = T (j < n)$, we consider the unitary (permutation) operator χ^n_j that swaps qudits j and n. Then, $\bigwedge(C, V) = \chi^n_j \bigwedge(\tilde{C}, V) \chi^n_j$, where $\tilde{C} = [C_1 C_2 \cdots C_{j-1} C_n C_{j+1} \cdots C_{n-1} T]$. Any $\bigwedge(C, V)$ gate can be implemented using $r = \lceil (\#C - 1)/(d-2) \rceil$ ancillary qudits and $2r(d-1) + 1$ two-qudit gates [14].

The key component of our universal qudit circuit is a subcircuit for quantum state synthesis: given a state vector $|\psi\rangle \in \mathcal{H}(n, d)$, we construct a unitary extension of the mapping $|0\rangle \to |\psi\rangle$. For $d = 2$, several works address this topic, e.g., [11,13,19]. Our state synthesis algorithm constructs a sequence of p gates, each a controlled one-qubit operator depending on $|\psi\rangle$, such that $\Pi_{k=1}^{p} \bigwedge[C(p-k+1), V(p-k+1)]|\psi\rangle = |0\rangle$. The unitary $\bigwedge[C(k), V(k)]$ denotes the controlled operation at step k. Inverting the sequence, $\Pi_{k=1}^{p} \bigwedge[C(k), V(k)^\dagger]|0\rangle = |\psi\rangle$. For our state synthesis algorithm, all the controlled operators are singly controlled. Since our construction realizes any $|\psi\rangle$ in $(d^n - 1)/(d-1) \in O(d^n)$ gates, then given the $\Omega(d^n)$ lower bound, qudit state synthesis generically requires $\Theta(d^n)$ gates.

An important primitive gate we make use of is the one-qudit Householder reflection [[20], Sec. 5.1] Suppose $|\varphi\rangle \in \mathcal{H}(1, d)$, perhaps not normalized. We construct a

unitary operator V such that $V|\psi\rangle$ is a multiple of $|0\rangle$. This V depends on an auxiliary state $|\eta\rangle$:

$$|\eta\rangle = |\varphi\rangle - \sqrt{\langle\varphi|\varphi\rangle}\,\frac{\langle 0|\varphi\rangle}{|\langle 0|\varphi\rangle|}|0\rangle,$$
$$V = I_d - (2/\langle\eta|\eta\rangle)|\eta\rangle\langle\eta|. \tag{2}$$

Then $V|\varphi\rangle$ is a multiple of $|0\rangle$.

We next describe the algorithm for state synthesis of an n-qudit state via the mapping $\Pi_{k=1}^{p}\bigwedge[C(p-k+1), V(p-k+1)]|\psi\rangle = |0\rangle$ with $\#C(k) \le 1$. In Algorithm 1, the total number of one-qudit Householder gates is $p = \Sigma_{k=1}^{n} d^{n-k} = (d^n - 1)/(d-1)$. The algorithm loops over dit strings $i_1 i_2 \cdots i_{\ell-1} j_\ell 0 \cdots 0$. The target qudit position, indicated by the subscript of the free index j, shifts left from qudit $\ell = n$ to qudit $\ell = 1$ as successive components of $|\psi\rangle$ are zeroed. In each dit string, the control position k is the rightmost qudit for which $i_k \neq 0$. If all $i_k = 0$, no control is needed. For the tth dit string, an appropriate one-qudit Householder $\bigwedge[C(p-t+1), V(p-t+1)]$ is applied to the current state $|\psi_t\rangle = \Pi_{k=1}^{t-1}\bigwedge[C(p-k+1), V(p-k+1)]|\psi\rangle$. For example, consider State-synth with an input of five $d = 6$ qudits, at the step t corresponding to the dit string $510j_40$. The control position is two and the target position is four. The singly controlled one-qudit Householder $\bigwedge\{[*, 1, *, T, *], V(p-t+1)\}$ at this step zeros the $d-1$ state vector components $\{\langle 510j_40|\psi_t\rangle, j_4 > 0\}$. The Householder is explicitly determined by forming the one-qudit state $|\varphi\rangle = \Sigma_{j=0}^{5}\langle 510j_40|\psi_t\rangle|j\rangle$ and using Eq. (2) to build $V(p-t+1)$ such that $V(p-t+1)|\varphi\rangle = |0\rangle$. A sequence of elementary gates to build a controlled one-qudit Householder is given in [9].

Listing 1. Algorithm1: State-synth $(|\psi\rangle, d, n)$ produces a series of singly controlled Householder reflections whose product U has $U|\psi_0\rangle = |0\rangle$. For more details, see [21]

```
t = 0
for i_1 = 0:d − 1
 for i_2 = 0:d − 1
  ⋱
   for i_{n−2} = 0:d − 1
    for i_{n−1} = 0:d − 1
     t = t + 1
     Use a one-qudit Householder to zero
     {⟨i_1 i_2 ⋯ i_{n−1} j_n|ψ_t⟩, j_n > 0}
    end
    t = t + 1
    Use a one-qudit Householder to zero
    {⟨i_1 i_2 ⋯ i_{n−2} j_{n−1} 0|ψ_t⟩, j_{n−1} > 0}
   end
   ⋮
  end
 t = t + 1
 Use a one-qudit Householder to zero
  {⟨i_1 j_2 0 ⋯ 0|ψ_t⟩, j_2 > 0}
end
t = t + 1
Use a one-qudit Householder to zero
{⟨j_1 0 ⋯ 0|ψ_t⟩, j_1 > 0}.
```

Figure 1 illustrates the order in which the Householders $\bigwedge(C, V)$ are applied in the State-synth sequence for $d = 3$, $n = 3$. Each box represents a one-qudit Householder and is labeled by the string $i_1 i_2 j$, $i_1 j 0$, or $j00$ from the corresponding step in the algorithm. The Householder is constructed to use the first entry in the box ($j = 0$) to zero the others ($j > 0$). The algorithm traverses the graph in depth-first order, left to right. To understand the controls, notice that the leftmost Householders on each level of the graph ($j00$, $0j0$, and $00j$) require no control. For example, the Householder labeled $0j0$ is applied to nine sets of elements: $0j0$, $0j1$, and $0j2$, all zeroed, and $1j0$, $1j1$, $1j2$, $2j0$, $2j1$, and $2j2$, not yet zeroed. For Householder boxes that are not leftmost in their level, the control is indicated in boldface. For the leftmost Householder in a group of d ($10j$ or $20j$, for example), we do not want to touch elements in boxes to the left of it, so we set the control to stay within the group; for example, the Householder labeled $\mathbf{1}0j$ is also applied to elements in $1\mathbf{1}j$ and $1\mathbf{2}j$. For other Householders in a group, the corresponding elements in boxes to the left are completely zero, and in boxes to the right are as yet unzeroed. For example, the Householder labeled $1\mathbf{1}j$ is applied to $0\mathbf{1}j$ (all zero since $0j0$ has already been applied) and $2\mathbf{1}j$.

We can now count the number of two-qudit gates in State-synth. Let $h(n, k)$ be the number of k-controls required in the State-synth reduction of some $|\psi\rangle \in \mathcal{H}_n$. By construction $h(n, k) = 0$ for $k \ge 2$. Moreover, at the steps of the algorithm corresponding to dit strings of the form $(00\cdots0j0\cdots0)$, no control is needed, and there are n such sequences. Thus, as the number of Householder reflections is $(d^n - 1)/(d-1)$, we see that

$$h(n, 1) = (d^n - 1)/(d-1) - n, \qquad h(n, 0) = n. \tag{3}$$

State synthesis can be generalized to mapping any n-qudit logical basis state $|j\rangle = |j_1 j_2 \ldots j_n\rangle$ to $|\psi\rangle$, by conjugation

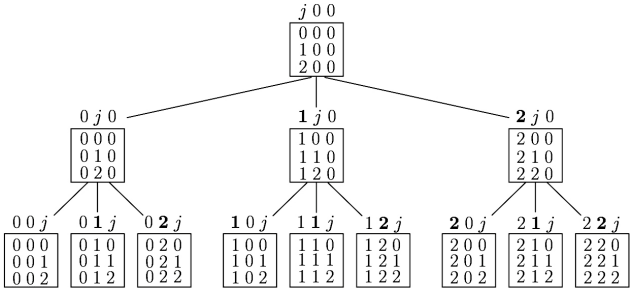

FIG. 1. Sequence of one-qudit Householders to reduce $|\psi\rangle$ to a multiple of $|0\rangle$ for $d = 3$, $n = 3$. Each box represents a Householder $\bigwedge(C, V)$ and is labeled by a dit string. The control position is indicated by the boldface entry in the label and the target position is indicated by the free index j. The $\bigwedge(C, V)$ is constructed to use the state vector component in the top row of the box to zero the $d-1$ components below it.

with local gates: $W_j = [\otimes_{k=1}^{n} \oplus (j_k)]W_0[\otimes_{k=1}^{n} \oplus (d - j_k)]$, where W_0 is a unitary extension of the mapping $|0\rangle \rightarrow |\psi\rangle$ and $\oplus$ denotes addition modulo d.

Efficient state synthesis circuits imply efficient unitary evolution circuits. This observation follows directly from the spectral decomposition [11] of a unitary U, i.e., $U = \Sigma_{j=0}^{d^n-1} e^{i\theta_j}|\lambda_j\rangle\langle\lambda_j|$. Then $U = \Pi_{j=0}^{d^n-1} W_j P_j W_j^\dagger$, where W_j is any unitary extension of the mapping of the logical basis state $|j\rangle$ to the eigenvector $|\lambda_j\rangle$, and $P_j = e^{i\theta_j|j\rangle\langle j|}$ are diagonal unitaries locally equivalent to the $(n-1)$-controlled one-qudit phase gate. Each such phase gate can be implemented using $O(n)$ two-qudit gates. In turn, each W_j admits a size $O(d^n)$ circuit obtained from Stateynth. The two-qudit gate count for an exact unitary implementation is

$$\ell = 2d^{2n}/(d-1) + d^n[2r(d-1) - 2/(d-1) - 2n + 1], \tag{4}$$

with assistance of $r = \lceil (n-2)/(d-2) \rceil$ ancillary qudits.

Finally, the asymptotic gate count is not affected by architectural constraints. Consider a graph with qudits as nodes and physically allowed interactions between nodes as links. Then the connected graph with largest diameter, or longest shortest path between nodes, is a linear chain of qudits restricted to nearest neighbor interactions. When the control and target qudits are separated by a distance k, the total cost is $2k - 1$ two-qudit gates including swap gates.

We count the total cost for state synthesis on n qudits as follows. Let $a(d, n, k)$ denote the number of singly controlled Householder gates with separation k between the control position and the target. The steps in the Stateynth algorithm where a gate is applied are uniquely labeled by a sequence of n-dit strings of the form $i_1 i_2 \cdots i_r 0 \cdots 0 j_s 0 \cdots 0$. The separation of the control and target at each step is $k = s - r$. If no control is needed the separation is $k = 0$. The function $a(d, n, k)$ can be solved by recursion on n. For a dit string on $n + 1$ qudits, a separation k operation results from $k - 1$ consecutive zeroes immediately to the left of the target. This can happen in one of two ways: a length n term of the form $i_1 i_2 \cdots i_r 0 \cdots 0 j_s 0 \cdots 0$ is prepended to become $i_1 i_2 i_3 \cdots i_{r+1} 0 \cdots 0 j_{s+1} 0 \cdots 0$; or the length n term of the form $0 \cdots 0 j_s 0 \cdots 0$ is prepended to become $i_1 0 \cdots 0 j_{s+1} 0 \cdots 0$, for $i_1 \neq 0$. The structure of the sequence produces the following recursion relations:

$$\begin{aligned} a(d, n+1, k) &= (d-1) + d\,a(d, n, k), \\ a(d, n, 0) &= n, \qquad a(d, n, k) = 0 \text{ if } k \geq n - 1. \end{aligned} \tag{5}$$

The recursion implies $a(d,n,k) = (d-1)\Sigma_{\ell=0}^{n-k-1} d^\ell = d^{n-k} - 1$. This implies that the number of nearest neighbor two-qudit gates for quantum state synthesis is $\Sigma_{\ell=0}^{n-1}(2k-1)a(d,n,k) = \Sigma_{\ell=0}^{n-1}(2k-1)(d^{n-k}-1) \in O(d^n)$. For arbitrary unitary synthesis there will also be a cost incurred due to swapping when implementing the $(n-1)$-controlled phase gates. However, the cost of swapping to ancillary qudits is linear. Thus, the asymptotic gate count remains $O(d^{2n})$ with locality restrictions.

We conclude with some remarks. Locality in quantum mechanics is a function of the tensor (Kronecker) product structure of the state space in question. In quantum computing, the Hilbert space factors are often finite dimensional. Measuring difficulty by counting two-particle interactions, we have generalized a recent optimal asymptotic of $\Theta(2^{2n})$ for two-level quantum bits to a new optimal asymptotic $\Theta(d^{2n})$ for d-level quantum dits. The result is exponentially better (asymptotically) than that obtained by emulating such qudits with qubits, given $d \neq 2^\ell$.

D. P. O. acknowledges support from the National Science Foundation under Grant No. CCR-0204084. G. K. B. is partially supported by a grant from DARPA/QUIST.

[1] D. Deutsch, Proc. R. Soc. A **425**, 73 (1989).
[2] P. Hoyer, quant-ph/9702028.
[3] S. G. Schirmer, A. D. Greentree, and D. K. L. Oi, quant-ph/0305052.
[4] E. A. Shapiro *et al.*, Phys. Rev. A **67**, 013406 (2003).
[5] S. D. Bartlett, H. de Guise, and B. C. Sanders, Phys. Rev. A **65**, 052316 (2002).
[6] G. Klose, G. Smith, and P. S. Jessen, Phys. Rev. Lett. **86**, 4721 (2001).
[7] R. Blume-Kahout, C. M. Caves, and I. H. Deutsch, Found. Phys. **32**, 1641 (2002).
[8] J.-L. Brylinski and R. Brylinski, *Mathematics of Quantum Computation*, edited by R. Brylinski and G. Chen (CRC Press, Boca Raton, 2002).
[9] G. K. Brennen, D. P. O'Leary, and S. S. Bullock, Phys. Rev. A **71**, 052318 (2005).
[10] V. V. Shende, I. L. Markov, and S. S. Bullock, Phys. Rev. A **69**, 062321 (2004).
[11] E. Knill, quant-ph/9508006.
[12] J. J. Vartiainen, M. Möttönen, and M. M. Salomaa, Phys. Rev. Lett. **92**, 177902 (2004).
[13] V. V. Shende, S. S. Bullock, and I. L. Markov, quant-ph/0406176.
[14] A. Muthukrishnan and C. R. Stroud, Jr., Phys. Rev. A **62**, 052309 (2000).
[15] A. Barenco *et al.*, Phys. Rev. A **52**, 3457 (1995).
[16] G. Cybenko, Comput. Sci. Eng. **3**, 27 (2001).
[17] D. Gottesman, Chaos Solitons Fractals **10**, 1749 (1999); E. Knill, quant-ph/9608048.
[18] S. S. Bullock and I. L. Markov, Quantum Inf. Comput. **4**, 27 (2004).
[19] M. Möttönen *et al.*, quant-ph/0407010.
[20] G. H. Golub and C. van Loan, *Matrix Computations* (Johns Hopkins Press, Baltimore, 1989).
[21] S. S. Bullock, D. P. O'Leary, and G. K. Brennen, quant-ph/0410116.