# Recognizing small-circuit structure in two-qubit operators

Vivek V. Shende,[1,*] Stephen S. Bullock,[2,†] and Igor L. Markov[3,‡]
[1]*Department of Mathematics, The University of Michigan, Ann Arbor, Michigan 48109-1109, USA*
[2]*Mathematical and Computational Sciences Division, National Institute of Standards and Technology, Gaithersburg, Maryland 20899, USA*
[3]*Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, Michigan 48109-2122, USA*
(Received 8 August 2003; revised manuscript received 5 January 2004; published 19 July 2004)

This work proposes numerical tests which determine whether a two-qubit operator has an atypically simple quantum circuit. Specifically, we describe formulas, written in terms of matrix coefficients, characterizing operators implementable with exactly zero, one, or two controlled-NOT (CNOT) gates and all other gates being one-qubit gates. We give an algorithm for synthesizing two-qubit circuits with an optimal number of CNOT gates and illustrate it on operators appearing in quantum algorithms by Deutsch-Josza, Shor, and Grover. In another application, our explicit numerical tests allow timing a given Hamiltonian to compute a CNOT modulo one-qubit gate, when this is possible.

DOI: 10.1103/PhysRevA.70.012310 PACS number(s): 03.67.Lx, 03.65.Fd, 03.65.Ud

## I. INTRODUCTION

Quantum circuits compactly represent unitary operators and find applications in quantum computing, communication, and cryptography [1]. Such a representation can often be interpreted as a program (e.g., a sequence of pulses for NMR) whose execution on a quantum system of choice performs a requested unitary evolution. Simple steps in the program correspond to gates in the circuit, and smaller circuits lead to faster programs. In this work we discuss exact implementations of two-qubit operators because (i) such operators suffice to implement arbitrary operators [2] and (ii) a number of controllable two-qubit systems were recently reported.

The simulation of generic two-qubit operators via CNOT gates and one-qubit operators has been thoroughly investigated, resulting in several three-CNOT decompositions [3–5]. It is known that the swap gate requires three CNOT gates [4] and also that an arbitrary $n$-qubit operator requires at least $\lceil \frac{1}{4}(4^n - 3n - 1) \rceil$. The proof of this latter result [5] holds for any controlled-$u$ gate, where $u$ is a given fixed one-qubit operator. For $n=2$, it has been shown that an arbitary controlled-$u$ gate is generically worse than the CNOT [6].

The above-mentioned results motivate the focus on the *basic-gate* library [7], which consists of the CNOT gate and all one-qubit gates: it is powerful and well understood. Yet given the diversity of implementation technologies, it is not clear that the CNOT gate will be directly available in a given implementation. Nonetheless, we believe results expressed in the basic-gate library will be relevant. An analogous situation occurs in the design of (classical) integrated circuits. In this context, first *technology-independent synthesis* is performed in terms of abstract gates (AND, OR, NOT). Later, during *technology mapping*, circuits are converted to use gates that are specific to a given implementation technology (e.g., NOR, NAND, and AOI gates, which require very few CMOS transistors). Work in the direction of quantum technology mapping includes techniques for expressing a CNOT gate in terms of a given entangling two-qubit gate and arbitary one-qubit gates [8]. The simulation of CNOT gates with implementation-specific resources is the basis of a major physical implementation technology [9].

The analogy with classical logic synthesis provides the following additional intuition: operators useful in practice will not be the worst-case operators studied in the aforementioned works. This belief is confirmed by published quantum algorithms and communication protocols. It is therefore important for quantum logic synthesis techniques to detect when a given operator can be implemented using fewer gates than are necessary in the worst case. For some classes of operators, this is easy; e.g., the algorithm in [10] implements tensor-product operators without CNOT gates. The matrix of a controlled-$U$ operator can be recognized by its pattern of zeros and ones (either directly or after pre- and post-multiplication by wire swaps). Song and Klappenecker [11] study optimal implementations of two-qubit controlled-unitary operators, known to require up to two CNOT gates. They contribute a catalog of numerical tests that detect when zero, one or two CNOT gates are required, and similar criteria for the number of basic one-qubit gates.

We address a related question for arbitrary two-qubit operators and contribute simple numerical tests to determine the minimal achievable number of CNOT gates, including a novel one-CNOT test. We also generalize a two-CNOT test from [3] and make it easier to compute. Such explicit numerical tests facilitate a new application. A given two-qubit Hamiltonian $H$, if timed precisely, may allow one to implement a CNOT using $e^{iHt}$ and one-qubit gates. We show how to compute correct durations.

## II. BACKGROUND AND NOTATION

It is well known that an arbitrary one-qubit gate $u$ can be written as $u = e^{i\Phi} R_z(\theta) R_y(\phi) R_z(\psi)$ [1]. Furthermore, the

*Electronic address: vshende@umich.edu
†Electronic address: stephen.bullock@nist.gov
‡Electronic address: imarkov@umich.edu

 ©2004 The American Physical Society

Bloch sphere isomorphism suggests that the choice of $y,z$ is arbitrary in the sense that any pair of orthogonal vectors will do: in particular, we may write

$$u = e^{i\Phi}R_z(\theta)R_x(\phi)R_z(\psi) = e^{i\Phi}R_x(\alpha)R_z(\delta)R_x(\beta).$$

These decompositions are more convenient when working with CNOT gates because $R_z$ gates commute through the control of the CNOT whereas $R_x$ gates commute through the target. We will denote by $C_j^k$ a CNOT with control on the $j$th wire and target on the $k$th. For convenience, we consider the CNOT gate to be normalized to have determinant 1.

Additional conventions are as follows. For $g$ any complex matrix, $g^T$ denotes the transpose and $g^\dagger$ denotes the adjoint—i.e., the complex-conjugate transpose. Additionally, $\chi[g] = \det[xI-g]$ denotes the characteristic polynomial of $g$. Finally, we fix the global phase of two-qubit unitary operators up to $\pm 1, \pm i$ by requiring them to be in SU(4).

We now consider when two-qubit operators $u,v$ differ by pre- or post-composing with one-qubit operators and possibly by an irrelevant global phase. In this case, we write $u \equiv v$ and say that $u$ and $v$ are equivalent up to one-qubit gates. The following invariant characterizes when this occurs.

*Proposition 1.* Let $\gamma: \mathrm{U}(4) \to \mathrm{U}(4)$ be given by the formula $u \mapsto u\sigma_y^{\otimes 2}u^T\sigma_y^{\otimes 2}$. Then for $u,v \in \mathrm{SU}(4)$, $u \equiv v$ if and only if $\chi[\gamma(u)] = \chi[\pm\gamma(v)]$.

We have discussed this invariant at length elsewhere, but include a proof in the Appendix for completeness [5]. However, note that this proof provides an explicit procedure for computing the one-qubit operators $a,b,c,d \in \mathrm{SU}(2)$ such that $(a\otimes b)u(c\otimes d) = e^{i\phi}v$ in the event that $\chi[\gamma(u)] = \chi[\pm\gamma(v)]$. Related invariants are discussed in [12,13] and generalizations in [14].

## III. OPTIMIZING CNOT COUNT

We now characterize which two-qubit operators admit a quantum circuit using only $m$ CNOT gates. Since any two-qubit operator is implemented by some three CNOT circuits, the relevant cases are $m=0,1,2$. We begin with case $m=0$.

*Proposition 2.* An operator $u \in \mathrm{SU}(4)$ can be simulated using no CNOT gates and arbitrary one-qubit gates from SU(2) iff $\chi[\gamma(u)] = (x+1)^4$ or $(x-1)^4$.

*Proof.* $u$ can be simulated using no CNOT gates iff $u \equiv I$. Thus $\chi[\gamma(u)] = \chi[\pm\gamma(I)] = \chi[\pm I] = (x\pm 1)^4$. ■

The case $m=1$ is similar. Note that the following test *requires* normalizing the global phase so that $\det(u)=1$, as mentioned in Sec. II.

*Proposition 3.* An operator $u \in \mathrm{SU}(4)$ can be simulated using one CNOT gate and arbitrary one-qubit gates from SU(2) iff $\chi[\gamma(u)] = (x+i)^2(x-i)^2$.

*Proof.* The operator $u$ can be simulated using one CNOT gate iff $u \equiv C_1^2$ or $u \equiv C_2^1$. Now $\gamma(C_2^1) = -i\sigma^z \otimes \sigma^x$; also, $\gamma(C_1^2) = -i\sigma^x \otimes \sigma^z$. In both cases, the characteristic polynomial is $(x+i)^2(x-i)^2$. ■

In particular, we see that $C_1^2 \equiv C_2^1$. This can also be seen from the well-known identity $(H\otimes H)C_1^2(H\otimes H) = C_2^1$. We will use this fact for the final case $m=2$.

*Proposition 4.* An operator $u \in \mathrm{SU}(4)$ can be simulated using two CNOT gates and arbitrary one-qubit gates from SU(2) iff $\chi[\gamma(u)]$ has all real coefficients, which occurs iff $\mathrm{tr}[\gamma(u)]$ is real. Moreover, if $\mathrm{tr}[\gamma(u)]$ is not real, then $u$ cannot be simulated with fewer than three CNOT gates.

*Proof.* Since $C_1^2 \equiv C_2^1$, it is clear that $u$ can be simulated using two CNOT gates iff $u \equiv C_1^2(a\otimes b)C_1^2$. We decompose $a = R_x(\alpha)R_z(\delta)R_x(\beta)$ and $b = R_z(\theta)R_x(\phi)R_z(\psi)$, and pass $R_x$ gates and $R_z$ gates outward through the target and control of the CNOT gates. Thus we are left with $u \equiv C_1^2(R_z(\delta)\otimes R_x(\phi))C_1^2$. Explicit computation yields $\chi[\gamma(C_1^2(R_z(\delta)\otimes R_x(\phi))C_1^2)] = (x+e^{i(\delta+\phi)})(x+e^{-i(\delta+\phi)})(x+e^{i(\delta-\phi)})(x+e^{-i(\delta-\phi)})$. On the other hand, if $\chi[\gamma(u)]$ has all real coefficients, then the eigenvalues come in conjugate pairs; it follows from this and Proposition 1 that $\chi[\gamma(u)]$ is as above for some $\delta,\phi$. Finally, we note that $\mathrm{tr}[\gamma(I)]=4$ and $\mathrm{tr}[\gamma(C_1^2)] = \mathrm{tr}[\gamma(C_1^2)] = 0$, thus if $u$ can be simulated with fewer than two CNOT gates, $\mathrm{tr}[\gamma(u)]$ is real. It follows that a two-qubit operator $u$ requires three CNOT gates iff $\mathrm{tr}[\gamma(u)]$ is not real.

Finally, we note that for $u \in \mathrm{SU}(N)$, and $\chi(u) = \Pi(x-\lambda_i)$, we have $\Pi\lambda_i = 1$. Thus $\chi(u) = (\Pi\overline{\lambda_i})\Pi(x-\lambda_i) = \Pi(\overline{\lambda_i}x-1)$. It follows that the coefficient of $x^k$ is the complex conjugate of the coefficient of $x^{N-k}$. In particular, for $N=4$, the coefficient of $x^2$ is real and the coefficients of $x^3, x$ are $\mathrm{tr}[u]$ and its conjugate. Since the constant term and the $x^4$ coefficient are 1, we see $\chi[u]$ has all real coefficients iff $\mathrm{tr}[u]$ is real. ■

We can use Proposition 4 to recover some previously known facts (originally proven in [4] using different methods). For $u \in \mathrm{SO}(4)$, it is clear that $\gamma(u)$ is real, since $\sigma_y^{\otimes 2}$ is real. Thus $\mathrm{tr}[\gamma(u)]$ is real, and $u$ can be simulated by a circuit with two CNOT gates. On the other hand, explicit computation of $\mathrm{tr}[\gamma]$ for the wire swap gives a nonreal value—note that we must first normalize the swap gate to determinant 1—hence it cannot be implemented with fewer than three CNOT gates, and the usual implementation is optimal.

In earlier versions of our work, we state the CNOT counting formulae without invoking the characteristic polynomial: an operator $u$ can be implemented using no CNOT gates iff $\gamma(u) = \pm I$, one CNOT iff $\gamma(u)$ is nonscalar and $\gamma(u)^2 = -I$, two CNOT gates iff $\mathrm{tr}[\gamma(u)]$, and three CNOT gates otherwise. We note that this formulation avoids any computation of eigenvalues, and so may be easier to use in practice. On the other hand, to actually determine the one-qubit gates, one must compute eigenvalues.

## IV. SYNTHESIS ALGORITHM AND ITS VALIDATION

The results of Sec. III can be combined with the techniques of Propositions III.3 and II.1 and the published literature to yield an explicit circuit synthesis algorithm:

(i) Given the matrix of a unitary operator $u \in \mathrm{U}(4)$, divide it by $\sqrt[4]{\det(u)}$ to ensure $u \in \mathrm{SU}(4)$.

(ii) Compute $\chi[\gamma(u)]$ to determine whether $u$ requires zero, one, two, or three CNOT gates.

(iii) If $u$ requires zero or one CNOT gates, use the techniques of the proof of Proposition VI.1 to determine which one-qubit operators are required.

(iv) If $u$ requires two CNOT gates, find the roots of $\chi[\gamma(u)]$ and determine the $\delta,\phi$ of Proposition III.3. Then use

the methods of Proposition VI.1 to determine what one-qubit gates are required at the ends of the circuit.

(v) Finally, if $u$ requires three CNOT gates, apply the methods of the literature [5].

By construction, the algorithm produces CNOT-optimal circuits in all cases. It also outperforms those in [3–5,10] in important special cases, as shown below.

*Example 1.* Many quantum algorithms, notably Grover's quantum search [15] and Shor's number factoring [16], use the operator $u=H\otimes H$ to create superpositions. Computing $\gamma(u)$ allows our synthesis algorithm to recognize that $u$ admits a quantum circuit containing no CNOT gates. ■

This example is less trivial than it seems: while writing $u=H\otimes H$ makes it obvious that $u$ requires no CNOT gates, a synthesis procedure will not receive an input of $u=H\otimes H$, but rather of the $4\times 4$ matrix corresponding to $u$. It is not *a priori* clear that any worst-case CNOT-optimal circuit decomposition will implement $u$ without CNOT gates. However, several previously published algorithms do. For the next example, previous two-qubit synthesis techniques produce circuits with more CNOTs than necessary.

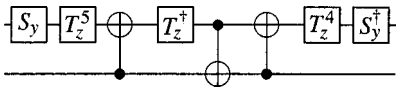
*Example 2.* The operator $u$ that swaps $|00\rangle\leftrightarrow|01\rangle$ while fixing $|10\rangle$ and $|11\rangle$ plays a prominent role in the Deutsch-Josza algorithm [1,17]. Note that $C_2^1(I\otimes\sigma_x)$ simulates $u$. Computing $\gamma(e^{i\pi/4}u)$ reveals that $u$ requires only one CNOT. However, depending on certain algorithmic choices, anywhere from one to four one-qubit gates could appear. In any event, this compares favorably to previous work [5] which synthesizes a circuit with two CNOTs and five one-qubit gates. ■

For further optimization, we may fine tune the algorithmic choices mentioned above. First, as the two CNOT gates $C_1^2$ and $C_2^1$ differ only by one-qubit gates, they are equivalent from the perspective of our methods. However, the number of one-qubit gates present in the resulting circuit depends on which of these is chosen. This is a finite problem: at most three CNOT gates appear and thus there are at most eight possibilities, so we simply run through them all. Additional degrees of freedom arise in finding a circuit that computes a given $v$ using a given $u$ and one-qubit operators, when this is possible. The proof of Proposition 1 describes an algorithm for this and requires picking a basis of eigenvectors for a certain matrix. If the eigenvalues are distinct, the only degree of freedom is the ordering of the basis of eigenvectors ($4!=24$ possibilities). However, repeated eigenvalues allow more flexibility in choosing basis vectors and potentially nontrivial circuit optimizations.

*Example 3.* At the heart of Shor's factoring algorithm [16] is the quantum Fourier transform [1]. On two qubits, it is given by the matrix

$$\mathcal{F}=\frac{1}{2}\begin{pmatrix}1 & 1 & 1 & 1\\ 1 & i & -1 & -i\\ 1 & -1 & 1 & -1\\ 1 & i & -1 & i\end{pmatrix}.$$

Explicit computation of $\chi[\gamma(\mathcal{F})]$ reveals that two CNOT gates do not suffice to simulate $\mathcal{F}$. Thus the following circuit to compute $\mathcal{F}$ is CNOT optimal:




Above, $T_z=e^{-i\sigma^z\pi/8}$ and $S_y=e^{-i\sigma^y\pi/4}$. Note that this circuit requires only three one-qubit gates, although two of these have been broken up for clarity. In fact, we can show that the number of one-qubit gates is optimal as well. For suppose two or fewer one-qubit gates sufficed. Any placement of two one-qubit gates in a three CNOT circuit will either leave one exposed CNOT at an end, or three CNOT gates together in the middle. In the second case, either two of the CNOT gates would cancel, or the three CNOT gates would reduce to a swap. Thus if $\mathcal{F}$ could be implemented using three CNOT gates and two or fewer one-qubit gates, then either one of $\mathcal{F}C_1^2$, $\mathcal{F}C_2^1$, $C_1^2\mathcal{F}$, $C_2^1\mathcal{F}$ could be implemented using two CNOT gates, or $\mathcal{F}\cdot X_{\mathrm{SWAP}}$ could be implemented using none. We use Propositions 2 and 4 to check that this is not the case.

It is common in the literature to give a circuit diagram for the $\mathcal{F}$ containing only two CNOT gates. We point out that these circuit diagrams contain an implicit wire swap at the end. One could determine *a priori* that $\mathcal{F}$ could be simulated by a circuit of this form by applying Proposition 4 to $\mathcal{F}\cdot X_{\mathrm{SWAP}}$.

## V. TIMING A HAMILTONIAN TO COMPUTE CNOT

In this section, we note that an important application Zhang *et al.* ([12], Sec. V) may be realized without specialty software. Their work discusses timing Hamiltonians of given physical systems—e.g., *XY* and Josephson Hamiltonians—to compute target computations modulo local operators. Typical targets include CNOT and $\sqrt{\mathrm{SWAP}}$. This is accomplished by using specialty numerical software or in certain cases direct analytic solutions to trace the equivalence class within a certain Weyl chamber of $\exp(itH)$ as $t$ varies. It is possible to associate such a Weyl chamber point with the roots of the characteristic polynomial computed above. This will not be done explicitly here. Rather, we note that by computing the characteristic polynomial, one may use standard matrix software (e.g., MATLAB) to determine which time for a given $H$ produces some $t_c$ with $X_{\mathrm{CNOT}}=(a\otimes b)\exp(it_cH)(c\otimes d)$, if any. Indeed, the algorithm is simply to numerically compute a large sequence of $v(t)=\exp(itH)$ for small time steps and then test each against Proposition 3.

Since earlier work [12] contains several physical examples, we illustrate our method with a simple example. For the following Hamiltonian $H_{42}$, note that the second term is well known to admit a time (e.g., $\pi/4$) for which it computes CNOT:

$$H_{42}=(0.42)I\otimes\sigma^z+\sigma^x\otimes\sigma^x.$$

Yet if the first one-qubit term of the Hamiltonian may not be switched off, this presents a problem. For the terms do not commute; one may not simply factor out $\exp[i(I\otimes\sigma^z)t]$. Excluding Zhang *et al.*, existing techniques resort to Trotterization, which implements $\exp(A+B)$ by separately turning on $A$ and $B$ for short periods of time. Below we find a simpler, direct implementation of CNOT from $H_{42}$. It is especially in-

teresting in light of concerns about the scalability of Trotterization [18].

We compute $\gamma(e^{iH_{42}t})$ for uniformly spaced trial values of $t$ and seek out those values at which the characteristic polynomial nears $p(x)=(x^2+1)^2=x^4+2x^2+1$. Out implementation in C++ finds $t_{\mathrm{CNOT}}=0.80587$ in 20 s on a common workstation. Hence, we produce a CNOT from $H_{42}$ and one-qubit gates without Trotterization. Specifically, since $e^{iH_{42}t_{\mathrm{CNOT}}}$ implements $C_2^1$ up to one-qubit operators, we use the technique of Proposition 5 to compute the relevant one-qubit operators. We find that the matrices

$$a_2=\frac{1}{2}\begin{pmatrix}1-i & -1+i\\ 1+i & 1+i\end{pmatrix},\quad c_2=0.707107\begin{pmatrix}-1 & -1\\ 1 & -1\end{pmatrix},$$

$$b_2=\begin{pmatrix}-0.21503-0.976607i & 0\\ 0 & -0.21503+0.976607i\end{pmatrix},$$

$$d_2=\begin{pmatrix}0.152049+0.690566i & -0.690566-0.152049i\\ 0.690566-0.152049i & 0.152049-0.690566i\end{pmatrix}$$

satisfy $C_2^1=(a_2\otimes b_2)e^{iH_{42}t_{\mathrm{CNOT}}}(c_2\otimes d_2)$ with a numerical precision of $10^{-6}$.

Further numerical experiments suggest that building a CNOT is possible whenever 0.42 is replaced by a weight $w$, $0\leqslant w\leqslant 1$. However, we have no analytical proof of this. Numerical experiments also suggest the *impossibility* of timing the Hamiltonian $H_{XYZ}=\sigma^x\otimes\sigma^x+\sigma^y\otimes\sigma^y+\sigma^z\otimes\sigma^z$ so as to compute a CNOT. In other words, trying values of $t$ in the range $-10\leqslant t\leqslant 10$ as above produced no candidate durations.

## VI. CONCLUSIONS AND FUTURE WORK

Our work addresses small-circuit structure in two-qubit unitary operators. In particular, we contribute tests for such structure, and our techniques can be viewed as algorithms for finding small circuits when they exist. We detail such an algorithm that produces the minimal possible number of CNOT gates (zero, one, two, or three) *for each input*. It is illustrated on circuit examples derived from well-known applications.

The one-CNOT test has an additional use. Given a two-qubit Hamiltonian $H$ that one can time to realize the CNOT, we can find the required duration. In other words, if $e^{itH}$ is the CNOT up to local unitaries, we can find $t$.

## ACKNOWLEDGMENTS

This work is supported by the DARPA QuIST program and an NSF grant.

## APPENDIX

*Proposition 5.* Let $\gamma:\mathrm{SU}(4)\to\mathrm{SU}(4)$ be given by the formula $u\mapsto u(\sigma^y)^{\otimes 2}u^T(\sigma^y)^{\otimes 2}$. Then for $u,v\in\mathrm{SU}(4)$, $u\equiv v\Leftrightarrow\chi[\gamma(u)]=\chi[\pm\gamma(v)]$.

*Proof.* By definition, $u\equiv v\Leftrightarrow u=(a\otimes b)\lambda v(a'\otimes b')$ for some one-qubit operators $a,b,a',b'$ and some scalar $\lambda$. Requiring $u,v\in\mathrm{SU}(4)$ implies $\lambda=\pm 1,\pm i$. We show below that $u=(a\otimes b)v(a'\otimes b')\Leftrightarrow\chi[\gamma(u)]=\chi[\gamma(v)]$; the proposition then follows from the fact that $\gamma(iu)=-\gamma(u)$.

We recall that there exist $E\in\mathrm{SU}(4)$ such that $E\ \mathrm{SO}(4)E^\dagger=\mathrm{SU}(2)^{\otimes 2}=\{a\otimes b:a,b\in\mathrm{SU}(2)\}$. Such matrices are characterized by the property that $EE^T=-\sigma^y\otimes\sigma^y$. This and related issues have been exhaustively dealt with in several papers [19–22].

The property $\chi[\gamma(u)]=\chi[\gamma(v)]$ is not changed by replacing $\gamma$ with $E^\dagger\gamma E$. Using the fact $\sigma^y\otimes\sigma^y=EE^T=(EE^T)^\dagger$, we compute $E^\dagger\gamma(u)E=E^\dagger uEE^Tu^TE^{T\dagger}E^\dagger E=(E^\dagger uE)(E^\dagger uE)^T$

By making the substitution $u\mapsto EuE^\dagger$; it suffices to prove that for $u,v\in\mathrm{SU}(4)$, there exists $x,y\in\mathrm{SO}(4)$ such that $xuy=v$ iff $\chi[uu^T]=\chi[vv^T]$. Here SO(4) are the real matrices within SU(4).

Note that for $P$ symmetric unitary, $P^{-1}=\overline{P}$; hence, $[P+\overline{P},P-\overline{P}]=0$. It follows that the real and imaginary parts of $P$ share an orthonomal basis of eigenvectors. As they are moreover real symmetric matrices, we know from the spectral theorem that their eigenvectors can be taken to be real. Thus there exists $q\in\mathrm{SO}(4)$ such that $quu^Tq^\dagger$ is diagonal. By reordering (and negating) the columns of $q$, we can reorder the diagonal elements of $quu^Tq^\dagger$ as desired. Thus, if $\chi[uu^T]=\chi[vv^T]$, we can find $q,r\in\mathrm{SO}(4)$ such that $quu^Tq^T=rvv^Tr^T$ by diagonalizing both; then, $(v^\dagger r^T qu)(v^\dagger r^t qu)^T=I$. Let $s=v^\dagger r^T qu\in\mathrm{SO}(4)$. We have $q^Trvs=u$, as desired. ■


[1] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, England, 2000).

[2] D. P. DiVincenzo, Phys. Rev. A **51**, 1015 (1995).

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

[4] F. Vatan and C. Williams, Phys. Rev. A **69**, 032315 (2004).

[5] V. V. Shende, I. L. Markov, and S. S. Bullock, Phys. Rev. A **69**, 062321 (2004).

[6] J. Zhang, J. Vala, Sh. Sastry, and K. B. Whaley, Phys. Rev. A **69**, 042309 (2003).

[7] A. Barenco *et al.*, Phys. Rev. A **52**, 3457 (1995).

[8] M. J. Bremner, C. M. Dawson, J. L. Dodd, A. Gilchrist, A. W. Harrow, D. Mortimer, M. A. Nielsen, and T. J. Osborne, Phys. Rev. Lett. **89**, 247902 (2002).

[9] D. Wineland, C. Monroe, W. Itano, D. Leibfried, B. King, and D. Meekhof, J. Res. Natl. Inst. Stand. Technol. **103**, 259 (1998).

[10] S. S. Bullock and I. L. Markov, Phys. Rev. A **68**, 012318 (2003).

[11] G. Song and A. Klappenecker, Quantum Inf. Comput. **3**, 139 (2003).

[12] J. Zhang *et al.*, Phys. Rev. A **67**, 042313 (2003).

[13] Yu. Makhlin, Quantum Inf. Comput. **1**, 234 (2002).
[14] S. Bullock and G. Brennen, e-print quant-ph/0309104.
[15] L. Grover, Phys. Rev. Lett. **79**, 325 (1997).
[16] P. W. Shor, SIAM J. Comput. **26**, 1484 (1997).
[17] D. Deutsch and R. Josza, Proc. R. Soc. London, Ser. A **439**, 553 (1992).
[18] A. M. Childs, H. L. Haselgrove, and M. A. Nielsen, Phys. Rev. A **68**, 052311 (2003).
[19] C. Bennett *et al.*, Phys. Rev. A **54**, 3824 (1996).
[20] S. Hill and K. Wooters, Phys. Rev. Lett. **78**, 5022 (1997).
[21] N. Khaneja, R. Brockett, and S. J. Glaser, Phys. Rev. A **63**, 032308 (2001).
[22] M. Lewenstein *et al.*, Phys. Rev. A **63**, 044304 (2001).