Asymmetric quantum error-correcting codes

Lev Ioffe[1] and Marc Mézard[2]
[1]*Center for Materials Theory, Department of Physics and Astronomy, Rutgers University, 136 Frelinghuysen Road, Piscataway, New Jersey 08854, USA*
[2]*CNRS; Université Paris-Sud, UMR 8626, LPTMS, Orsay Cedex, F-91405 France*
(Received 12 June 2006; published 29 March 2007)

The noise in physical qubits is fundamentally asymmetric: in most devices, phase errors are much more probable than bit flips. We propose a quantum error-correcting code that takes advantage of this asymmetry and shows good performance at a relatively small cost in redundancy, requiring less than a doubling of the number of physical qubits for error correction. This code is particularly adapted for building an efficient quantum memory.

DOI: 10.1103/PhysRevA.75.032345 PACS number(s): 03.67.Hk, 03.67.Lx

I. INTRODUCTION

The quest for a quantum computer has stimulated a lot of interesting developments in recent years. However, despite remarkable progress, none of the physical devices realized so far allows the building of even a very small computer. One crucial aspect is noise control. Quantum computing faces two antagonistic constraints: one should be able to manipulate and address the results of a computation, and at the same time one must keep the noise level low. While some hardware architecture may help to achieve this compromise, it is clear now that there will never exist a quantum computer without efficient quantum error correction (QEC).

The basic principles of QEC have been written down in [1–4], and a number of QEC codes have been developed since then [5,6]. However, most of them require in practice a high level of redundancy (in coding language, a low rate): the number of physical qubits needed to effectively protect one logical qubit is large. In this paper we introduce a family of QEC codes that reach good performance with much less redundancy (typically $\leqslant 2-3$ physical qubits for one logical qubit). This is achieved by using information blocks of large size, and by exploiting the fundamental asymmetry of the noise in physical devices.

Classical coding theory shows that performance is improved by using large information blocks: in this limit, the uncorrected errors correspond to a correlated flip of a large number of physical bits, the probability of which gets exponentially small. Ideal classical codes (which get close to the Shannon limit) are found in the limit of infinite block length. Using large blocks in QEC is difficult because of the possible appearance of computation errors in the decoding. To avoid them, an efficient error-correction scheme should involve a relatively small $[o(N)]$ readout operations on each bit. Our code achieves this by using the so-called low-density parity check (LDPC) classical codes. These codes, based on old ideas of Gallager [7], have been shown recently to be very efficient in terms of performance, and they can be decoded with a small number of operations [8–10].

The generalization of these classical schemes for quantum error correction is made very difficult by the requirement that a quantum scheme should correct two types of errors: bit flips as well as phase errors. So far, the main attempt at finding such codes is the work [11]. It uses so-called self-dual codes which are tailored to deal with a noise that is symmetric in all channels. We argue that, in the physical devices conceived so far, the noise is typically asymmetric (a phase error is much more probable than a bit flip), and one can exploit this asymmetry to develop more efficient QEC codes. The construction that we propose here makes use of two standard classical codes which are among the most efficient ones: it handles the relatively rare bit errors through a Bose Ray-Chaudhuri Hocquenghem (BCH) code [12] and the more frequent phase errors through a LDPC code.

II. PHYSICAL NOISE

The level of the noise in a single physical bit is conveniently characterized by the relaxation time T_1 and dephasing time T_2, the two parameters that enter Bloch equation for a single-bit (spin) dynamics. Because relaxation always implies dephasing, the dephasing rate $1/T_2$ has a contribution from the relaxation processes and a pure dephasing: $1/T_2 = 1/(2T_1) + \Gamma_\phi$. Generally, there are many ways to control the relaxation rate. First, the relaxation between two states with energy difference ΔE requires a transfer of energy to the environment, the amplitude of which becomes smaller as $\Delta E \to 0$. Furthermore, in many physical implementations these two states are separated by a large barrier that makes transitions between them rare. The situation is completely different with the dephasing rate Γ_ϕ which is physically due to the fluctuations of ΔE with time. All low-frequency processes contributing to the $\Delta E(t)$ dependence result in a decrease of the $\langle \exp[-i\int \Delta E(t)dt] \rangle$ correlator, i.e., they lead to the dephasing. In this respect, a particularly damaging effect comes from omnipresent $1/f$ noise. Thus, it is not surprising that, in almost all devices studied so far, the relaxation rate can be made much slower than the dephasing: in a typical NMR device $T_1 \sim 10-100$ s while $T_2 \sim 1$ s [16], in superconducting phase qubits $T_1 \sim 10$ μs while $T_2 \sim 100$ ns [17], in superconducting charge qubits $T_1 \sim 100$ ns while $T_2 \sim 1$ ns [18], and finally for spin dots $T_1 \sim 1$ μs while $T_2 \sim 10$ ns [19,20]. In atomic and ionic systems the main source of errors is due to dephasing, as well. For instance, in ions confined to microtraps it originates from the motion of the charges trapped in the insulator [21], similarly to the mecha-

nism responsible for the charge noise in larger superconducting qubits.

In the following we shall therefore assume that in physical qubits the noise is strongly asymmetric. Specifically, we study a noise channel defined as follows. Noise acts independently on each bit. It induces a bit flip with probability p_x, and independently it induces a phase flip with probability p_z. The original state of the system, $|\psi_0\rangle$, is thus changed to $|\psi\rangle=\Pi_i[(\sigma_z^i)^{m_i}(\sigma_x^i)^{n_i}]|\psi_0\rangle$ with probability $p_z^{\Sigma_i n_i}(1-p_z)^{N-\Sigma_i n_i}p_x^{\Sigma_i m_i}(1-p_x)^{N-\Sigma_i m_i}$, where $m_i,n_i\in\{0,1\}$. The channel acts on bit i by applying an operator $U_i\in\{\mathcal{I},\sigma_x^i,\sigma_z^i,\sigma_x^i\sigma_z^i\}$.

III. CONSTRUCTION OF THE CODE

A. Calderbank-Shor-Steane codes

Our family of codes is of the Calderbank-Shor-Steane (CSS) type [3,4]. It consists of two independent encoding and decoding devices dealing separately with bit and phase flips, for a string of N physical qubits. It uses M_z z checks and M_x x checks. The ath z check is defined by a set $V(a)\in\{1,\ldots,N\}$ and by the operator $C_a^z=\Pi_{i\in V(a)}\sigma_i^z$. Similarly, the ath x check is defined by a set $W(a)\in\{1,\ldots,N\}$ and by the operator $C_a^x=\Pi_{i\in W(a)}\sigma_i^x$.

By construction, the z and x checks all commute with each other, and the original state $|\psi_0\rangle$ is an eigenstate of all the operators $C_a^z,C_{a'}^x$ with eigenvalue 1. As U_i either commutes or anticommutes with these check operators, the noise-perturbed state $|\psi\rangle$ is an eigenstate of the operators $C_a^z,C_{a'}^x$. The decoding operation consists of three steps: (i) measure the eigenvalues of the check operators, (ii) infer from these eigenvalues what was the corrupting operator, and (iii) apply the correction operator.

Step (i). The ath z syndrome is defined as the number $u_a\in\{0,1\}$ such that $C_a^z|\psi\rangle=(1-2u_a)|\psi\rangle$. Similarly, the ath x syndrome is defined as the number $v_a\in\{0,1\}$ such that $C_a^x|\psi\rangle=(1-2v_a)|\psi\rangle$.

Step (ii). From the z syndromes $\{u_a\}$, $a\in\{1,\ldots,M_z\}$, we compute N numbers $\{m_1',\ldots,m_N'\}$ such that, for each $a\in\{1,\ldots,M_z\}$, $\Sigma_{i\in V(a)}m_i'=u_a(\mathrm{mod}2)$, with the smallest possible number of m''s equal to 1. From the x syndromes $\{v_a\}$, $a\in\{1,\ldots,M_x\}$, we compute N numbers $\{n_1',\ldots,n_N'\}$ such that, for each $a\in\{1,\ldots,M_x\}$, $\Sigma_{i\in W(a)}n_i'=v_a(\mathrm{mod}2)$, with the smallest possible number of n''s equal to 1.

Step (iii). Generate $|\psi'\rangle=\Pi_{i=1}^N[(\sigma_x^i)^{n_i'}(\sigma_z^i)^{m_i'}]|\psi\rangle$. If the error correction is successful, one should find $|\psi'\rangle=|\psi_0\rangle$.

A CSS code is thus characterized by the sets $V(a)$ and $W(a)$ defining the checks. In building such a code, one must ensure that all check operators commute. This imposes that, $\forall\ a\in\{1,\ldots,M_z\}$, $\forall\ a'\in\{1,\ldots,M_x\}$, $|V(a)\cup W(a')|$ be even. It is useful to define the parity check matrices of the two codes. The matrix H^z is an $M_z\times N$ matrix with entries in $\{0,1\}$, defined by $H_{ai}^z=1$ if and only if $i\in V(a)$. Similarly, H^x is the $M_x\times N$ matrix defined by $H_{ai}^x=1$ if and only if $i\in W(a)$. The commutativity condition is satisfied when $H^z(H^x)^T=0$ [using Boolean algebra, i.e., mod(2) additions]. The z codewords are strings of N bits $x_i\in\{0,1\}$ such that, $\forall\ a$, $\Sigma_i H_{ai}^z x_i=0(\mathrm{mod}2)$. Any x check a defines a z codeword through $x_i=1$ if $i\in W(a)$, and $x_i=0$ otherwise. Similarly, z checks define x codewords. Most of the research on QEC so far has focused on the design of relatively small codes with good distance properties. If, for instance, all pairs of x codewords are at a Hamming distance $\geqslant 2d+1$, the code will correct any set of $\leqslant d$ flip errors. While this suggests building codes that maximize the smallest distance between codewords, this strategy is not necessarily optimal when dealing with large block length ($N\gg 1$). Instead, what is practically required is that the probability of an error is small and it turns out that the best classical codes often have (rare) pairs of codewords that are close to each other [10]. We shall use this approach to construct our x checks.

B. z checks: BCH code

Our z code is an efficient classical construction, a binary primitive BCH code (see Ref. [13] for an extended presentation). The code depends on two parameter m,t. The first one determines the Galois field $GF(2^m)$ that is used, and the number t is equal to the number of errors (bit flips) that the code can correct. The number of variables (and therefore the number of qubits) is given by $N=2^m-1$. If α is a primitive element of the field $GF(2^m)$, the powers $\alpha^r, r\in\{1,\ldots,N\}$, are N distinct elements of the field, building a cyclic group under multiplication. At the same time, $GF(2^m)$ is a vector space of dimension m over $GF(2)$: every element α^r can be decomposed as $\alpha^r=\sum_{p=0}^{m-1}\gamma_{rp}\alpha^p$, where the coefficients γ are in $\{0,1\}$. The check matrix H of the code is defined as

$$H=\begin{bmatrix}1 & \alpha & \alpha^2 & \ldots & \alpha^{N-1}\\ 1 & (\alpha^3) & (\alpha^3)^2 & \ldots & (\alpha^3)^{N-1}\\ 1 & (\alpha^5) & (\alpha^5)^2 & \ldots & (\alpha^5)^{N-1}\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ 1 & (\alpha^{2t-1}) & (\alpha^{2t-1})^2 & \ldots & (\alpha^{2t-1})^{N-1}\end{bmatrix}. \quad (1)$$

This matrix can be seen as a $t\times N$ matrix with elements in $GF(2^m)$, but another interpretation is also useful. If we write each element α^r of H as the m-component vector

$$\begin{pmatrix}\gamma_{r0}\\ \vdots\\ \gamma_{r(m-1)}\end{pmatrix},$$

we obtain the $tm\times N$ parity check matrix H^z with entries in $GF(2)=\{0,1\}$. Therefore $M_z=tm$. BCH decoding relies on algebraic properties which are most easily written in terms of polynomials. Here we shall just present the basic result in the case $t=2$. If two of the N bits are flipped by noise, and these indices correspond to the elements of $GF(2^m)$ called β_1,β_2, the check matrix H, applied to the error vector, gives two syndromes $\zeta_1=\beta_1+\beta_2$ and $\zeta_3=\beta_1^3+\beta_2^3$. Decoding consists in finding β_1,β_2 given ζ_1,ζ_2. It is easily seen that this system has a unique solution in $GF(2^m)$ (up to the permutation of β_1 and β_2): the code with $t=2$ corrects exactly any set of $\leqslant 2$ errors. The same construction works for arbitrary t, and good

decoding algorithms exist: the code corrects any set of $\leqslant t$ errors. In practice we have used the Berlekamp algorithm [13], adapting some software available from [15].

C. Generation of the x checks: LDPC code

Some BCH codes are self-dual; in such a case one gets a quantum code using $H^x = H^z$ [14]. But in order to get a much better performance (for large N) on the x channel, we prefer to use a code as close as possible to the random LDPC codes. The commutation of the x and z checks is obtained by the following procedure. Given a BCH code with parameters m,t, we can generate an x check a with any degree $n \geqslant 2t+1$ using a variant of its standard decoding algorithm. The first $n-t$ elements of $W(a)$ are chosen as a random subset of $\{1,\ldots,N\}$ with distinct elements, taken uniformly among all such subsets. Let us call $\beta_1,\ldots,\beta_{n-t}$ the corresponding elements of $\mathcal{F}_G(2^m)$. We look for the remaining t elements which are solutions of the decoding equations

$$\sum_{r=1}^{t} (\beta_{n-t+r})^{2s-1} = -\sum_{r=1}^{n-t} (\beta_r)^{2s-1} \quad \forall\, s \in \{1,\ldots,t\}.$$

Because all $t!$ permutations of the solution elements $\{\beta_{d-t+1},\ldots,\beta_d\}$ lead to the same t elements on the right-hand side, one expects that the solution exists with probability $1/t!$, which was confirmed numerically. Provided that it exists, the elements $\beta_{d-t+1},\ldots,\beta_d$ can be found using any standard BCH decoding algorithm, like Berlekamp one. The indices corresponding to the elements $\beta_1,\ldots,\beta_{n-t},\ldots,\beta_n$ form the subset $W(a)$ defining the ath x check. As $\beta_1,\ldots,\beta_n$ is a codeword of the BCH code, the commutativity condition is satisfied.

Clearly, the indices in $V(a)$ do not form a random subset of size n. However, if the map used in generating $\beta_{n-t+1},\ldots,\beta_n$ from $\beta_1,\ldots,\beta_{n-t}$ is chaotic enough (we shall refer to this hypothesis as the "chaos hypothesis" in the following), one can hope to generate a set of x checks with performances close to the ones of classical random LDPC codes. This is what we have found numerically. In practice, for a given value of t, we generate with this procedure a large enough pool of $\mathcal{S} \gg 1$ possible z checks, all having degree $n=2t+1$. From this pool, we select a number M_z of checks in such a way that the degrees of the variables in the corresponding factor graph have a narrow distribution. This is done by the following inductive procedure. Suppose we have already selected $r < M_z$ checks. For this system of checks, we compute the degree of each variable. We then look at the $\mathcal{S}-r$ remaining checks in the pool, and compute for each of them its "quality," defined as the number of minimal degree variables that would be affected by addition of this check. We then add one (randomly chosen) check of the highest quality and repeat the procedure.

The practical decoding of our LDPC code uses the standard belief propagation (BP) algorithm [8,9], a message-passing algorithm which is equivalent to an iterative solution of the Bethe equations.

TABLE I. Performance of various codes: The block error has been fixed to $P_{\text{block}}=10^{-4}$; p_z and p_x give the corresponding noise thresholds in the two channels.

N	m	t	p_z	M_z	R_z	p_x	M_x	R_x	R
1023	10	2	8.40×10^{-5}	20	0.980	8.4×10^{-3}	563	0.45	0.43
1023	10	3	2.26×10^{-4}	30	0.971	2.26×10^{-2}	460	0.55	0.52
1023	10	4	4.34×10^{-4}	40	0.961	4.34×10^{-2}	530	0.48	0.44
1023	10	3	2.26×10^{-4}	30	0.971	2.26×10^{-3}	460	0.55	0.52
1023	10	4	4.34×10^{-4}	40	0.961	4.34×10^{-3}	344	0.66	0.62
1023	10	5	6.98×10^{-4}	50	0.951	6.98×10^{-3}	271	0.73	0.69
1023	10	6	1.01×10^{-3}	60	0.941	1.01×10^{-2}	285	0.72	0.66
4095	12	3	5.66×10^{-5}	36	0.991	5.66×10^{-3}	1577	0.61	0.61
4095	12	4	1.08×10^{-4}	48	0.988	1.08×10^{-2}	1378	0.66	0.65
4095	12	5	1.74×10^{-4}	60	0.985	1.74×10^{-2}	1189	0.71	0.69
4095	12	6	2.52×10^{-4}	72	0.982	2.52×10^{-2}	1191	0.71	0.69

IV. PERFORMANCE

An important parameter of the code is its degree of redundancy. We have checked that the various checks are generically linearly independent, so the z rate (x rate) is obtained as $R_z = 1 - M_z/N$ ($R_x = 1 - M_x/N$) and the quantum rate of the code is $R = 1 - (M_x + M_z)/N$.

The error-correction ability depends on the channel. In the z channel (bit-flip errors), by construction, the BCH code is able to decode up to t errors. Therefore the probability of error in decoding this channel is

$$P_{\text{err}}^z = \sum_{j=t+1}^{N} \binom{N}{j} p_z^j (1-p_z)^{N-j}, \tag{2}$$

which is well approximated, for the small values of p_z that interest us here, by $1 - e^{-Np_z} \Sigma_{j=0}^{t} (Np_z)^j / j!$.

Let us now turn to the x channel. The performance of BP decoding for random LDPC codes can be studied analytically in the limit of large block length [10]. Within the chaos hypothesis, one could thus derive the threshold for zero error decoding in the large-N limit. However, in practice we are interested in not-too-large values of N. We have thus tested numerically the BP decoding of our x code.

The simulation is run as follows. We fix an "acceptable" value of the block error P_{block} for decoding N bits, both in the x and in the z channel, in practice $P_{\text{block}}=10^{-4}$. For given values of N (or m) and t, Eq. (2) gives the noise level p_z that can be corrected in the z channel, and the channel asymmetry gives the ratio p_z/p_x. We then test various x codes, varying M_x until the block error in the x channel is less than P_{block}. Results are summarized in Table I, which studies asymmetries $p_z/p_x = 0.01, 0.1$. Notice that we consider as successful only the cases in which $\forall i\ n_i' = n_i$ and $m_i' = m_i$. Therefore we compute an upper bound to the real error (because one might have $|\psi'\rangle = |\psi\rangle$ even when this condition is not realized).

We see that large enough codes provide a good performance. For instance, an $m=12, t=6$ code with $N=4095$ qubits is able to correct a noise level of $p_z = 2.5\times10^{-4}$ in the z channel and $p_x = 2.5\times10^{-2}$ in the x channel with block error

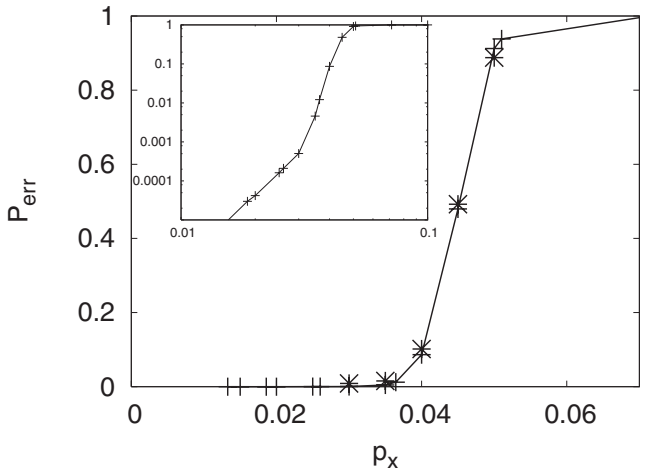

FIG. 1. Block error in the x-channel, P_{err}^{x}, versus the phase error probability p_x, for the code with $m=12$, $t=4$, $M_x=1378(+)$. The line is a guide to the eye. Also shown is the same curve for a random LDPC code ($\times$). The inset gives the same data in a log-log plot.

probability smaller than 10^{-4}. Notice that for these values of p_z, p_x, the probability of a block error *without* any error correction (i.e., the probability of at least one error somewhere in the block) would be $1-(1-p_{z,x})^N$, giving 0.63 for the z channel and 1 for the x channel. The computation of syndromes requires a number of quantum operations (application of σ_x^i or σ_z^j) per logical qubit stored equal to $N_{\text{op}}=[(m+2)t+1]/R$, while the correction of error takes $(p_x+p_z)/R$ operations per stored qubit. Figure 1 gives the block error in the x channel, P_{err}^{x}, versus the phase error probability p_x, for one given code.

V. CONCLUSIONS

We have provided an explicit construction of quantum codes with rates $R\sim 0.5$ that are able to correct a few errors in one channel (bit flips) and have close to optimal performance in another (phase errors), together with efficient decoding procedures. One important aspect of these codes is the fact that the number of quantum operations to be done to decode one given bit is much smaller than N. In the z channel this is due to the fact that we use a small value of t; in the x channel it is due to the intrinsic low density of the code. Due to this key feature, the large block length of our codes does not become a serious problem in the presence of some errors during the decoding procedure. These codes might thus be quite useful for the realistic physical implementation of a quantum memory. The next step is to apply similar ideas in fault-tolerant quantum computations, which have been using only very short block lengths so far [22]. Because the suggested code belongs to the CSS family this must be in principle possible; the challenge is to come up with an efficient procedure that does not propagate errors from the x to the z-channel.

ACKNOWLEDGMENTS

We thank J. S. Yedidia for interesting discussions. This work has been supported in part by the EC grants "Stipco," No. HPRN-CT-2002-00319, "Evergrow," No. IP 1935 in the FET-IST program, NSF Grants No. DMR-0210575, and No. ECS-0608842, and ARO Grant No. W911NF-06-1-0208. L.I. thanks LPTMS for the hospitality that made this work possible.

[1] P. W. Shor, Phys. Rev. A **52**, R2493 (1995).
[2] A. M. Steane, Phys. Rev. Lett. **77**, 793 (1996).
[3] A. R. Calderbank and P. W. Shor, Phys. Rev. A **54**, 1098 (1996).
[4] A. M. Steane, Proc. R. Soc. London, Ser. A **452**, 2551 (1996).
[5] See, e. g., A. R. Calderbank, E. M. Rains, P. W. Shor and N. J. A Sloane, IEEE Trans. Inf. Theory **44**, 1369 (1998), and references therein.
[6] A. Yu. Kitaev, Ann. Phys. (N.Y.) **303**, 2 (2003); E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, J. Math. Phys. **43**, 4452 (2002).
[7] R. G. Gallager, *Low-Density Parity-Check Codes* (MIT Press, Cambridge, MA, 1963).
[8] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms* (Cambridge University Press, Cambridge, U.K., 2003).
[9] T. Richardson and R. Urbanke (unpublished).
[10] IEEE Trans. Inf. Theory **47**(2) (2001), special issue on codes, graphs and iterative algorithms.
[11] D. J. C. MacKay, G. Mitchison, and P. L. McFadden, IEEE Trans. Inf. Theory **50**, 2315 (2004).
[12] R. C. Bose and C. R. Ray-Chaudhuri, Information and Control **3**, 68 (1960); A. Hocquenghem, Chiffres **2**, 147 (1959).
[13] S. Lin and D. J. Costello, *Error Control Coding* (Pearson Prentice-Hall, Upper Saddle River, NJ, 2004).
[14] A. M. Steane, IEEE Trans. Inf. Theory **45**, 2492 (1999).
[15] R. Morelos-Zaragoza, http://www.eccpage.com/
[16] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, Science **414**, 883 (2001).
[17] P. Bertet, I. Chiorescu, G. Burkard, K. Semba, C. J. P. M. Harmans, D. P. DiVincenzo, and J. E. Mooij, Phys. Rev. Lett. **95**, 257002 (2005).
[18] O. Astafiev, Yu. A. Pashkin, Y. Nakamura, T. Yamamoto, and J. S. Tsai, Phys. Rev. Lett. **93**, 267007 (2004).
[19] J. M. Elzerman, R. Hanson, L. H. Willems van Beveren, B. Witkamp, L. M. K. Vandersypen, and L. P. Kouwenhoven, Nature (London) **430**, 431 (2004).
[20] Y. Kato, R. C. Myers, A. C. Gossard, and D. D. Awschalom, Nature (London) **427**, 50 (2004).
[21] L. Deslauriers, S. Olmschenk, D. Stick, W. K. Hensinger, J. Sterk, and C. Monroe, Phys. Rev. Lett. **97**, 103007 (2006).
[22] J. Preskill, in *Introduction to Quantum Computation and Information*, edited by H.-K. Lo, S. Popescu, and T. Spiller (World Scientific, Singapore, 1998), Chap. 8.