Noise in Grover's quantum search algorithm

B. Pablo-Norman and M. Ruiz-Altaba
Instituto de Física, Universidad Nacional Autónoma de México, A.P. 20-364, 01000 México, D.F., Mexico
(Received 22 July 1999; published 8 December 1999)

Grover's quantum algorithm improves any classical search algorithm. We show how random Gaussian noise at each step of the algorithm can be modeled easily because of the exact recursion formulas available for computing the quantum amplitude in Grover's algorithm. We study the algorithm's intrinsic robustess when no quantum correction codes are used, and evaluate how much noise the algorithm can bear with, in terms of the size of the phone book and a desired probability of finding the correct result. The algorithm loses efficiency when noise is added, but does not slow down. We also study the maximal noise under which the iterated quantum algorithm is just as slow as the classical algorithm. In all cases, the width of the allowed noise scales with the size of the phone book as $N^{-2/3}$.

PACS number(s): 03.67.Lx

I. INTRODUCTION

There exist problems where the algorithm that solves them scales exponentially as the size of the input is increased, for example computing all possible chess games, factoring a very large number, etc. This dependence on the size makes them physically unsolvable for large enough inputs. Quantum algorithms have been invented to bypass this problem, like Shor's [1], that turns tractable the problem of factoring numbers, and Grover's [2], that improves the classical search for an item in a phone book. In fact, the classical search algorithm does not scale exponentially. Rather, it is linear in the size of the phone book; Grover's quantum algorithm improves it to a square-root dependence. Recently, an experimental application of a quantum algorithm was implemented [7], and agreement between theory and experiment was found.

Nevertheless, the strength of a quantum algorithm is also its weakness: a quantum computer performs simultaneous operations over large superpositions of states, which are very sensitive to decoherence. Fortunately, quantum correction codes have been developed [3,4] with which a quantum computer can recover from errors in the presence of moderate decoherence. But these quantum correction codes are themselves subject to decoherence, and it is not fully understood how decoherence affects the correction itself. In this work, we study the intrinsic robustness of Grover's algorithm, when quantum correction codes are not implemented.

II. GROVER'S QUANTUM SEARCH ALGORITHM

Any classical algorithm for finding an item in a randomly ordered phone book (whether deterministic or probabilistic) requires $N/2$ steps on the average, because the only way to perform the search is to analyze each item one by one until the searched–for item is found. Recently, Grover invented a quantum algorithm [2] that runs like $O(\sqrt{N})$. Let us review it briefly.

In a phone book with $N=2^n$ entries, each item can be represented by a binary label of length n or, equivalently, by a pure state of n spin-$\frac{1}{2}$ particles. The algorithm is based on constructing a coherent superposition of all these states, and repeatedly applying certain unitary transformations to it.

Assume, for concreteness, that the item we are looking for is represented by the state $|\downarrow\downarrow\cdots\downarrow\rangle$, i.e. by n spin-down particles. The algorithm works via the repeated action of the unitary steps below, starting from an initial state which we take to be the full coherent superposition of all states in the system, namely,

$$\Psi_0=\frac{1}{\sqrt{N}}\begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix}. \tag{2.1}$$

Of course, one could start equally well with some other initial state [5]. The two unitary steps to be repeated are the following

First, invert the phase of the looked–for state trough the unitary transformation

$$U_1=\begin{pmatrix}-1&0&\cdots&0\\0&1&0&\vdots\\\vdots&0&\ddots&0\\0&0&\cdots&1\end{pmatrix}. \tag{2.2}$$

Second, invert, with respect to the average, the phase of the looked-for state trough the unitary diffusion matrix

$$(U_2)_{ij}=\frac{2}{N}-\delta_{ij}. \tag{2.3}$$

These two steps are equivalent to the action of the following single unitary transformation:

$$U=U_2U_1=\frac{2}{N}\begin{pmatrix}-1+\frac{N}{2}&1&\cdots&1\\-1&1-\frac{N}{2}&1&\vdots\\\vdots&1&\ddots&1\\-1&1&1&1-\frac{N}{2}\end{pmatrix}. \tag{2.4}$$

When the unitary transformation U has been applied m times to the initial state Ψ_0, the new quantum state will be

$$\Psi_m = U^m \Psi_0 = \begin{pmatrix} A_m \\ B_m \\ \vdots \\ B_m \end{pmatrix}. \tag{2.5}$$

The action of U on the initial state Ψ_0 yields only two distinct amplitudes A_m and B_m, whereby it is possible to recast the recursion relation in just two dimensions. The restriction of U to this two−dimensional subspace will be denoted by S. Explicitly, the amplitudes A_m and B_m are given by the recursion formula

$$\begin{pmatrix} A_{m+1} \\ B_{m+1} \end{pmatrix} = \begin{pmatrix} 1-\frac{2}{N} & 2-\frac{2}{N} \\ \frac{-2}{N} & 1-\frac{2}{N} \end{pmatrix} \begin{pmatrix} A_m \\ B_m \end{pmatrix} = S \begin{pmatrix} A_m \\ B_m \end{pmatrix}$$

$$= S^{m+1} \begin{pmatrix} \frac{1}{\sqrt{N}} \\ \frac{1}{\sqrt{N}} \end{pmatrix}, \tag{2.6}$$

The two-dimensional matrix S has eigenvalues $e^{\pm i\varphi}$, with $\cos\varphi = 1-(1/N)$, whereby

$$A_m = \frac{1}{\sqrt{N}}(\cos m\varphi + \sqrt{N-1}\sin m\varphi), \tag{2.7}$$

$$B_m = \frac{1}{\sqrt{N}}\left(\cos m\varphi - \frac{1}{\sqrt{N-1}}\sin m\varphi\right). \tag{2.8}$$

From Eq. (2.7), the probability of finding the state we are looking for if we measure Ψ_m is thus

$$P(m) = |A_m|^2 = \frac{1}{N}(\cos m\varphi + \sqrt{N-1}\sin m\varphi)^2. \tag{2.9}$$

With the change of variables $\varphi = 2\theta$, $P(m)$ can be written as [6]

$$P(m) = \sin^2[\theta(2m+1)], \tag{2.10}$$

Clearly, $P(m)$ is periodic, with maxima at

$$\theta(2m+1) = n\pi, \quad n \text{ integer}, \tag{2.11}$$

The first maximum for large N is approximately at

$$m_{\max} \simeq \frac{\pi\sqrt{N}}{4} \tag{2.12}$$

and $P_{\max} = P(m_{\max}) \simeq 1$. The number of steps required to find the state with almost certainty scales like $\sqrt{N}$, as shown in Eq. (2.12).

III. MODELING NOISE IN GROVER'S ALGORITHM

As stated in Sec. I, quantum correction codes have been developed and it is supposed that in the presence of low but physically realistic levels of noise they are useful [3,4]. These codes can be implemented only if a small enough subset of the quantum computer's q bits undergo errors, and when the probability of occurrence of an error in the computation is lower than a certain bound. On the other hand, the real effect of the noise introduced by these correction codes over the original algorithm is not completely known, because they are quantum computations too. Hopefully, such errors are small and tractable. But what happens if they are not? Or, even worse, what happens if many q bits undergo errors? Is it still possible to make sense of the computation under this hypothetical noisy situation when quantum correction codes do not suffice or cannot be implemented? If it does, how much noise the algorithm can bear with on its own? We now turn to the answer to these questions.

In the particular case of Grover's algorithm, there is a simple way to model noise, because of the explicit recursion formula (2.6) for the amplitudes of the searched-for state. Suppose that in each step of the algorithm, a white or Gaussian noise modifies the state of the whole phone book according to

$$\begin{pmatrix} A_{m+1} \\ B_{m+1} \end{pmatrix} = \frac{1}{\mathcal{N}}\left[S\begin{pmatrix} A_m \\ B_m \end{pmatrix} + \begin{pmatrix} a_m \\ b_m \end{pmatrix}\right], \tag{3.1}$$

where S is defined in Eq. (2.6), and both a_m and b_m are noise, determined randomly by the standard deviation σ (common to both, for simplicity) of their Gaussian distribution. Of course, the new state Ψ_{m+1} is appropriately normalized (that's what the denominator $\mathcal{N}$ is for). Explicitly,

$$\begin{pmatrix} a_m \\ b_m \end{pmatrix} = \sqrt{-2\sigma\log x_1}\begin{pmatrix} \sin 2\pi x_2 \\ \cos 2\pi x_2 \end{pmatrix}, \tag{3.2}$$

where x_1 and x_2 are computer-generated random variables uniformly distributed over the interval [0,1]. The two Gaussian variables a_m and b_m are mutually independent, and change, randomly, from one iteration of Eq. (3.1) to the next. Note that when $\sigma = 0$, a_m and b_m are always zero, and thus there is no noise.

A crucial *caveat* is in order here: note that we introduce only two different errors, one for the searched-for state and one for all the other pure states. This approximation is physically unrealistic, but worthy of study. The full noisy situation would call for allowing N different random variables to be added independently to each of the N components of the state vector, instead of restricting ourselves to noise in the two-dimensional subspace where S (instead of U) acts.

Now we want to find the maximal allowed noise, quantified by σ, in terms of both (a) the size N of the phone book and (b) a given probability P_{cut} for finding the searched-for state after a suitable number of iterations. If we set $P_{\text{cut}} = P_{\max}$, then of course σ can only be zero. As we allow for a decreased certainty of finding the result, and thus decrease P_{cut}, the algorithm can bear with an increasing amount of

TABLE I. In the iterated quantum algorithm, for various sizes N of a phone book, the absolute maximal allowed Gaussian width $\sigma_{\max}$ of the white noise, and its statistical uncertainty (between 5% and 10%). Also shown is the (low) limiting probability P_{cut} at maximum.

N	P_{cut}	$\sigma_{\max}$	$\Delta\sigma_{\max}$
1024	0.034	2.33×10^{-3}	1.0×10^{-4}
2048	0.024	1.48×10^{-4}	4.3×10^{-5}
4096	0.017	9.03×10^{-4}	2.6×10^{-5}
8192	0.012	5.68×10^{-4}	1.6×10^{-5}
16384	0.0085	3.28×10^{-4}	1.7×10^{-5}
32768	0.0060	2.13×10^{-4}	1.7×10^{-5}
65536	0.0043	1.17×10^{-4}	1.1×10^{-5}

noise. In the absurd limit of being happy with $P_{\text{cut}}\simeq 0$, which means we will not find the result, then any amount of noise is allowed. Of course, for any given $P_{\text{cut}}>0$, a large enough noise will destroy the algorithm. In Sec. IV we establish the dependence of this maximal allowed noise, $\sigma_{\max}$, in terms of N and P_{cut}.

Computations and results

To find when the algorithm breaks down as we increase the noise, we treat the noise as a perturbation on the exact algorithm (recovered when $\sigma=0$). Beforehand, we fix the phone book's size N and the desired probability of finding the result P_{cut}.

First, we take a very small initial value of σ and evolve the initial state Ψ_0 in Eq. (2.1) according to the noisy iteration given in Eq. (3.1). After m iterations, the probability $P(m)$ of finding the result is still $|A_m|^2$, where now the amplitude A_m includes m additions of noise. It turns out that, on the average, $P(m)$ still reaches its maximum after $m_{\max}$ steps. This is a pleasant surprise. At first thought, one could have imagined that noise not only decreased $P_{\max}$ (as it does), but also slowed down the algorithm (which it does not). To maximize the likelihood of finding the result we must measure the quantum state after $m_{\max}$ iterations, with $m_{\max}$ given by the noiseless equation (2.12).

Now we compute $P_{\max}=P(m_{\max})$ and compare it with P_{cut}. If $P_{\max}$ is greater than P_{cut}, we increase the value of σ and repeat the computation, otherwise we stop (see the Appendix for details). In this way, we find the maximal σ, labeled $\sigma_{\max}$, which is the limiting noise for $P_{\max}\geqslant P_{\text{cut}}$. Because of the probabilistic nature of the computations, we repeat this computation of $\sigma_{\max}$ many times (200): the value of $\sigma_{\max}$ we exhibit is the average, with a statistical error.

We have carried out the evaluation of $\sigma_{\max}$ for seven different phone book sizes $N=2^n$ (with n from 10 to 16) and for five different values of P_{cut} (from 0.9 to 0.5 in steps of 0.1).

For fixed P_{cut}, the dependence of $\sigma_{\max}$ on N is always of the form

$$\sigma_{\max}(N,P_{\text{cut}})=\alpha(P_{\text{cut}})N^{\phi}, \tag{3.3}$$

where ϕ is a true constant, found to be

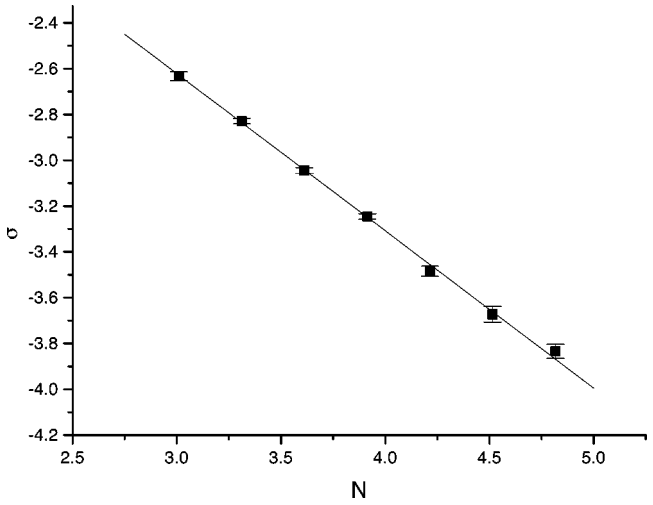

FIG. 1. Plot of log $\sigma_{\max}$ as a function of log$N$ for the iterated quantum algorithm with minimal $P_{\text{cut}}=1-0.5^{\pi/(2\sqrt{N})}$ that still improves the classical search algorithm. Even though to each N corresponds a different P_{cut}, the plot still displays the universal $N^{-2/3}$ dependence.

$$\phi=-0.696\pm0.027, \tag{3.4}$$

and α varies smoothly from 0.9 to 0.15 as P_{cut} decreases from 0.9 to 0.5 (see the Appendix).

One of our main results is that the amount of noise that the algorithm can handle decreases roughly as $N^{-2/3}$ with the size N of the list. In general, since the number of steps needed in each iteration is of the order of $N^{1/2}$, and at each step we add a noise of width σ, we expect the maximal allowed $\sigma_{\max}$ to decrease with N faster than $N^{-1/2}$. Equivalently, we expect ϕ to be smaller than $-\frac{1}{2}$. The actual value found, Eq. (3.4), satisfies this bound. We have not found a general analytic argument to pin down the actual value of ϕ.

Alternatively, keeping N fixed instead of P_{cut}, the relation between $\sigma_{\max}$ and P_{cut} can be written as

$$\sigma_{\max}(N,P_{\text{cut}})=\gamma(N)-\delta(N)P_{\text{cut}}, \tag{3.5}$$

where γ goes from 0.0024 to 0.00015, and δ from -0.0020 to 0.00013 ($\log_2 N=n=10$ and 16, respectively), with errors of about 10% (see the Appendix for details). This means that the width of the maximal white noise that may be allowed increases linearly with decreasing P_{cut}. Note that Eqs. (3.3) and (3.5) are just convenient slices of a surface in the three-dimensional space with coordinates $(N,P_{\text{cut}},\sigma_{\max})$.

IV. GROVER'S ALGORITHM IS USEFUL EVEN IF $P_{\text{cut}}<0.5$

In the derivation of the above results we exploited the experimental fact that the number of steps needed to find the searched-for state does not change when noise is present. Thus, another way to estimate the real maximal noise that the noisy Grover's algorithm can handle, while still improving the results of the classical search algorithm, is to let P_{cut} be even lower than 0.5. We now explain this.

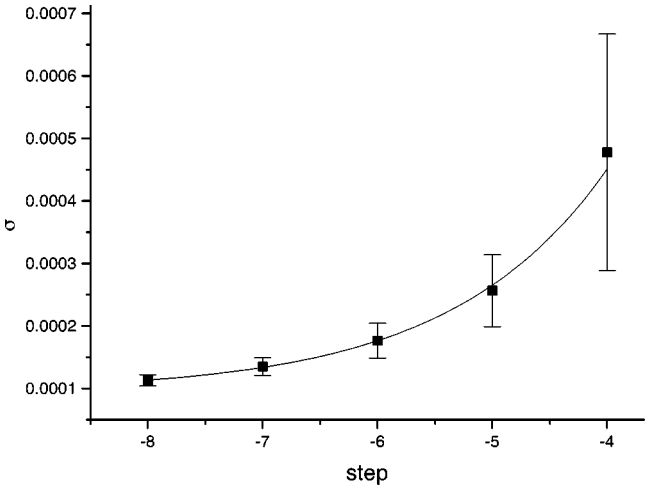

FIG. 2. Plot of $\sigma_{\max}$ as a function of $\log d\sigma$: the maximum allowed value of noise characterized by $\sigma_{\max}$ before $P_{\max} \leqslant P_{\text{cut}}$ depends on the size of the step $d\sigma$ by which σ is increased in the program. This plot is for $N=32768$ and $P_{\text{cut}}=0.7$.

Since $m_{\max}=(\pi/4)\sqrt{N}$ is always larger than $N/2$, there is an integer I_N such that $I_N m_{\max} \leqslant N/2$, namely,

$$I_N \simeq \frac{2}{\pi}\sqrt{N}. \tag{4.1}$$

Therefore, we can repeat the quantum search I_N times with a low P_{cut} such that $1-(1-P_{\text{cut}})^{I_N} \geqslant 0.5$. We are assured that we will find the searched-for state with probability $\frac{1}{2}$ in the same number of steps as the classical algorithm. Of course, the classical algorithm finds the result for sure, and compared with that finding the result only half the time is not very satisfactory. Instead of 0.5, we could equally well have chosen some other (higher) probability to be satisfied with, but we take 0.5 for definiteness as the extreme, illustrative case. The point is that the P_{cut} we need to enforce on the noisy quantum algorithm is smaller than 0.5. Note also that we are disregarding the $\log_2 N$ steps needed in each of the I_N independent iterations to prepare the initial state Ψ_0. Including them would of course lower a bit the maximal allowed noise.

The limiting probability at maximum with which the iterated quantum algorithm is as slow as the classical one is

$$P_{\text{cut}} \geqslant 1-0.5^{\pi/(2\sqrt{N})}. \tag{4.2}$$

The meaning of this is, again, that we can let P_{cut} be smaller than 0.5 for a given N because if we run I_N times the quantum algorithm with $m_{\max} \simeq \pi\sqrt{N}/4$ steps, we will find the searched-for state with a probability of at least 0.5, and the total number of steps will be less or equal to $N/2$ (ignoring the $\log_2 N$ steps required for constructing the initial state Ψ_0).

To estimate this maximal noise that the quantum algorithm can bear before it slows down all the way to equivalence with the classical one, we proceed as follows. First, we choose the size N of the list to be searched, and keep it fixed. Then, using the bound (4.2), we determine P_{cut}, which is very low. Finally, Eq. (3.5) yields $\sigma_{\max}$, which is now significantly higher. For a variety of N, our results are shown in Table I. In Fig. 1, we plot $\sigma_{\max}$ as a function of N for the data of Table I; the equation which fits it is

$$\sigma_{\max} = (0.275 \pm 0.031) N^{(-0.68 \pm 0.01)}; \tag{4.3}$$

note that the exponent of N in Eq. (4.3) is essentially the same as the exponent ϕ in Eq. (3.3), even though P_{cut} depends on N and is one or two orders of magnitude smaller than in Sec. III.

V. CONCLUSIONS

At the moment, quantum correction codes are restricted to the case when only a small enough subset of the quantum computer's q bits undergo errors, and the probability of occurrence of an error is smaller than some bound, but it is believed that quantum computations will be possible with physically realistic levels of noise even if the quantum correction codes employed undergo errors themselves.

With this in mind, we studied the intrinsic robustness of Grover's quantum search algorithm in a noisy environment. We modeled the noise with a single parameter, the width of a Gaussian distribution, and allowed for two independent noises at each step of Grover's quantum algorithm.

We found that the quantum search algorithm still reaches the maximum likelihood of finding the searched-for state in $\pi\sqrt{N}/4$ steps. The strongest effect of noise is to decrease the maximum probability from virtually 1 (the noiseless case) to lower values, depending on the size of the noise, [Eqs. (3.3) and (3.5)]. How much noise can we add to the quantum computer, with the criterion that a repeated application of the quantum algorithm, is still faster than the classical one is given by Eq. (4.2). In both cases, the allowed maximal noise decreases with the size of the phone book approximately as $N^{-2/3}$.

The presence of noise and the absence of quantum correction codes is not completely disastrous: the quantum search algorithm can handle by itself a reasonable amount of noise. Nevertheless, for large enough databases, the allowed noise becomes tiny.

ACKNOWLEDGMENTS

This work was supported in part by CONACYT Grant Nos. 25504-E and DGAPA–UNAM IN103997. B.P.N. enjoys financial support from CONACYT.

APPENDIX

The computer program we used to derive the results in Sec. III needs an initial value of σ (which we set to zero), and then computes $P(m)$. If $P(m_{\max}) > P_{\text{cut}}$, then the algorithm increases σ, repeating the process until the bound is surpassed. This gives one value for $\sigma_{\max}$. We repeat the whole story again and again and average over the values of $\sigma_{\max}$ found.

Let us illustrate our procedure with an example. Let $P_{\text{cut}}=0.7$. For each n from 10 to 16, the program increases the value of σ starting from 0 in steps of $d\sigma=0.0001$. The average maximal values of σ thus found (in 200 runs) is then $\sigma_{\max}$, shown below with its statistical uncertainty. Note that the error seems dominated by the step size:

N	$\sigma_{\max}$
1024	0.0033 ± 0.0007
2048	0.0022 ± 0.0005
4096	0.0014 ± 0.0004
8192	0.00098 ± 0.00028
16384	0.00070 ± 0.00020
32768	0.00048 ± 0.00019
65536	0.00011 ± 0.00017

(A1)

Taking a smaller step, $d\sigma = 0.00001$, we carry through the same computations and find instead

N	$\sigma_{\max}$
1024	0.0022 ± 0.0003
2048	0.0015 ± 0.0002
4096	0.00095 ± 0.00015
8192	0.00060 ± 0.00011
16384	0.00040 ± 0.00008
32768	0.00026 ± 0.00006
65536	0.00017 ± 0.00005

(A2)

Curiously, when we decrease the step both the error and the central value of $\sigma_{\max}$ decrease. This can be understood easily, since we take as value for maximal σ in each run the first σ for which the probability after $m_{\max}$ iterations is too small (smaller than 0.7 in this example), and thus we clearly underestimate it in gross dependence with the step. We are thus forced to repeat the computation of $\sigma_{\max}$ and $\Delta\sigma_{\max}$ with smaller and smaller steps, from $d\sigma = 10^{-4}$ to $d\sigma = 10^{-8}$. We must now fit the dependence of $\sigma_{\max}$ on $d\sigma$ (see Fig. 2) and extrapolate to $d\sigma = 0$.

The generic relation we found is

$$\sigma_{\max}(N, d\sigma) = \zeta(N) + \xi(N) d\sigma^{\alpha}, \tag{A3}$$

where $\alpha = 0.30 \pm 0.06$ is a true constant. The values of the N-dependent ζ and ξ are the following:

N	ζ	ξ
1024	0.00104 ± 0.00004	0.0240 ± 0.0046
2048	0.00065 ± 0.00001	0.0163 ± 0.0017
4096	0.00038 ± 0.00002	0.0114 ± 0.0027
8192	0.00023 ± 0.00001	0.0086 ± 0.0020
16384	0.00015 ± 0.00001	0.0079 ± 0.0025
32768	$0.00009 \pm 5 \times 10^{-6}$	0.0068 ± 0.0020
65536	$0.00006 \pm 4 \times 10^{-6}$	0.0076 ± 0.0034

(A4)

Taking the limit $d\sigma \to 0$, we obtain the final value of $\sigma_{\max}$ for each N at this $P_{\text{cut}} = 0.7$:

N	$\sigma_{\max}$
1024	0.00104 ± 0.00004
2048	0.00065 ± 0.00001
4096	0.00038 ± 0.00002
8192	0.00023 ± 0.00001
16384	0.00015 ± 0.00001
32768	$0.00009 \pm 5 \times 10^{-6}$
65536	$0.00006 \pm 4 \times 10^{-6}$

(A5)

The above numbers are very well fit by a straight line (in log N).

From the data of Eq. (A5), for this value of $P_{\text{cut}} = 0.7$, we finally find the relation

$$\sigma_{\max}(N, P_{\text{cut}} = 0.7) = \alpha(P_{\text{cut}}) N^{\phi}, \tag{A6}$$

with $\alpha = 0.138 \pm 0.012$, and $\phi = -0.704 \pm 0.01$.

Similarly, for other values of P_{cut} we found

P_{cut}	α	ϕ
0.5	0.158 ± 0.011	-0.687 ± 0.007
0.6	0.146 ± 0.010	-0.691 ± 0.008
0.7	0.138 ± 0.012	-0.704 ± 0.010
0.8	0.083 ± 0.006	-0.669 ± 0.008
0.9	0.094 ± 0.001	-0.724 ± 0.015

(A7)

The value quoted in the text [Eq. (3.4)] is an average of these numbers. To establish Eq. (3.5), we found for each P_{cut} a table like Eq. (A5), and then, fixing N, we found a good linear fit [Eq. (3.5)], with the following values of $\gamma(N)$ and $\delta(N)$:

N	γ	δ
1024	0.0024 ± 0.001	0.0020 ± 0.0001
2048	0.0015 ± 0.00005	0.0013 ± 0.00005
4096	0.00092 ± 0.00003	0.00077 ± 0.00003
8192	0.00057 ± 0.00002	0.00048 ± 0.00002
16384	0.00033 ± 0.00002	0.00026 ± 0.00002
13768	0.00021 ± 0.00002	0.00017 ± 0.00002
65536	0.00015 ± 0.00001	0.00013 ± 0.00001

(A8)

[1] P. W. Shor, in *Proceedings of the 35th Symposium on the Foundations of Computer Science*, edited by S. Goldwasser (IEEE Computer Society Press, New York, 1994), pp. 124–134; SIAM Comput. **26**, 1484 (1997).

[2] L. K. Grover, Phys. Rev. Lett. **9**, 325 (1997); **80**, 4329 (1998).

[3] A. R. Calderbank and P. W. Shor, Phys. Rev. A **54**, 1098 (1996).

[4] A. Steane, Proc. R. Soc. London, Ser. A **452**, 2551 (1996).

[5] E. Biham, O. Biham, D. Biron, M. Grassl, and D. Lidar, e-print quant-ph/9807027 [Phys. Rev. A (to be published)].

[6] M. Boyer, G. Brassard, P. Hoeyer, and A. Tapp (unpublished).

[7] Issac L. Chuang, Lieven M. K. Vandersypen, Xinlan Zhou, Debbie W. Leung, and Seth Loyd, Nature (London) **393**, 143 (1998).