PHYSICAL REVIEW A **75**, 052311 (2007)

Effects of imperfections for Shor's factorization algorithm

Ignacio García-Mata, Klaus M. Frahm, and Dima L. Shepelyansky*

Laboratoire de Physique Théorique, UMR 5152 du CNRS, Université Paul Sabatier, 31062 Toulouse Cedex 4, France

(Received 23 January 2007; published 9 May 2007)

We study effects of imperfections induced by residual couplings between qubits on the accuracy of Shor's algorithm using numerical simulations of realistic quantum computations with up to 30 qubits. The factoring of numbers up to $N=943$ show that the width of peaks, which frequencies allow to determine the factors, grow exponentially with the number of qubits. However, the algorithm remains operational up to a critical coupling strength ϵ_c which drops only polynomially with $\log_2 N$. The numerical dependence of ϵ_c on $\log_2 N$ is explained by analytical estimates that allows one to obtain the scaling for functionality of Shor's algorithm on realistic quantum computers with a large number of qubits.

DOI: 10.1103/PhysRevA.75.052311 PACS number(s): 03.67.Lx, 24.10.Cn, 05.45.Mt

I. INTRODUCTION

It is quite clear that the main interest to quantum computing has been generated by Shor's factorization algorithm [1] which has exponential efficiency gain compared to any known classical algorithm. Indeed, Shor's algorithm allows one to find the factors of a large number N with $O(\ln^3 N)$ quantum gates while all known classical algorithms require a number of operations that grow exponentially with $\ln N$ (see, e.g., review and relevant references in [2]). Without any doubt this result has fundamental importance from a mathematical viewpoint. However, its implementation in real life requires an understanding of effects of imperfections and errors unavoidably present in any real physical realization of the algorithm on a realistic quantum computer. Again, here a mathematician can be satisfied by a mathematical statement that in quantum computations the errors grow not faster than quadratically with the number of performed quantum gates (see [2]) and thus the global accuracy of the algorithm is good enough if the norm of errors in each quantum gate is sufficiently small. However, a physicist generally would like to see more concrete and realistic estimates of the algorithm accuracy. Unfortunately, direct experimental verification of the accuracy for a large number of gates and qubits is not possible at present. Indeed, the most advanced quantum computation of Shor's algorithm has been done on a seven-qubit NMR-based quantum computer that allowed one to factorize only a rather small number $N=15$ (even if certain simplifications of the original algorithm have been used) [3].

Therefore the only possibility remaining is the method of numerical simulations testing various types of realistic errors and imperfections. The first steps in this direction have been done in [4–6]. A number of interesting effects of errors on the accuracy of Shor's algorithm have been found in these pioneering works but the factorized number was still $N=15$ and therefore it was not possible to determine the accuracy scaling at large values of N. More recently, additional numerical studies have been performed to investigate the effects of finite accuracy in quantum phase rotations of the quantum Fourier transform (QFT) algorithm used in Shor's factorization [7,8], dynamical phase errors in Shor's algorithm with N up to 33 [9], and discrete qubit flip errors [10] with N up to 247. In the latter case the QFT part of Shor's algorithm has been performed in a semiclassical way using the one-qubit control trick (see, e.g., [11–14]) while the modular multiplication has been performed with up to 20 qubits including the workspace using the circuit described in [15].

In this work we perform extensive numerical simulations investigating effects of imperfections on the accuracy of Shor's algorithm factorizing numbers up to a maximal value $N=943$ using up to $L=30$ qubits. We concentrate our studies on the case of static imperfections which induce static one-qubit energy shifts and residual static couplings between qubits following the lines started in [16]. This type of imperfection is especially important since generally the errors produced in this case are accumulated coherently and lead to a more rapid drop of fidelity and accuracy of quantum computations compared to the cases of noisy unitary errors in quantum gates [17,18] and dissipative decoherence [19–21]. It is known that for the quantum algorithms simulating problems of quantum chaos periodic in time the Floquet eigenstates are exponentially sensitive to static imperfections [22]. Due to that the study of their effects on the accuracy of Shor's algorithm becomes especially relevant since recently it has been shown that certain blocks of the Shor algorithm are characterized by the properties of quantum chaos [23]. Also it is important to note that Shor's algorithm is essentially based on a determination of a certain frequency of return. In some cases, like in the Grover algorithm, such a frequency can be exponentially sensitive to static couplings [24] that make the investigation of static imperfections effects in Shor's algorithm even more important.

Thus in the present work we present the first numerical studies of how the static imperfections affect the accuracy of Shor's algorithm. Our aim is to determine the parametric dependence of the accuracy on the imperfection strength, number of qubits, and number of gates. For this we use a simplified but generic model of imperfections which can be applied to various implementations of Shor's algorithm discussed in the literature [5,12,14,25–30].

The paper has the following structure: Section II gives a brief description of the ideal Shor's algorithm, Sec. III describes the model of errors introduced by static imperfec-

*Electronic address: http://www.quantware.ups-tlse.fr tions, the results of numerical studies are presented in Sec. IV, and the discussion of the results is given in Sec. V.

II. IDEAL REALIZATION OF SHOR'S ALGORITHM

First we briefly describe the main structure of Shor's algorithm [1] factorizing a large integer number N using parallelism of many-body quantum evolution. Following Shor we choose a random number x relatively prime to N and calculate its *order* r (also called *period*) defined as the minimal positive integer value such that

$$x^r \equiv 1 \bmod N. \tag{1}$$

Once r is known there is a high probability of obtaining two nontrivial factors of N by a classical computation in polynomial time (in the number of binary digits of N). This procedure fails in rare cases [1] and in such a case one has simply to chose a different value of x and restart again.

The difficult task is to compute the order r and this task can be efficiently achieved by Shor's algorithm provided we have a reliable quantum computer with a sufficient number of qubits at our disposal. This algorithm requires an L-qubit state composed of two quantum registers which we will call the *control register* (with n_l qubits) and the *computational register* (with $n_q = L - n_l$ qubits). We associate to the basis states of both registers integer numbers by

$$|\tilde{l}\rangle = |\alpha_{n-1}\rangle_{n-1} \otimes \cdots \otimes |\alpha_0\rangle_0, \tag{2}$$

where in binary representation

$$\tilde{l} = \alpha_0 + 2\alpha_1 + \cdots + 2^{n-1}\alpha_{n-1} \tag{3}$$

and $n = n_l$ for the control register or $n = n_q$ for the computational register. Here $|\alpha_j\rangle_j$ represents the jth qubit of the register and $\alpha_j \in \{0,1\}$. In order to factorize a number N one needs to choose n_l and n_q such that $2^{n_q} > N$ and $Q \equiv 2^{n_l} > N^2$, therefore typically $n_l \approx 2n_q$.

We first prepare the initial state

$$|\psi_0\rangle = |0\rangle_{n_l}|1\rangle_{n_q} \tag{4}$$

and then apply single qubit Hadamard gates to every qubit in the control register and get (dropping subscripts)

$$|\psi_1\rangle = \frac{1}{\sqrt{Q}}\sum_{a=0}^{Q-1} |a\rangle|1\rangle. \tag{5}$$

The principal idea of Shor's algorithm is the observation that one can construct a combination of quantum gates, acting on both registers, that performs for all $a = 0, \ldots, Q-1$ simultaneously the operation

$$|a\rangle|1\rangle \to |a\rangle|x^a \bmod N\rangle \tag{6}$$

which gives the state

$$|\psi_2\rangle = \frac{1}{\sqrt{Q}}\sum_{a} |a\rangle|x^a \bmod N\rangle. \tag{7}$$

Then, after obtaining the state $|\psi_2\rangle$, we apply the QFT [2] to the control register

$$|\psi_3\rangle = \frac{1}{Q}\sum_{c=0}^{Q-1}\sum_{a=0}^{Q-1} e^{i2\pi ac/Q}|c\rangle|x^a \bmod N\rangle \tag{8}$$

and measure both arguments to get

$$P(c,x^k) \equiv |\langle\psi_3|c\rangle|x^k \bmod N\rangle|^2 = \left|\frac{1}{Q}\sum_{\bar{a}:x^{\bar{a}}\equiv x^r} e^{i2\pi c\bar{a}/Q}\right|^2, \tag{9}$$

where $k = 0, \ldots, r-1$ is arbitrary and the sum over $\bar{a}$ runs over all values such that $x^{\bar{a}} \equiv x^k \bmod N$. Therefore $\bar{a} = r\nu + k$ where $\nu = 0, \ldots, M_k - 1$ and $M_k \equiv [(Q-k-1)/r] + 1$ and the evaluation of the sum yields

$$P(c,x^k) = \frac{1}{Q^2}\frac{\sin^2(M_k\pi cr/Q)}{\sin^2(\pi cr/Q)}. \tag{10}$$

This function only depends weakly on the choice of k (since $Q > N^2$ and $N > r > k$ such that $Q \gg k$ and $M_k \approx Q/r \gg 1$ is nearly constant in k) and as a function of c it has r equidistant strongly localized peaks of width unity, of height $M_k^2/Q^2 \approx 1/r^2$, and located at mQ/r with $m = 0, 1, \ldots, r-1$.

If the algorithm is run by an *ideal* quantum computer, then with a very high probability the outcome of a measurement will be given by an integer value of c which is very close to one of the peaks mQ/r. Thus using a continuous fraction expansion we can determine the rational number p/q closest to c/Q with a denominator smaller than N. Here the choice $Q > N^2$ ensures that there is at most one such number inside the peak and therefore p/q coincides with m/r. Furthermore the position number m of the peak is quite random and if by chance m is relatively prime to r one obtains directly $r = q$ and the algorithm succeeds. However, if m and r have a common divisor larger than unity we have $q = r/\gcd(m,r) < r$ [where $\gcd(m,r)$ is the greatest common divisor of m and r] and the algorithm did not succeed. Therefore one has to check classically if the candidate "q" for r is indeed a solution of $x^q = 1(\bmod N)$. This fortunately can be done in a polynomial time. In case of failure the algorithm has to be repeated and even though the probability of success is not very high one obtains after a few $[O(\log\log r)]$ measurements [1] the correct value $q = r$.

Practically it is more convenient to measure only the control register which provides c with the total probability:

$$P(c) = \sum_k P(c,x^k) \approx rP(c,x^k). \tag{11}$$

We note that the dependence of M_k on k in Eq. (10) is rather weak and therefore the above procedure to determine the minimal period r remains the same.

However, this description of the algorithm still lacks some precision how to implement the operation described in Eq. (6). Suppose we are able to perform on the computational register the multiplication by $x \bmod N$:

$$|y\rangle \to U_{\text{mult}}(x)|y\rangle \equiv |(yx) \bmod N\rangle \tag{12}$$

by some unitary operator. Of course this operator cannot be unitary if we require this for all values $y = 0, \ldots, 2^{n_q} - 1$ simply because the classical application $y \to (xy) \bmod N$ is not

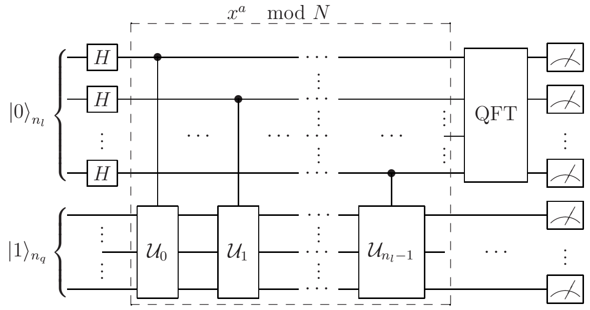

FIG. 1. Quantum circuit of Shor's algorithm on an ideal quantum computer with the quantum multiplication operator $\mathcal{U}_j = U_{\mathrm{mult}}(x^{2^j} \bmod N)$ as defined in Eq. (13).

unique on this set (unless $N=2^{n_q}$ which is of no interest). If we require that x and N are relatively prime then this application is unique at least for $y=0,\ldots,N-1$ and for $y=N,\ldots,2^{n_q}-1$ we have to complete it in some unique way, for example, by $y\rightarrow y$ if $y\geqslant N$. Therefore we define the quantum multiplication operator by x mod N by

$$U_{\mathrm{mult}}(x)|y\rangle \equiv \begin{cases} |(yx) \bmod N\rangle, & y=0,\ldots N-1 \\ |y\rangle, & y=N,\ldots,2^{n_q}-1. \end{cases} \tag{13}$$

The states $|y\rangle$ with $y\geqslant N$ are in principle not relevant for the *ideal* Shor algorithm because they are never populated in the perfect computation and the effect of the quantum gates on these states is rarely discussed in the literature [1,5,25]. However, they are important to ensure overall unitarity and they may be very well populated if the quantum computation is subjected to errors or imperfections. Furthermore, we note that in the definition (13) we could in principle replace the unit-operator acting on the nonrelevant states by an arbitrary unitary operator (acting on a space of dimension $2^{n_q}-N$) provided that we do not mix relevant ($y<N$) and nonrelevant states ($y\geqslant N$).

We now introduce the controlled multiplication operator $U_{\mathrm{Cmult}}^{(j)}(x)$ acting on both registers (control and computational register) applying the simple multiplication (13) on the computational register if and only if the jth qubit of the control register is $|1\rangle$. Developing $a=\sum_{j=0}^{n_l-1} a_j 2^j$ with $a_j\in\{0,1\}$ we see that the operation (6) can be performed by the unitary operator

$$U_{\mathrm{Fmult}}(x) = \prod_{j=0}^{n_l-1} U_{\mathrm{Cmult}}^{(j)}(x^{2^j} \bmod N) \tag{14}$$

since

$$x^a = \prod_{j=0}^{n_l-1} (x^{2^j})^{a_j} = \prod_{j=0,a_j=1}^{n_l-1} x^{2^j}, \tag{15}$$

and where in the last equation every multiplication is taken modulo N.

Figure 1 shows the schematic quantum circuit of Shor's algorithm on an ideal quantum computer in terms of the quantum multiplication operator (13). To complete an explicit implementation one has to show that this operator can be realized in terms of elementary one- or two-qubit quantum gates. We do not enter into details here and mention as examples the important works [5,25] that provided explicit implementations of the quantum multiplication by x mod N using $O(n_q^2)$ elementary gates. These implementations require also additional work space qubits which are initially $|0\cdots 0\rangle$ and must remain so after completion of this operator, i.e., the implementations must eventually provide code to reversibly "erase" the additional work space qubits. We assume that there are no errors inside this additional work space and no errors coupling it to the control and computational registers. In this way we may restrict our consideration only to $L=n_q+n_l$ qubits.

III. SHOR'S ALGORITHM WITH STATIC IMPERFECTIONS

We now turn to Shor's algorithm in the case of static imperfections [16] generated by residual couplings between qubits and energy level shifts. The effects of these imperfections and their numerical modeling have been considered in detail in [18] on examples of quantum chaos algorithms (see also references in [18] on other works). There it has been shown that effects of static residual couplings can be modeled by an additional unitary rotation acting between two arbitrary gates: $U_s=e^{i\delta H}$. Here δH represents the Hamiltonian due to the residual static couplings between qubits which provides a nontrivial evolution of the state stored in the quantum register even in the absence of any quantum gate. In this approach the quantum gates are considered to be exactly ideal. In principle δH may couple all qubits in the control register, in the computational register, and in the additional work space necessary for the concrete implementations of the quantum multiplication operator (13). However, in this work we use a simplified error model in which δH couples only the qubits in the computational register and therefore in Shor's algorithm the initial Hadamard gates or the final quantum Fourier transform are not affected by these errors. In principle the quantum Fourier transform is considered as relatively stable with respect to errors [4] and the number of Hadamard gates n_l is relatively small.

Furthermore, we do not consider a specific implementation of the quantum multiplication operator (13), we only assume that it can be written as a product

$$U_{\mathrm{mult}}(x) = U_{n_m} \times \cdots \times U_2 U_1, \tag{16}$$

where U_j, $j=1,\ldots,n_m$ are the elementary quantum gates which constitute this operator and $n_m=O(n_q^2)$ is the number of these elementary gates. A specific choice of U_j depends on the classical variable x, and also on N, and since the value of x significantly affects the algorithm implementation we have a different set of gates U_j for each x (and N).

Thus in the presence of static imperfections the quantum multiplication operator $\tilde{U}_{\mathrm{mult}}(x)$ has the form

$$\tilde{U}_{\mathrm{mult}}(x) = U_{n_m} e^{i\delta H} \times \cdots \times U_2 e^{i\delta H} U_1 e^{i\delta H}. \tag{17}$$

We now introduce an effective perturbation operator for the full multiplication operator by

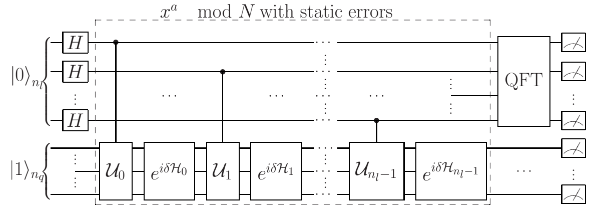
x^a mod N with static errors
$|0\rangle_{n_l}$
H
H
H
$|1\rangle_{n_q}$
$\mathcal{U}_0$
$e^{i\delta\mathcal{H}_0}$
$\mathcal{U}_1$
$e^{i\delta\mathcal{H}_1}$
$\mathcal{U}_{n_l-1}$
$e^{i\delta\mathcal{H}_{n_l-1}}$
QFT

FIG. 2. Quantum circuit of Shor's algorithm on a quantum computer with static imperfections in the quantum multiplication operator $\mathcal{U}_j = U_{\text{mult}}(x^{2^j} \bmod N)$ [Eq. (13)] and the effective static perturbation $\delta\mathcal{H}_j = \delta H_{\text{eff}}(x^{2^j} \bmod N)$ [see Eqs. (13) and (19)], where in numerical simulations $\delta\mathcal{H}_j$ is given by Eq. (21) with random realizations of δ_i, J_i for practically each j (see text).

$$\tilde{U}_{\text{mult}}(x) = U_{\text{mult}}(x)e^{i\delta H_{\text{eff}}(x)}. \tag{18}$$

From Eq. (17) we may determine $\delta H_{\text{eff}}(x)$ as

$$e^{i\delta H_{\text{eff}}(x)} = e^{i\delta H(n_m-1)} \times \cdots \times e^{i\delta H(1)}e^{i\delta H} \tag{19}$$

with

$$\delta H(j) = U_{j-1}^{-1} \times \cdots \times U_1^{-1}\delta H U_1 \times \cdots \times U_{j-1}. \tag{20}$$

We mention that the precise relation between $\delta H_{\text{eff}}(x)$ and δH is not really important in our approach since we directly model $\delta H_{\text{eff}}(x)$ in our numerical simulations and use the expression (18) without entering into details of a particular implementation of $U_{\text{mult}}(x)$. We remind that in Eqs. (19) and (20) the dependence of $\delta H_{\text{eff}}(x)$ on x is given by the choice of elementary gates U_j which are changed with a change of x. A schematic quantum circuit of Shor's algorithm on a quantum computer with static imperfections in the quantum multiplication operator (13) is shown in Fig. 2.

As we already mentioned, the implementations of the quantum multiplication operator (13) described in Refs. [5,25] require additional work space qubits which are initially placed in the state $|0\cdots 0\rangle$ and are erased after the computation. The implementation of Ref. [5] contains a quantum code that erases the work space qubits correctly but only for the relevant states $|y\rangle$ with $0 \leqslant y < N$ as initial states and not for the nonrelevant states with $y \geqslant N$. For a perfect quantum computer this is of course not a problem, but when taking into account errors the nonrelevant states may be populated and different implementations, which are absolutely equivalent for the relevant states, may potentially behave quite differently with errors. Even if the particular implementation ensures that a nonrelevant state as the initial state produces a properly erased work space register, the errors may still produce nonerased contributions.

Actually the use of work space qubits implies that the notion of nonrelevant states has to be enlarged, i.e., a combined state $|y\rangle|\psi_{\text{work}}\rangle$ in the computational and work space register has to be considered as *nonrelevant* if either $y \geqslant N$ for $|\psi_{\text{work}}\rangle = |0\cdots 0\rangle$ or y arbitrary for $|\psi_{\text{work}}\rangle \perp |0\cdots 0\rangle$. If Shor's algorithm is implemented on a perfect quantum computer without any imperfections these nonrelevant states are never populated. However, errors and imperfections will populate these states and their role is potentially quite important in this context. In this work we do not want to enter into the details of the effects due to the work space qubits. So, we simply assume that our model of imperfection effects (17) acts only in the computational register, or in other words the static imperfections do not couple computational qubits with work space qubits. However, even in this approximation we still keep track of the nonrelevant states in the computational register (the states $|y\rangle$ with $y \geqslant N$).

For numerical simulations of Shor's algorithm in the presence of imperfections we use a classical computer taking into account the control register (with up to 20 qubits) and the computational register (with up to 10 qubits) and up to 30 qubits in total. We do not implement the quantum multiplication operator in terms of elementary gates but we directly implement the unitary operator as given in Eq. (13). To model the static imperfections we used the multiplication operator with errors given by Eq. (18) and with the effective perturbation operator given by

$$\delta H_{\text{eff}}(x) = \sum_{i=0}^{n_q-1} \delta_i \sigma_i^{(z)} + 2\sum_{i=0}^{n_q-2} J_i \sigma_i^{(x)}\sigma_{i+1}^{(x)}, \tag{21}$$

where $\sigma_i^{(\nu)}$ are the Pauli operators acting on the ith qubit (of the computational register) and δ_j, J_j are random coefficients, chosen differently for each value of x and distributed according to

$$\delta_i, J_i \in [\sqrt{3}\epsilon, \sqrt{3}\epsilon]. \tag{22}$$

We remind that even for static imperfections $\delta H_{\text{eff}}(x)$ given by Eqs. (19) and (20) strongly depends on the actual value of x because this factor is hardcoded in realistic implementations by the choice of elementary gates U_j. According to Eq. (14), we have to apply the (controlled version) of the multiplication operator for all values

$$x \in \{x^{2^j} \bmod N | j = 0, \ldots, n_l - 1\}. \tag{23}$$

In our numerical simulations we have ensured by the proper choice of δ_i, J_i that $\delta H_{\text{eff}}(x^{2^j} \bmod N)$ is identical to $\delta H_{\text{eff}}(x^{2^l} \bmod N)$ if for $j \neq l$ we have $x^{2^j} = x^{2^l} \bmod N$. Otherwise, we have chosen different realizations of δ_i, J_i for each value of $x^{2^j} \bmod N$ assuming that the x-dependence of the hard coded implementation is sufficiently complex to render $\delta H_{\text{eff}}(x)$ uncorrelated for different values of x. This introduces some kind of slight correlation that takes into account the static property of the imperfections. However, we have also checked that neglecting these correlations [choosing each time a different realization of $\delta H_{\text{eff}}(x)$ even if the same x-value appears again] does not affect significantly our numerical results discussed below. We also note that in potential applications (for "real" quantum computers) with larger values of N and n_l, n_q these kinds of correlations will become less important. So, in the majority of cases for each j we have $\delta H_{\text{eff}}(j)$ with independent random realizations of δ_i, J_i in Eq. (21) distributed as in Eq. (22).

In principle the unitary operator $e^{i\delta H_{\text{eff}}(x)}$ is quite random due to Eqs. (19) and (20) and should directly couple many qubits in the computational register. Our model (21) for the imperfections is quite convenient for numerical computations and is similar to the model used in [17,18] but with a difference that in those works it is the elementary residual Hamiltonian δH which is chosen in this way [see Eq. (17)]. Therefore we should expect that $\delta H_{\text{eff}}(x)$ has a more complicated structure than Eq. (21). However, choosing δ_j and J_j of comparable size we are well in the quantum chaos regime [16,18,22] and therefore the model (21) describes well the effects of static imperfections. It should be also noted that the quantum gates of the algorithm introduce an additional strong mixing between all qubits even if they are not directly coupled by $\delta H_{\text{eff}}(x)$ (see [22] for details). We also checked that a change of $\delta H_{\text{eff}}(x)$ from the form of Eq. (21) to the case when all qubits are coupled by residual interactions does not affect significantly the results of numerical simulations. This is in agreement with the results obtained in [18,22]. Another advantage of a choice of $\delta H_{\text{eff}}(x)$ in the form (21) is the local structure of couplings between qubits that corresponds to a physical reality. It is also important to note that when we have $n_g = n_l$ gates as in Fig. 2, then the effective strength of ϵ is effectively renormalized as $\epsilon \to \epsilon\sqrt{n_g}$ since static errors in each realization of $\delta\mathcal{H}_j$ (see Fig. 2) are independent and random. We leave the question about possible strong correlations between $\delta\mathcal{H}_j$ due to a specific implementation of the algorithm for future studies.

The above consideration assumes that sufficient randomization of static imperfections takes place along the path of a specific quantum circuit for the modular multiplication. In this case we may assume that the effective Hamiltonian $\delta H_{\text{eff}}(x)$ in the propagator contains different random couplings between qubits for each value of x [see Eqs. (19) and (22)]. However, it is possible that the errors remain well correlated along the path of this circuit and in this case it is more appropriate to consider that $\delta H_{\text{eff}}(x)$ does not depend on x and remains the same along the whole Shor's algorithm. In our numeral studies we mainly concentrate on the first possibility ("generic imperfection model") but in order to have the complete picture of the effects of static imperfections we also considered the second case with $\delta H_{\text{eff}}(x)$ remaining constant along the full circuit ("correlated imperfection model"). According to our previous discussion the important property of both models is that the errors appear only via the positions of the propagator $e^{i\delta H_{\text{eff}}(x)}$ between the modular multiplications in the full circuit of the algorithm. Hence the specific implementation of the modular multiplication circuit does not affect the random properties of interqubit couplings in $\delta H_{\text{eff}}(x)$. Therefore once the parametric dependence on the imperfection strength ϵ, number of qubits n_q, and number of gates n_g is established through numerical simulations, we can apply these results to arbitrary implementations currently discussed in the literature.

In some sense, the model of static errors considered here can be viewed as a kind of generic static error model. It shows sufficiently rich and generic effects of errors and due to its certain simplicity allows one to make numerical simulations with factorization of larger N values compared to previous numerical studies [5,6,9,10]. This allowed us to determine the accuracy dependence on the parameters and to obtain the scaling law for a large number of qubits. This required us to perform extensive numerical simulations with up to 30 qubits which became possible because we neglected the errors in the work space qubits. However, as soon as we obtain the parametric dependence of the algorithm accuracy we may reincorporate the effect of imperfections in the work space by modifying the effective qubit number in the computational register. We also neglected the static imperfections in the control register since the number of gates in the QFT (operating in the control register) is much smaller than the number of gates in the main part of Shor's algorithm. However, in the case of the correlated imperfection model, we verified that the introduction of couplings in the control register does not modify the established parametric dependence on the number qubits. We emphasize that our numerical calculations keep the exact quantum entanglement for the whole quantum evolution with up to 30 qubits. We note that a further increase of the factorized number N can be achieved by replacing the control register by one qubit combined with appropriate measurements of this qubit and a semiclassical implementation of the QFT [10–14]. However, this approach simulates the quantum measurement process in the algorithm and does not give direct access to the full probability distribution in the quantum register which is substantially used in our studies. We present obtained numerical results in the next section.

IV. NUMERICAL RESULTS

The effects of static imperfections in Shor's algorithm are studied numerically following the approach described in the previous section: a wave vector of size 2^L is propagated numerically according to the quantum circuits shown in Figs. 1 and 2, all quantum gates are assumed to be exact, the imperfections, induced by residual couplings between qubits in the computational register, are encountered by the propagators $\exp(i\delta\mathcal{H}_j)$ appearing n_l times in the circuit as it is described in Fig. 2. We factorize numbers N up to $N \approx 1000$. This means that we simulate numerically a quantum computer with up to 30 qubits, 10 computational qubits and 20 control qubits (we assume ideal evolution in the workspace). The list of factorized numbers N used for numerical simulations is given in Table I. We try to consider mainly the most difficult cases when N has only two factors and their values are more or less comparable.

In Fig. 3 we show a typical example of the probability distribution $P(c)$ of Eq. (11) for the ideal case $\epsilon=0$ (top) and for $\epsilon=0.1$ (bottom). It can be seen that the imperfections significantly reduce the amplitudes of the main r peaks and lead to the appearance of new small peaks in new positions.

Since the success of the algorithm depends essentially on a probability of hitting r-peaks in the process of measurement then the most direct way to study this probability is by *clashing* all the peaks into one, or in other words, adding them all together by taking c modulus s where s is the nearest integer value of the ratio Q/r and thus reducing all probabilities inside one cell with s states. In this way we obtain a new distribution of global search probability $W(c)$:

$$W(c) = \sum_{j=0}^{r-1} P([c + s + jQ/r] \bmod s), \qquad (24)$$

where now $c=-s/2,\ldots,s/2-1$ (the difference of c for P and W is clear from the context) and $s \approx Q/r$ is the distance be-

TABLE I. Values for the data presented in Figs. 8–10 for the generic imperfection model. Only the values with symbols are plotted in Figs. 8 and 9.

N	n_q	L	ϵ_c	x	r		Number real
$14=2\times 7$	4	12	0.440	3	6	□	45
$21=3\times 7$	5	15	0.240	2	6	■	80
$33=3\times 11$	6	18	0.155	2	10	○	35
$35=5\times 7$	6	18	0.157	4	6		60
$35=5\times 7$	6	18	0.175	2	12	●	80
$55=5\times 11$	6	18	0.155	6	10		70
$55=5\times 11$	6	18	0.175	2	20	△	80
$77=7\times 11$	7	21	0.155	10	6		70
$77=7\times 11$	7	21	0.145	6	10	▲	70
$77=7\times 11$	7	21	0.140	2	30		70
$91=7\times 13$	7	21	0.135	3	6		70
$91=7\times 13$	7	21	0.150	2	12	▽	35
$143=11\times 13$	8	24	0.115	2	60	▼	35
$221=13\times 17$	8	24	0.132	2	24	◇	50
$299=13\times 23$	9	27	0.106	2	132	◆	23
$323=17\times 19$	9	27	0.108	2	72	+	30
$437=19\times 23$	9	27	0.099	2	198	×	10
$437=19\times 23$	9	27	0.103	18	22		10
$505=5\times 101$	9	27	0.106	2	100	*	10
$667=23\times 29$	10	30	0.098	2	308	□	10
$943=23\times 41$	10	30	0.096	2	220	■	10

tween peaks. For the ideal algorithm this global probability $W(c)$ has one peak at $c=0$ that stresses the important property of Shor's algorithm: it is not important what peak from the main chain of r peaks is selected by measurement, it is important to know its exact position modulus s that allows one to determine the r value and then to find the factors of N by classical computations. The global probability $W(c)$ is distributed over states with $c=-s/2,\ldots,s/2-1$ and is normalized to unity in this interval.

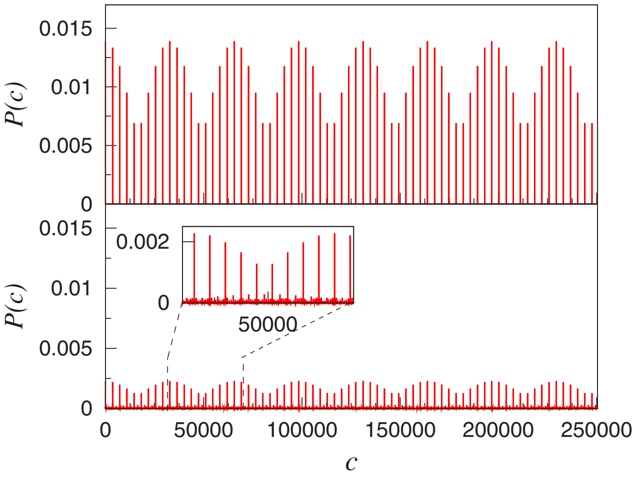

FIG. 3. (Color online) Probability $P(c)$ [Eqs. (9) and (11)] at the final stage of Shor's algorithm for $\epsilon=0$ (top) and $\epsilon=0.1$ (bottom) for values $N=323$, $n_q=9$, $L=27$, $x=2$, and $r=72$.

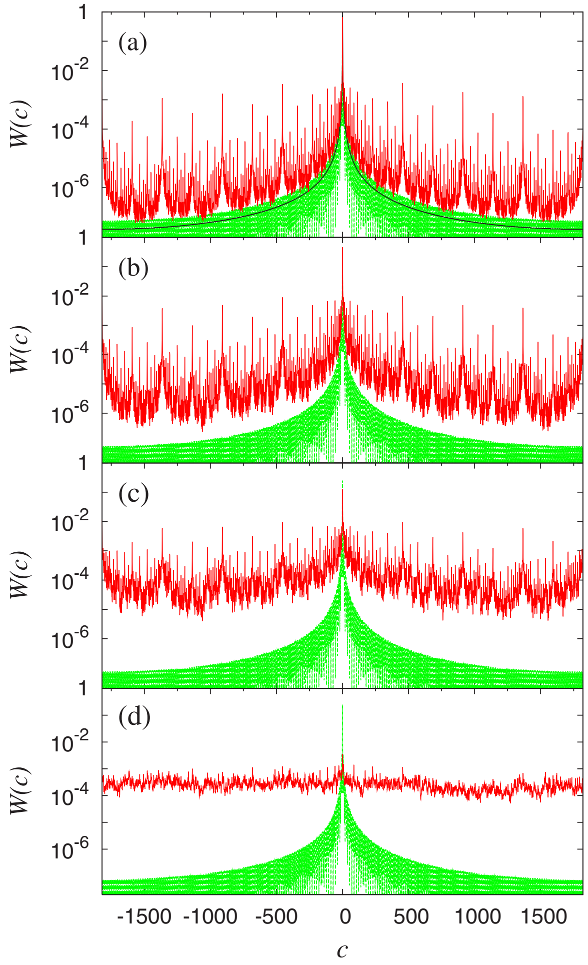

FIG. 4. (Color online) The global probability distribution $W(c)$, as defined in Eq. (24), averaged over ten realizations of random static imperfections, for different values of coupling strength ϵ: (a) $\epsilon=0.025$, (b) $\epsilon=0.05$, (c) $\epsilon=0.1$, and (d) $\epsilon=0.2$. The fast oscillating green (gray) lower curve shows the theoretical probability $G(c)$ at $\epsilon=0$ [Eq. (25)]. The solid (black) curve in (a) is the actual probability at $\epsilon=0$ obtained numerically. The red (dark gray) curves show $W(c)$ obtained numerically at given values of $\epsilon>0$. Here, as in Fig. 3, $N=323$, $n_q=9$, $L=27$, $x=2$, and $r=72$.

In Fig. 4 we show a typical example of the global probability $W(c)$ variation with the increase of coupling strength ϵ. The distribution $W(c)$ for the ideal algorithm at $\epsilon=0$ is well-described by the envelope function $W_0(c)=[\sin(\pi c)/(\pi c)]^2$ of the distribution $G(c)$ discussed in [31] [see also Eq. (10)]:

$$G(c)=\left(\frac{r}{Q}\right)^2\left(\frac{\sin(\pi c)}{\sin(\pi c r/Q)}\right)^2. \qquad (25)$$

Shor's algorithm is successful if the probability at $c=0$ is significant (comparable to 1). This is indeed the case for small values of ϵ [Figs. 4(a) and 4(b)]. In these cases the main probability is concentrated near $c=0$. There are new peaks appearing at very large values of c but they have rather small total probability. With a further growth of ϵ the number of such peaks and their probability grow [Fig. 4(c)], the am-

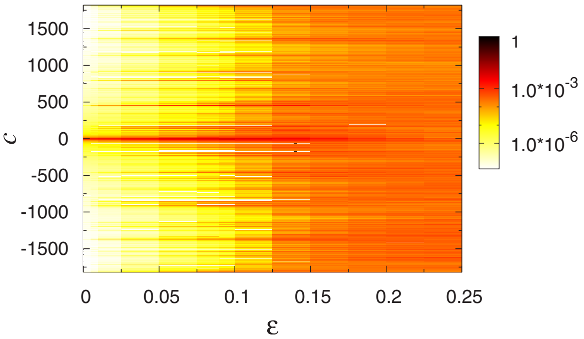

FIG. 5. (Color online) Quantum melting of Shor's algorithm induced by imperfections: color density plot of the global search probability $W(c)$ as a function of coupling strength between qubits ϵ for $N=323$, $n_q=9$, $L=27$, $x=2$, and $r=72$ [$W(c)$ is averaged over 20 realizations].

plitude of the central peak at $c=0$ drops and above a certain ϵ the distribution $W(c)$ becomes practically flat [Fig. 4(d)] that signifies the complete destruction of the algorithm. A pictorial view of variation of $W(c)$ with ϵ is shown in Fig. 5.

In order to study the effects of static imperfections on the algorithm accuracy in a more quantitative way it is convenient to use the inverse participation ratio (IPR)

$$\xi = \sum_c |W(c)|^{-2}, \tag{26}$$

which gives a number of effectively populated states in the distribution $W(c)$. This quantity is extensively used to characterize the properties of many-body quantum states (see, e.g., [16,32]). Another convenient characteristics is the width of the distribution defined as

$$\Delta n = \sqrt{\sum_c W(c)(c - \langle c \rangle)^2}. \tag{27}$$

The dependence of these quantities on the perturbation strength ϵ is shown in Figs. 6 and 7 for the typical case $N=323$. The value of ξ is practically constant up to a value $\epsilon_c \approx 0.1$ after which it starts to grow abruptly. On the contrary, the width Δn grows starting from small values of ϵ. At large ϵ the saturation of growth takes place due to a finite number of states inside the distribution $W(c)$.

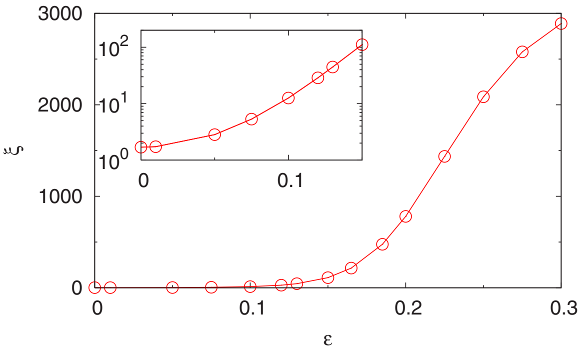

FIG. 6. (Color online) Averaged IPR ξ given by Eq. (26) as a function of ϵ for $N=323$, $n_q=9$, $L=27$, and $x=2$, the inset shows the dependence on small ϵ in log-scale, the average is done over the number of realizations given in Table I.

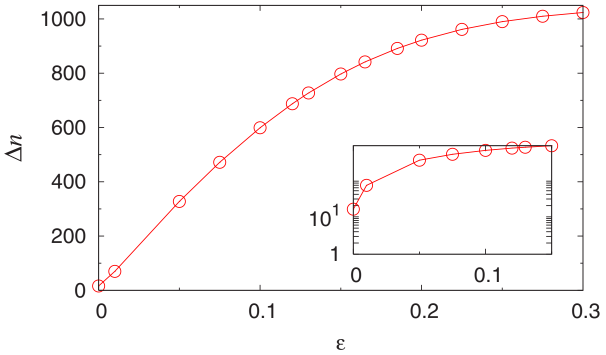

FIG. 7. (Color online) Same as in Fig. 6 but for Δn given by Eq. (27).

The dependence of the IPR ξ on ϵ for different N is shown in Fig. 8. The data clearly show that the dependence becomes more and more sharp with the increase of N. For Δn we see a strong increase with N but there is no such sharp behavior (see Fig. 9). We attribute such a difference to the fact that even small ϵ gives far transitions with exponentially large $c \sim s \approx Q/r \propto 2^{n_q}$. Due to that the second moment of the probability distribution grows exponentially with the number of qubits. The numerical data on dependence of Δn on N at a small fixed ϵ indeed show the exponential growth with $\Delta n \approx A\epsilon N$ with a numerical constant $A \approx 14$ (Fig. 9, bottom panel). A similar behavior has been seen for quantum chaos algorithms [33,34]. The mechanism of this exponential

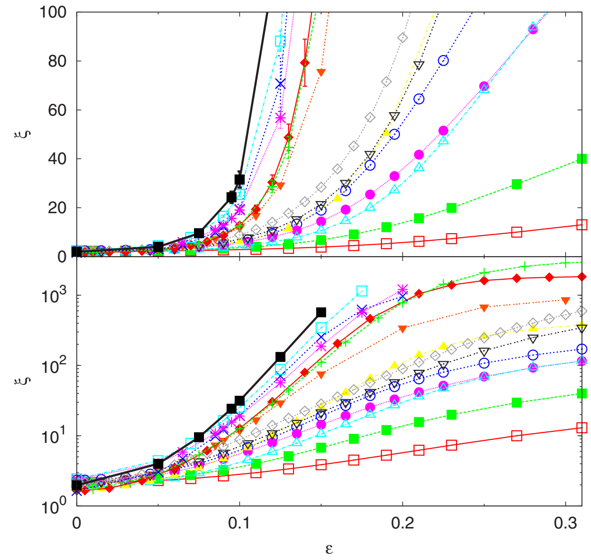

FIG. 8. (Color online) Dependence of averaged IPR ξ on strength of imperfections ϵ for different values of N, curves with symbols from the list of Table I (from top curves with the largest $N=943$ to bottom curves with the smallest $N=14$); the number of disorder realizations used for averaging is given in Table I. For the large values of N we show typical statistical error bars, for small N the error bars are comparable with the symbol size and we do not show them. Top and bottom panels show ξ in normal and logarithmic scale, respectively.

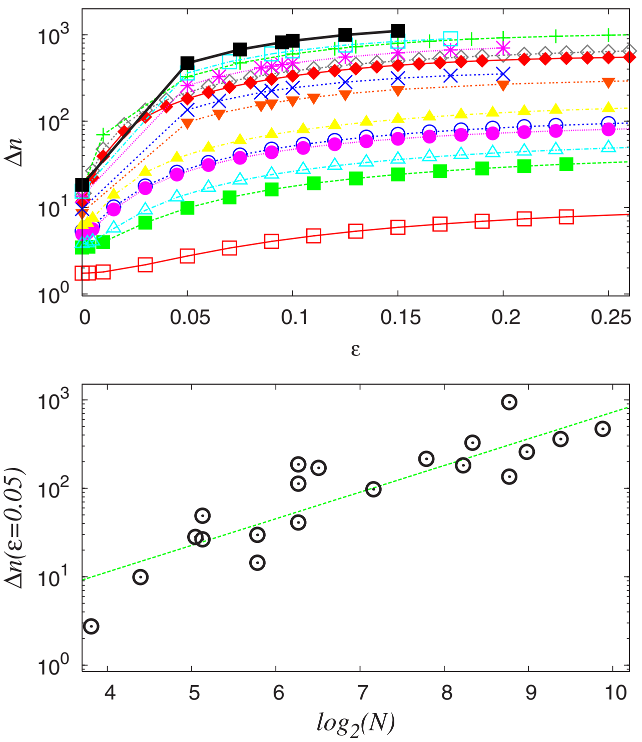

FIG. 9. (Color online) Top panel: dependence of averaged Δn on ϵ for different N with the same symbols from Table I as in Fig. 8. Bottom panel shows the dependence of Δn on N in log-log scale for $\epsilon=0.05$, the straight line shows the dependence $\Delta n = A\epsilon N$ with $A \approx 14$.

growth is the following [33,34]: the gates with imperfections transfer a probability $W_\epsilon \sim \epsilon^2 n_q$ from the search state at $c \approx 0$ to about n_q peaks (see Fig. 4) distributed in the interval of size $s \sim N$. Here, n_q comes from the norm of the Hamiltonian (21) with n_q qubits with local couplings. There are $n_g = n_l \approx 2n_q$ such transitions W_ϵ during the whole algorithm computation. Thus we obtain the second moment of the distribution $W(c)$:

$$(\Delta n)^2 \approx a^2 \epsilon^2 n_q N^2, \qquad (28)$$

where according to numerical data of Fig. 9 (bottom) the numerical coefficient $a \approx A/\sqrt{n_q} \approx 4.5$ is close to the one obtained in [33,34]. Of course, the fluctuations in Fig. 9 (bottom) are rather large. We think that the main origin of these fluctuations is related to the arithmetic properties of x, r, and N. Indeed, r varies significantly with x and N (see Table I) that clearly affects the transition probability induced by imperfections [35]. In spite of these fluctuations the global exponential growth of Δn with n_q is seen rather clearly. Such an exponential sensitivity of $(\Delta n)^2$ on N is not very pleasant for the algorithm accuracy, but in principle this behavior is not so dangerous. Indeed, the total probability to have exponentially large values of c is very small and doing a few measurements and making a majority "vote" will eliminate such extreme values of Δn.

Therefore more crucial is the behavior of ξ since above a certain critical value ϵ_c the probability $W(c)$ spreads over very many levels and the algorithm stops to work. Indeed, it is known that static imperfections can lead to a complete delocalization, for example, in the case of a quantum algorithm simulating the Anderson localization in three dimensions [36].

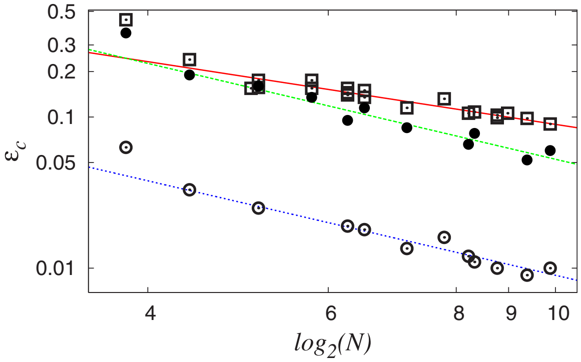

FIG. 10. (Color online) Dependence of ϵ_c, obtained from the criterion $\xi(\epsilon_c)=10\xi(\epsilon=0)$, on $\log_2 N$ in log-log scale. The numerical data are shown for the generic imperfection model (squares, data from Table I), the correlated imperfection model with qubit couplings in the computational register (full circles), and the correlated imperfection model with all qubits in the control and computational registers coupled by interactions (open circles), see text for model description. The straight lines show the fit $\epsilon_c = B/(\log_2 N)^\beta$ in the interval $4 < \log_2 N < 10$ with $B=0.98\pm0.16$, $\beta=1.04\pm0.094$ for squares (top line), $B=2.06\pm0.42$, $\beta=1.6\pm0.11$ for full circles (middle line), and $B=0.33\pm0.05$, $\beta=1.57\pm0.09$ for open circles (bottom line).

To determine the delocalization border for Shor's algorithm and the dependence of ϵ_c on N in the generic imperfection model we use a numerical criterion $\xi(\epsilon_c)=10\xi(\epsilon=0)$. Indeed, an increase by a factor of 10 is sufficiently large to obtain the transition border in ϵ. The dependence of ϵ_c on N is shown in Fig. 10 (squares and top line). From the theoretical viewpoint the errors are accumulated randomly so that the probability W_t transferred from $c=0$ to all other states grows proportionally to the number of gates n_g with errors and thus $W_t \sim W_\epsilon n_g \sim \epsilon^2 n_q n_g \sim \epsilon^2 n_q^2$. We expect that above the border $W_t \sim 1$ the probability becomes delocalized over exponentially many states and the algorithm is destroyed. This gives the quantum chaos border

$$\epsilon_c(N) = B/\log_2(N) \approx \sqrt{2}B/\sqrt{n_q n_g}, \qquad (29)$$

where B is a numerical constant. For our generic imperfection model we have in the second equality $n_g \approx 2n_q$ but in the case when the errors related with the workspace qubits are taken into account we have $n_g \sim n_q^3 \sim (\log_2 N)^3$. The numerical data for ϵ_c are presented in Fig. 10. The fit of the dependence in the form $\epsilon_c = B/(\log_2 N)^\beta$ in the interval $4 < \log_2 N < 10$ gives $B=0.98\pm0.16$, $\beta=1.04\pm0.094$. Thus the numerical data confirm the theoretical estimate (29) with $B \approx 1$. The border ϵ_c drops polynomially with $\log_2 N$ since the whole Shor algorithm is performed in a polynomial number of gates $n_g \sim (\log_2 N)^3$. In this respect the situation is different from the case of the Grover algorithm with imperfections considered in [21] where the number of gates grows exponentially with n_q. The exponential sensitivity of Floquet eigenstates to static imperfections in quantum chaos algorithms [22] also corresponds to a different situation since in a sense the eigenstate corresponds to a very long time scale where the number of gates becomes exponentially large.

In the above consideration for the generic imperfection model, we assumed that the quantum circuit effectively modifies the couplings between qubits in the propagator from one gate to another. Another limiting case corresponds to the correlated imperfection model, where these couplings remain unchanged from gate to gate (see Sec. III). For this particular model we also performed extensive numerical simulations considering two cases: (a) the interactions exist only between qubits in the computational register (see Fig. 10, full circles, middle line) and (b) the interactions exist between all qubits in the control and computation registers (see Fig. 10, open circles, bottom line). For the numerical study of these two cases we used the same quantities as those described above for the generic imperfection model. We do not reproduce all data here but only show the cumulative final dependence for the quantum chaos border ϵ_c defined by the same relation $\xi(\epsilon_c)=10\xi(\epsilon=0)$. The fit of the numerical data in the form $\epsilon_c=B/(\log_2 N)^{\beta}$ gives the same exponent $\beta\approx 1.6$ for both cases of the correlated imperfection model with the numerical factors as in Fig. 10. Naturally B becomes smaller when all qubits are coupled. The value of β is definitely larger as compared to the generic imperfection model (where $\beta\approx 1$). This can be understood on the following physical grounds: the errors accumulate coherently along n_g gates so that the transition probability from the target state to all other states is $W_t \sim W_\epsilon n_g^2 \sim \epsilon^2 n_q n_g^2 \sim \epsilon^2 n_q^3$. The quantum chaos border is given by the condition $W_t \sim 1$ that gives

$$\epsilon_c(N) = B/\log_2(N)^{3/2} \approx 2B/\sqrt{n_q n_g^2}, \qquad (30)$$

since we always chose $n_g \approx 2n_q$. The theoretical exponent $\beta=1.5$ is in good agreement with the numerical fit $\beta=1.6\pm 0.1$. We also clearly see that the fact of coupling all qubits does not affect the parametric dependence of the chaos border on $\log_2(N)$ and gives only a change of the numerical prefactor B. It is important to note that the quantum chaos border is lower for the correlated imperfection model.

V. CONCLUSION

We performed extensive numerical simulations of Shor's algorithm factorizing numbers up to $N=943$ on a quantum computer with up to 30 qubits in the presence of residual static couplings between qubits. Our studies show that the width Δn of r peaks, whose positions are essential for determination of factors of N, grow exponentially with N [see Eq. (28)]. However, the use of majority vote with few measurements allows us to eliminate the rare events which contribute to this exponential growth. In fact the algorithm remains operational up to the critical coupling strength ϵ_c which drops polynomially with $\log_2 N$ [see Eq. (29)]. Since with the work space qubits the total number of gates in Shor's algorithm is $n_g \sim (\log_2 N)^3$ the relation (29) gives $\epsilon_c \sim 1/(\log_2 N)^2$. In this estimate, based on Eq. (29) with $n_q \sim \log_2 N$ and $n_g \sim (\log_2 N)^3$, we assume the validity of the generic imperfection model where couplings fluctuate from gate to gate. Another limit corresponds to the case of the correlated imperfection model where couplings remain fixed for all gates. In this case the relation (30) gives $\epsilon_c \sim 1/(\log_2 N)^{7/2}$. A presence of finite correlation length $1 \leqslant n_{gcor} \leqslant n_g$ in the number of gates n_g will give interpolation between these two limiting cases with $\epsilon_c \sim 1/[(\log_2 N)^2 \sqrt{n_{gcor}}]$. At present, the latest RSA challenge number factored is RSA-640 with $\log_2 N = 640$ [37]. Thus, assuming a more optimistic case of the generic imperfection model, a quantum computer which factors this number should have a dimensionless coupling strength $\epsilon < \epsilon_c \sim 2\times 10^{-6}$. The value of ϵ can be interpreted as $\epsilon \approx J_{\mathrm{res}}\delta t$, where J_{res} is a strength of residual couplings and $\delta t \approx 1/J_g$ is a time duration of a two-qubit gate which is related to a typical value of coupling J_g between two qubits which implements this gate. As a result, we obtain that $\epsilon \sim J_{\mathrm{res}}/J_g$ has the meaning of the ratio between a residual coupling between qubits and a coupling strength implementing a two-qubit gate. According to the above estimate in a quantum computer this ratio should be kept as small as $J_{\mathrm{res}}/J_g < \epsilon_c \sim 2\times 10^{-6}$ to have a possibility to beat a modern classical computer in the RSA-factorization. Such a restriction raises serious requirements to experimental implementations of quantum computers, but it is possible to hope that future technological progress will make this possible. Finally we note that we do not consider here quantum error corrections (see [38] and references therein) which may improve the situation but on a price of significant increase of the total number of qubits required for computations.

ACKNOWLEDGMENTS

This work was supported in part by the EC IST-FET project EuroSQIP. For numerical simulations we used the codes of Quantware Library [39].

[1] P. W. Shor, in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, edited by S. Goldwasser (IEEE Computer Society, Los Alamitos, CA, 1994).

[2] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge Univ. Press, Cambridge, England, 2000).

[3] L. M. K. Vanderspyen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, Nature (London) **414**, 883 (2001).

[4] J. I. Cirac and P. Zoller, Phys. Rev. Lett. **74**, 4091 (1995).

[5] C. Miquel, J. P. Paz, and R. Perazzo, Phys. Rev. A **54**, 2605 (1996).

[6] C. Miquel, J. P. Paz, and W. H. Zurek, Phys. Rev. Lett. **78**, 3971 (1997).

[7] H. Guo, G.-L. Long, and Y. Sang, J. Chin. Chem. Soc. (Taipei) **48**, 449 (2001).

[8] A. G. Fowler and L. C. L. Hollenberg, Phys. Rev. A **70**, 032329 (2004).

[9] L. F. Wei, X. Li, X. Hu, and F. Nori, Phys. Rev. A **71**, 022317 (2005).

[10] S. J. Devitt, A. G. Fowler, and L. C. L. Hollenberg, Quantum Inf. Comput. **6**, 616 (2006).

[11] M. Mosca and A. Ekert, Lecture Notes in Computer Science Vol. 1509 (Springer, New York, 1999), p. 174; e-print arXiv:quant-ph/9903071, Proceedings of the First NASA International Conference on Quantum Computing and Quantum Communication (Springer-Verlag, Berlin, in press).

[12] C. Zalka, e-print arXiv:quant-ph/9806084.

[13] S. Parker and M. B. Plenio, Phys. Rev. Lett. **85**, 3049 (2000).

[14] S. Beauregard, Quantum Inf. Comput. **3**, 175 (2003).

[15] A. G. Fowler, S. J. Devitt, and L. C. L. Hollenberg, Quantum Inf. Comput. **4**, 237 (2004).

[16] B. Georgeot and D. L. Shepelyansky, Phys. Rev. E **62**, 3504 (2000); **62**, 6366 (2000).

[17] G. Benenti, G. Casati, S. Montangero, and D. L. Shepelyansky, Phys. Rev. Lett. **87**, 227901 (2001).

[18] K. M. Frahm, R. Fleckinger, and D. L. Shepelyansky, Eur. Phys. J. D **29**, 139 (2004).

[19] G. G. Carlo, G. Benenti, G. Casati, and C. Mejia-Monasterio, Phys. Rev. A **69**, 062317 (2004).

[20] J. W. Lee and D. L. Shepelyansky, Phys. Rev. E **71**, 056202 (2005).

[21] O. V. Zhirov and D. L. Shepelyansky, Eur. Phys. J. D **38**, 405 (2006).

[22] G. Benenti, G. Casati, S. Montangero, and D. L. Shepelyansky, Eur. Phys. J. D **20**, 293 (2002).

[23] K. Maity and A. Lakshminarayan, Phys. Rev. E **74**, 035203(R) (2006).

[24] A. A. Pomeransky, O. V. Zhirov, and D. L. Shepelyansky, Eur. Phys. J. D **31**, 131 (2004).

[25] V. Vedral, A. Barenco, and A. Ekert, Phys. Rev. A **54**, 147 (1996).

[26] D. Beckman, A. N. Chari, S. Devabhaktuni, and J. Preskill, Phys. Rev. A **54**, 1034 (1996).

[27] P. Gossett, e-print arXiv:quant-ph/9808061.

[28] T. G. Draper, S. A. Kutin, E. M. Rains, and K. M. Svore, Quantum Inf. Comput. **6**, 351 (2006).

[29] R. Van Meter and K. M. Itoh, Phys. Rev. A **71**, 052320 (2005).

[30] C. Zalka, e-print arXiv:quant-ph/0601097.

[31] E. Gerjuoy, Am. J. Phys. **73**, 521 (2005).

[32] B. Georgeot and D. L. Shepelyansky, Phys. Rev. Lett. **79**, 4365 (1997).

[33] P. H. Song and D. L. Shepelyansky, Phys. Rev. Lett. **86**, 2162 (2001).

[34] B. Levi, B. Georgeot, and D. L. Shepelyansky, Phys. Rev. E **67**, 046220 (2003).

[35] We note that our choice of errors shown in Fig. 2 does not affect the probability of an ideal algorithm for a specific case of N with x and r being powers of 2 (e.g., $N=15$, $x=2$).

[36] A. A. Pomeransky and D. L. Shepelyansky, Phys. Rev. A **69**, 014302 (2004).

[37] http://www.rsasecurity.com/rsalabs/node.asp?id=2964.

[38] D. Gottesman, e-print arXiv:quant-ph/0701112, special issue of Physics in Canada (to be published).

[39] *Quantware Library: Quantum Numerical Recipes*, edited by K. M. Frahm and D. L. Shepelyansky, http://www.quantware.ups-tlse.fr/QWLIB/