Optimal resources for topological two-dimensional stabilizer codes: Comparative study

H. Bombin and M. A. Martin-Delgado
Departamento de Física Teórica I, Universidad Complutense, 28040 Madrid, Spain

(Received 4 April 2007; published 6 July 2007)

We study the resources needed to construct topological two-dimensional stabilizer codes as a way to estimate in part their efficiency, and this leads us to perform a comparative study of surface codes and color codes. This study clarifies the similarities and differences between these two types of stabilizer code. We compute the topological error-correcting rate $C := n/d^2$ for surface codes C_s and color codes C_c in several instances. On the torus, typical values are $C_s=2$ and $C_c=3/2$, but we find that the optimal values are $C_s=1$ and $C_c=9/8$. For planar codes, a typical value is $C_s=2$, while we find that the optimal values are $C_s=1$ and $C_c=3/4$. In general, a color code encodes twice as many logical qubits as does a surface code.

DOI: 10.1103/PhysRevA.76.012305 PACS number(s): 03.67.Lx, 03.67.Pp

I. INTRODUCTION

Decoherence of quantum states is one of the main reasons why we have not achieved so far many of the impressive results predicted by quantum-information theory. Battling decoherence has become a very important issue in this field. Devising new strategies to deal with decoherence effects is equally important. One of these strategies, "the topological way," relies on quantum states endowed with a robustness arising when they are embedded into certain Hilbert spaces that exhibit topological protection [1].

Quantum error correction has provided us with definite techniques to do error correction on quantum states belonging to quantum codes [2,3]. A suitable formalism to study quantum error-correction codes is the stabilizer formalism [4]. In fact, a topological quantum code is a special type of stabilizer code [1], as will be discussed in Sec. II. It is a reservoir of states that are intrinsically robust against decoherence due to the encoding of information in the topology of the system.

From the point of view of quantum computation, a quantum error-correcting code is a quantum memory [5–7]. Thus, a topological code amounts to a quantum memory with topological protection, and it can be endowed with extra computational capabilities under certain circumstances [8–10]. Moreover, one remarkable property of topological codes is the fact that their generators are local in the physical qubits of the quantum system. This means that each code generator involves only near-neighbor qubits. This locality property proves to be very useful for performing quantum error-correcting tasks with ancilla qubits, and it becomes an advantage with respect to standard codes, which are nontopological.

The physical mechanism that underlies a topological quantum code is called a topological order [11–13]. This is a new type of quantum phase for matter. In a topological order, there exists ground-state degeneracy without breaking any symmetry, in sharp contrast with more standard phases based on the spontaneous symmetry-breaking mechanism. This degeneracy has a topological origin. Thus, topological orders deviate significantly from more standard orders treated within the Landau symmetry-breaking theory [14–17].

Topological protection is very appealing and has many virtues, but there are also difficulties to implement it in practice. This is currently an active and broad area. We shall not be concerned with experimental realizations of topological codes here.

Our main interest is to analyze the resources needed for their construction and the optimality of those resources. In doing so, we shall perform a very illustrative comparative study of the similarities and differences between the main examples of topological stabilizer codes, namely, surface codes [1] and color codes [18].

This paper is organized as follows. In Sec. II, we introduce the surface codes in a slightly different manner than the usual one [1], but otherwise equivalent. We do so because their comparison with color codes [18] is more transparent in this way. Color codes are constructed with certain two-dimensional complexes, two-colexes, introduced in [15]. We point out the shortcoming of the surface codes with respect to color codes for implementation of a variety of important transverse quantum logic gates belonging to the Clifford group. Another advantage of color codes is that they encode twice the number of logical qubits as do surface codes. In Sec. III, we introduce the notion of topological error-correcting rate C for topological codes. It gives information about how good a code is for error correction when the number of physical qubits, n, is increased. We compute this scaling for both surface codes and color codes with the topology of a torus and a plane, which are the most important examples of topologies in two-dimensional (2D) for practical reasons. Moreover, we compute the optimal values of this figure of merit C for those topological 2D stabilizer codes. Section IV is devoted to conclusions.

II. TOPOLOGICAL 2D STABILIZER CODES

We start by introducing the notion of a stabilizer quantum error-correcting code [4]. Let X, Y, and Z denote the usual Pauli matrices, which act on the space $\mathcal{H}_2$ of a single qubit. A Pauli operator p_n of length n is any tensor product of the form

$$p_n := \bigotimes_{i=1}^{n} \sigma_i, \qquad \sigma_i \in \{I,X,Y,Z\}. \tag{1}$$

The closure of such operators as a group is the Pauli group $\mathbf{P}_n$. Given an Abelian subgroup $\mathcal{S} \subset \mathbf{P}_n$ not containing

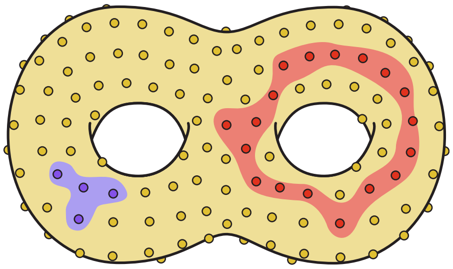


FIG. 1. (Color online) A two-torus is an example of a topological space in which a topological stabilizer code can be constructed. Here the dots represent the qubits pinned down onto the surface. The small blue (darker gray) area on the left is the support of a local generator of the stabilizer $\mathcal{S}$ (2). The big red (darker gray) area on the right, which cannot be deformed to a point, is the support of an undetectable error (in $\mathcal{Z}-\mathcal{S}$).

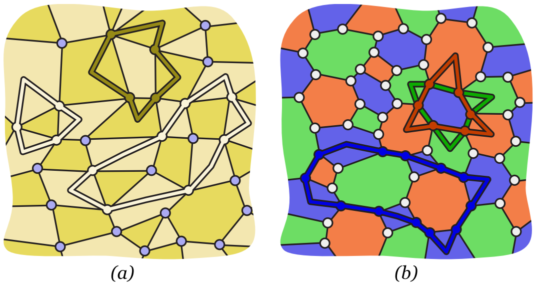
I, a stabilizer code of length n is the subspace $\mathcal{C} \subset \mathcal{H}_2^{\otimes n}$ formed by those vectors $|\phi\rangle$ with eigenvalue 1 for any element $s \subset \mathcal{S}$,

$$s|\phi\rangle = |\phi\rangle. \qquad (2)$$

Let $\mathcal{Z}$ be the centralizer of S in $\mathbf{P}_n$, i.e., the set of operators in $\mathbf{P}_n$ that commute with the elements of $\mathcal{S}$. A Pauli operator $z \in \mathcal{Z}$ not contained in S up to a phase leaves $\mathcal{C}$ invariant and acts nontrivially in $\mathcal{C}$. Such operators, when regarded as errors, are clearly undetectable. Let the weight of an operator be the number of qubits in which it acts nontrivially. Then the minimal length among the operators in $\mathcal{Z}-\mathcal{S}$ is called the distance of the code. Indeed, the code is capable of correcting a set of Pauli errors $\mathcal{E}$ as long as for any $M,N \in \mathcal{E}$ the operator $M^{\dagger}N$ is not an undetectable error. Therefore, a code of distance $d=2t+1$ can correct all the errors of length less than or equal to t. Given $z \in \mathcal{Z}$ and $s \in \mathcal{S}$, z and zs act equally in $\mathcal{C}$. Then, choosing suitably among the equivalence classes of $\mathcal{Z}/\mathcal{S}$, we can find a Pauli operator basis for the encoded qubits.

Topological stabilizer codes can be roughly defined as stabilizer codes in which the generators of $\mathcal{S}$ can be chosen to be local and undetectable errors have a support that is topologically nontrivial, as shown in Fig. 1. We are assuming that the physical qubits that make up the stabilizer code are placed in a certain topological space. In particular, we will only consider codes placed onto two-dimensional surfaces. One of the ideas behind topological stabilizer codes is that the locality of the generators is something very advantageous in order to perform error correction. Another important idea is that of self-protected quantum memories, something that we will touch upon later.

The first example of topological stabilizer codes were the toric codes [1], in which the qubits are placed in a torus. More generally, other surfaces and arbitrary lattices on them can be considered, and the resulting codes were termed, in general, surface codes [19,5]. We will introduce here surface codes in a way that differs slightly from the original one but is absolutely equivalent. Consider any tetravalent lattice [20] with bicolorable plaquettes, such as the one shown in Fig. 2(a). The plaquettes are split into two sets, which we label as dark and light sets of plaquettes. A surface code can be obtained from such a lattice by placing a qubit at each of its sites and choosing suitable plaquette operators. In general, given a plaquette p, we define the plaquette operators



(a) (b)

FIG. 2. (Color online) Typical lattices for both kinds of 2D topological stabilizer codes. In both cases, qubits are represented as circles placed at sites. (a) A piece of a surface code [1]. Dark plaquettes have B_p^X stabilizers attached, while light plaquettes have B_p^Z stabilizers attached. In both cases, these operators correspond to closed strings. For example, an X-string operator which is the product of three dark plaquette operators is shown. (b) A piece of a color code [18]. All plaquettes have B_p^X and B_p^Z stabilizers attached, which can be both visualized as two strings of different colors. In the case of a blue (darkest gray) plaquette, its operators can be considered either as green (softest gray) or as red strings. For example, a blue string operator which is the product of two red and one green plaquette operators is shown.

$$B_p^{\sigma} := \bigotimes_i \sigma^{s_p(i)}, \qquad \sigma = X,Z, \qquad (3)$$

where the product is over all the sites and $s_p(i)$ equals 1 for sites belonging to p and zero otherwise. In the case of surface codes, the generators of $\mathcal{S}$ are B_p^X for dark plaquettes and B_p^Z for light plaquettes. Note how all these operators commute, thus generating an Abelian subgroup. The encoded states $|\phi\rangle$ satisfy the conditions

$$B_p^X|\phi\rangle = |\phi\rangle, \qquad \forall \quad p \in P_D, \qquad (4)$$

$$B_p^Z|\phi\rangle = |\phi\rangle, \qquad \forall \quad p \in P_L, \qquad (5)$$

where P_D and P_L are, respectively, the sets of dark and light plaquettes.

Let a Z operator (X operator) be any tensor product of Z's (X's) and I's. Then any Z operator (X operator) can be visualized as a string that connects dark (light) plaquettes and acts nontrivially on those qubits it goes through [see Fig. 2(a)]. Any light (dark) plaquette operator is a Z-string (X-string) operator. Then the product of several plaquette operators of the same kind is a string operator lying on the boundary of certain area containing precisely the plaquettes [see Fig. 2(a)]. Any Pauli operator is, up to a phase, the product of an X string and a Z string. In this sense, the strings belonging to $\mathcal{S}$ are all boundaries. More generally, any operator in $\mathcal{Z}$ is a product of closed string operators. Closed strings are strings without end points, and their importance is now clear since those closed strings which are not boundaries make up precisely the set of undetectable errors. From these undetectable errors we can choose a Pauli operator ba-

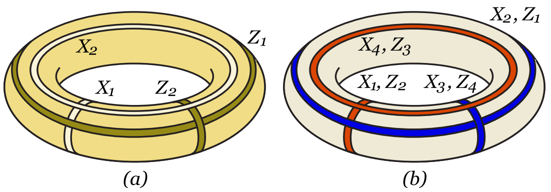


FIG. 3. (Color online) Pauli operator bases in the torus. The encoded operators correspond to certain strings operators. (a) For surface codes [1], the number of encoded qubits is 2. (b) For color codes [18], the number of encoded qubits is 4. Note that each string has two operators attached, one of each type.

sis for the encoded qubits, that is, we can choose the encoded Z and X operators, acting on the logical qubits. It is customary to denote these encoded operators as $\bar{Z}$ and $\bar{X}$ to distinguish them from the standard operators acting on the physical qubits instead. To this end, we note two properties. (i) String operators of the same kind that differ only by a boundary, that is, which are equal up to a deformation, have the same action on encoded states. (ii) A light and a dark string operator commute if they cross an even number of times and anticommute otherwise. Observe that this crossing parity is preserved by the string deformations just mentioned.

Taking this into account, one can obtain the desired Pauli operator basis and find out that the number of encoded qubits is $2g$ for a g torus, that is, a sphere with g "handles," [see Fig. 3(a)].

The topological nature of surface codes makes them very attractive. In particular, the measurements required for quantum error correction can be locally performed and involve the few qubits lying on each plaquette. On the other hand, they are not so nice if one intends to perform transversal operations with codes. In fact, only the controlled-NOT (CNOT) gate can be performed transversally in surface codes. It was precisely with the aim to overcome this difficulty that color codes [18] were devised, which are also 2D topological stabilizer codes but allow the transverse implementation of any operation in the Clifford group. This set of operations is especially suited for quantum-information tasks, such as quantum teleportation or entanglement distillation.

In the case of color codes [18], the starting point is a trivalent lattice with tricolorable plaquettes [see Fig. 2(b)]. We label the plaquettes as green, red, or blue. Again, qubits must be placed at sites, but now for each plaquette p we have both operators B_p^X and B_p^Z as generators of the stabilizer $\mathcal{S}$. The encoded states $|\phi\rangle$ satisfy the conditions

$$B_p^X|\phi\rangle = |\phi\rangle, \quad \forall \quad p, \tag{6}$$

$$B_p^Z|\phi\rangle = |\phi\rangle, \quad \forall \quad p. \tag{7}$$

As in the case of surface codes, string operators are essential for the analysis of color codes. However, now the same geometrical string can be attached to two different operators, the corresponding X and Z strings. An extra labeling turns out to be extremely useful, and so we speak of red, green, and blue strings. Blue strings connect blue plaquettes, and so on, just as X strings connect dark plaquettes in surface codes. Observe how any blue plaquette operator, for example, can be considered both a green and a blue string operator. Also, the product of, say, several green and red plaquette operators is a blue string operator lying in the boundary of certain area containing precisely the plaquettes [see Fig. 2(b)]. As in surface codes, the strings appearing in $\mathcal{S}$ are all boundaries, and the strings appearing in $\mathcal{Z}$ are closed strings. Also, those closed strings that are not boundaries comprise undetectable errors, and from these undetectable errors we can choose the encoded $\bar{Z}$'s and $\bar{X}$'s. Again, we have two guiding properties. (i) String operators of the same type (X or Z) and color that are equal up to a deformation have the same action on encoded states. (ii) String operators commute with one another, unless they cross an odd number of times and have different color and type.

Taking this into account one can obtain a Pauli operator basis and find that the number of encoded qubits is $4g$ for a g torus, that is, two times the number of encoded qubits in surface codes [see Fig. 3(b)]. It is customary to denote a quantum error correcting code made with n physical qubits, encoding k logical qubits and with distance d as $[[n,k,d]]$. With this notation we have that for a fixed surface topology

$$k_c = 2k_s, \tag{8}$$

where the subscript c stands for color codes and s for surface codes. Thus, we see that color codes are more efficient than surface codes as far as the number of encoded qubits is concerned. However, we may wonder whether this doubling of logical qubits has been achieved at the expense of introducing a bigger number of physical qubits n, or whether it affects the correcting capabilities d of the code.

III. EFFICIENCY OF TOPOLOGICAL CODES

To answer those questions, we need to study how efficient 2D topological stabilizer codes are in terms of the number of qubits required with respect to the distance of the code. In fact, regular lattices in which all plaquettes have the same number of qubits are especially relevant. More specifically, it is instructive to consider surface and color codes obtained from regular lattices on the torus. In this case, the plaquettes must be squares for surface codes, and hexagons for color codes (see Fig. 4). Let us consider first the family of surface codes corresponding to Fig. 4(a), which was in fact the first family of topological stabilizer codes [1]. The code in the figure has distance $d=4$, the number of physical qubits is $n=32$, and the number of encoded qubits is $k=2$. More generally, this particular example can easily be generalized to a family of codes in which clearly

$$C_s \coloneqq \frac{n_s}{d_s^2} = 2. \tag{9}$$

Here, we have defined the notion of error-correcting rate C_s for surface codes, a figure of merit which allows us to compare 2D topological stabilizer codes. It is a measure of how the error-correction capabilities of the code scales when the number of physical qubits is increased. The fact that the number of required qubits scales quadratically with distance

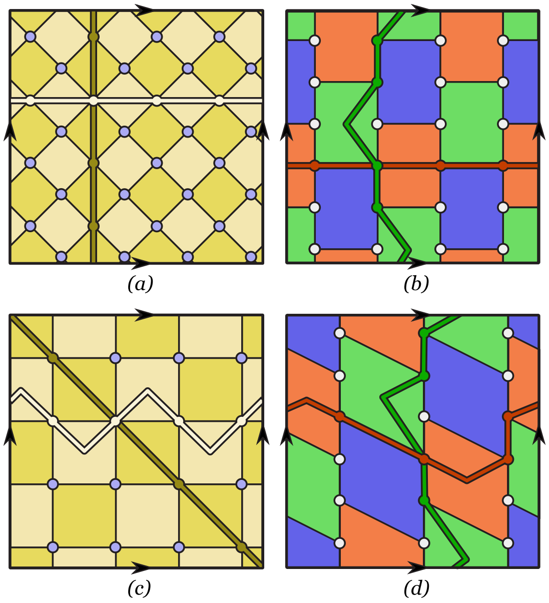


FIG. 4. (Color online) Examples of regular codes in the torus with distance $d=4$. Some nontrivial strings of minimal length are displayed. (a) In the surface code [1], plaquettes are squares and the number of physical qubits is $n=32$. (b) For the color code [18], plaquettes are hexagons and $n=24$. (c) An optimal regular surface code reduces the number of qubits to $n=16$. (d) An optimal regular color code with $n=18$.

is not surprising and is a common feature of all 2D topological stabilizer codes. However, the asymptotic value of C for a quantum error-correction code may vary, and the value 2 is not a particularly good one, as we shall see. Let us consider now the color code in Fig. 4(b). It encodes $k=4$ qubits, is made up of $n=24$ qubits, and its distance is $d=4$. Then we have that the error-correcting rate for this color code is

$$C_c \coloneqq \frac{n_c}{d_c^2} = \frac{3}{2}. \tag{10}$$

Moreover, this is just an example of an infinite family in which this ratio is preserved, so that apparently color codes not only encode more qubits (8), but also require fewer physical qubits for a given distance.

However, as was noted in [6], the surface codes just considered are not optimal in terms of the number of physical qubits n. In fact, if the optimal codes are chosen, only half of them are really needed, as Fig. 4(c) illustrates. Thus, for optimal regular surface codes in a torus we have

$$C_s^{\text{op}} = 1. \tag{11}$$

Can a similar optimization be obtained for regular color codes? The answer is yes, and the corresponding lattice is illustrated in Fig. 4(d). For this code we have

$$C_c^{\text{op}} = \frac{9}{8}, \tag{12}$$

very close to the value for surface codes. Again, attaching l^2 copies of this lattice together gives a family of codes of distance $4l$ with the same ratio n/d^2. Therefore, we conclude that, in a torus, color codes encode twice as many qubits but surface codes require slightly fewer physical qubits for optimal regular lattices.

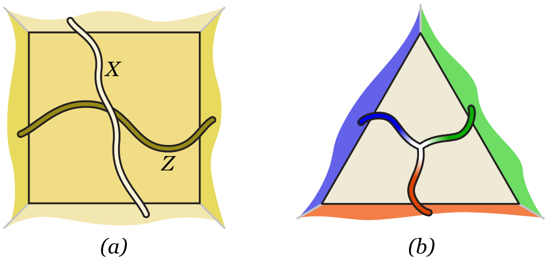


FIG. 5. (Color online) Geometry of planar codes encoding a single qubit. The colors in the borders represent the class of the missing face. Only suitable strings can have end points at each border. The string operators corresponding to encoded operators are shown. (a) A planar surface code. (b) The triangular (color) code, in which encoded operators are related to a string net.

Is this true also for other surfaces, or is it something particular to the torus? Instead of trying to answer this question in general, we consider probably the most important example of topological stabilizer codes in practice. By this, we mean planar codes, that is, topological codes that can be placed in a piece of planar surface. More particularly, we want to compare surface and color codes encoding a single qubit, which are the most interesting not only as quantum memories but also for quantum computation [5,8].

The trick to obtain planar codes from lattices related to surfaces without borders is the same for surface and color codes. In particular, it is enough to remove plaquettes from the original lattice until the resulting surface can be unfolded onto a plane. The new lattice has borders, and we have to explain which are the strings that now play the role that closed strings played before in the case of a compact surface like the torus. First, consider what happens when a dark plaquette, that is, the corresponding plaquette operator, is removed from a surface code. Take any Z string that has an end point in the removed plaquette. Since this Z-string operator does not commute with the plaquette operator from the removed plaquette, prior to the removal it was not in $\mathcal{Z}$, but after the removal it could be, at least as far as this end point is concerned. Therefore, the removal of a dark (light) plaquette creates a dark (light) border at which only Z strings (X strings) can end. Additionally, some string operators winding around the removed plaquette are no longer boundaries, but these considerations do not have relevance in the geometries that we are considering. As for color codes, the situation is similar. When a blue plaquette is removed, a blue border is created in which only blue strings can end, and so on and so forth.

With these ideas in mind, it is not difficult to understand the code geometries shown in Fig. 5, which solves the question we have raised before.

For surface codes there are two borders of each type, so that the nontrivial X strings (Z strings) connect light (dark) borders [see Fig. 5(a)]. In the case of color codes, there are three borders, one of each color, and the nature of the encoded operators shows a feature of color codes that we have

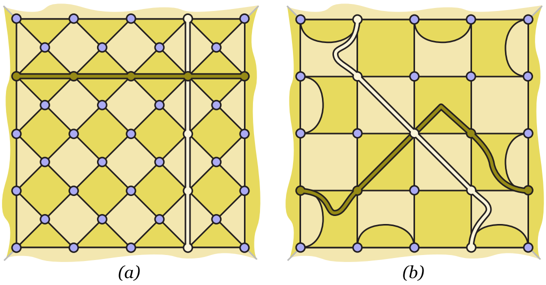


FIG. 6. (Color online) Surface codes encoding a single qubit and with distance $d=5$. (a) A nonoptimal version with $n=41$. (b) An optimized version with $n=25$.

not discussed yet. The point is that in color codes string nets are allowed. In particular, branching points at which three different colored strings of the same type meet are allowed. This means that such a configuration does not violate any of the plaquette conditions. Then, for these triangular codes, the encoded $\bar{X}$ and $\bar{Z}$ operators are constructed with such a string net, as Fig. 5(b) shows.

We want to consider the efficiency of these families of planar codes, which encode a single qubit. In Fig. 6, two different versions of the surface code with distance $d=5$ are shown. The version of Fig. 6(a) belongs to the original family of single-qubit surface codes, for which the asymptotic value of the ratio is

$$C_s \sim 2. \tag{13}$$

This is not the best that can be done, as the code of Fig. 6(b) shows with approximately half the number of qubits and equal distance. In fact, the optimized value for planar surface codes is

$$C_s^{\mathrm{op}} = 1. \tag{14}$$

As for triangular codes, examples for distances $d=3,5,7$ are shown in Fig. 7. It is straightforward to continue this family for arbitrarily large distances. For these color codes, the asymptotic value yields the following optimized value:

$$C_c^{\mathrm{op}} \sim \frac{3}{4}. \tag{15}$$

Therefore, triangular codes are not only particularly interesting for quantum computation [8], but also more efficient in terms of the number of physical qubits required.

Finally, we would like to touch upon how topological stabilizer codes give rise to the idea of self-protected quantum memories. To this end, one must consider a physical system in which qubits are placed according to the geometry of the topological code, and introduce certain Hamiltonian dictated by the generators of the stabilizer. In the case of surface codes, the Hamiltonian is

$$H \coloneqq -\sum_{p\in P_D} B_p^X - \sum_{p\in P_L} B_p^Z, \tag{16}$$

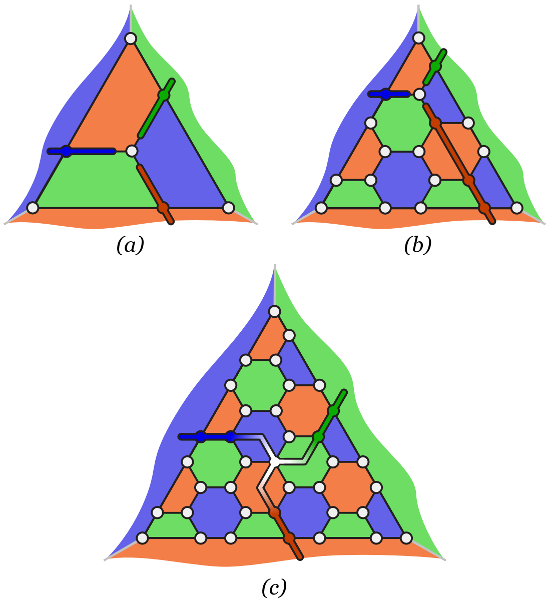


FIG. 7. (Color online) Color codes encoding a single qubit. (a) Triangular code with distance $d=3$ and number of qubits $n=7$. (b) Triangular code with $d=5$ and $n=19$. (c) Triangular code with $d=7$ and $n=37$.

while for color codes it is

$$H \coloneqq -\sum_{p} (B_p^X + B_p^Z). \tag{17}$$

One of the main differences between these Hamiltonians is that for color codes all plaquettes play the same role, whereas in the case of surface codes we have to distinguish between light and dark plaquettes. In any case, in both cases the ground states correspond to encoded states, and there exists a gap that separates them from excited states. Moreover, no local operator can connect ground states. Only those operators with a topologically nontrivial support are able to distinguish among these protected states, something which makes this quantum memories remarkably robust against perturbations with a local nature.

IV. CONCLUSIONS

In this paper, we have made a presentation of surface codes [1] and color codes [18] on equal footing. This allows us to make a comparative study of their properties in more detail, such as the possible set of gates that they can implement transversally and the number k of encoded qubits (logical qubits).

We have also introduced the notion of the error-correcting rate $C \coloneqq n/d^2$ as a means to evaluate the performance of topological 2D codes as far as the error-correction capability is concerned. We have computed this figure of merit for surface codes and color codes in the most representative and important topologies: the torus and the plane. In the torus, we find that the optimal value for surface codes is $C_s=1$, while for color codes we find $C_c=\frac{9}{8}$, which is are very close. For practical applications, planar codes are the most valuable topologies. For them we find that the optimized values for surface codes are again $C_s=1$, but this time color codes yield a better value, $C_c=\frac{3}{4}$. Having in mind that the number of encoded logical qubits for color codes is always, i.e., in any topology, twice as much as for surface codes (8), this means that color codes demand a lower number of physical qubits for their construction.

Finally, we want to point out that it would be very interesting to perform additional comparative studies between Kitaev's codes and color codes regarding other performance properties like the percolation threshold, for instance.

ACKNOWLEDGMENTS

We acknowledge financial support from the EJ-GV (H.B.), a DGS grant under Contract No. BFM 2003-05316-C02-01 (M.A.M.D.), and a CAM-UCM Grant No. 910758.

[1] A. Yu. Kitaev, Ann. Phys. (N.Y.) **303**, 2 (2003).
[2] P. Shor, Phys. Rev. A **52**, R2493 (1995).
[3] A. M. Steane, Phys. Rev. Lett. **77**, 793 (1996).
[4] D. Gottesman, Phys. Rev. A **54**, 1862 (1996).
[5] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, J. Math. Phys. **43**, 4452 (2002).
[6] H. Bombin and M. A. Martin-Delgado, Phys. Rev. A **73**, 062303 (2006).
[7] H. Bombin and M. A. Martin-Delgado, J. Math. Phys. **48**, 052105 (2007).
[8] H. Bombin and M. A. Martin-Delgado, Phys. Rev. Lett. **98**, 160502 (2007).
[9] R. Raussendorf, J. Harrington, and K. Goyal, e-print arXiv:quant-ph/0703143.
[10] R. Raussendorf and J. Harrington, Phys. Rev. Lett. **98**, 190504 (2007).
[11] X.-G. Wen and Q. Niu, Phys. Rev. B **41**, 9377 (1990).
[12] X.-G. Wen, Int. J. Mod. Phys. B **4**, 239 (1990).
[13] X.-G. Wen, *Quantum Field Theory of Many-body Systems* (Oxford University Press, Oxford, 2004).
[14] M. A. Levin and X.-G. Wen, Phys. Rev. B **71**, 045110 (2005).
[15] H. Bombin and M. A. Martin-Delgado, Phys. Rev. B **75**, 075103 (2007).
[16] X.-Y. Feng, G.-M. Zhang, and T. Xiang, Phys. Rev. Lett. **98**, 087204 (2007).
[17] A. Hamma and D. A. Lidar, e-print arXiv:quant-ph/0607145.
[18] H. Bombin and M. A. Martin-Delgado, Phys. Rev. Lett. **97**, 180501 (2006).
[19] S. B. Bravyi and A. Yu. Kitaev, arXiv:quant-ph/9811052.
[20] A tetravalent lattice is a lattice with coordination number 4, i.e., each site has four nearest-neighbor sites.