Quantum Error Correcting Codes Using Qudit Graph States

Shiang Yong Looi,[1, *] Li Yu,[1] Vlad Gheorghiu,[1] and Robert B. Griffiths[1]

[1]*Department of Physics, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, U.S.A.*
(Dated: Version of 13 June 2008)

Graph states are generalized from qubits to collections of n qudits of arbitrary dimension D, and simple graphical methods are used to construct both additive and nonadditive, as well as degenerate and nondegenerate, quantum error correcting codes. Codes of distance 2 saturating the quantum Singleton bound for arbitrarily large n and D are constructed using simple graphs, except when n is odd and D is even. Computer searches have produced a number of codes with distances 3 and 4, some previously known and some new. The concept of a stabilizer is extended to general D, and shown to provide a dual representation of an additive graph code.

I. INTRODUCTION

Quantum error correction is an important part of various schemes for quantum computation and quantum communication, and hence quantum error correcting codes, first introduced about a decade ago [1, 2, 3] have received a great deal of attention. For a detailed discussion see Ch. 10 of [4]. Most of the early work dealt with codes for qubits, with a Hilbert space of dimension $D = 2$, but qudit codes with $D > 2$ have also been studied [5, 6, 7, 8, 9, 10, 11]. They are of intrinsic interest and could turn out to be of some practical value.

Cluster or graph states, which were initially introduced in connection with measurement based or one-way quantum computing [12], are also quite useful for constructing quantum codes, as shown in [7, 8, 9] in a context in which both the encoding operation and the resulting encoded information are represented in terms of graph states. In the present paper we follow [9] in focusing on qudits with general D, thought of as elements of the additive group $\mathbb{Z}_D$ of integers mod D. However, our strategy is somewhat different, in that we use graph states and an associated basis (graph basis) of the n-qudit Hilbert space in order to construct the coding subspace, while *not* concerning ourselves with the encoding process. This leads to a considerable simplification of the problem along with the possibility of treating nonadditive graph codes on exactly the same basis as additive or stabilizer codes. It also clarifies the relationship (within the context of graph codes as we define them) of degenerate and nondegenerate codes, though in this paper we focus mainly on the latter. The approach used here was developed independently in [13] and [14] for $D = 2$, and in [15] for $D > 2$; thus several of our results are similar to those reported in these references.

Following an introduction in Sec. II to Pauli operators, graph states, and the graph basis, as used in this paper, the construction of graph codes is the topic of Sec. III.

In Sec. III A we review the conditions for an $((n, K, \delta))_D$ code, where n is the number of carriers, K the number of codewords or dimension of the coding space, δ the distance of the code, and D the dimension of the Hilbert space of one qudit. We also consider the distinction between degenerate and nondegenerate codes. Our definition of graph codes follows in Sec. III B, and the techniques we use to find nondegenerate codes, which are the main focus of this paper, are indicated in Sec. III C, while various results in terms of specific codes are the subject of Sec. IV.

In Sec. IV B we show how to construct graph codes with $\delta = 2$ that saturate the quantum Singleton (QS) bound for arbitrarily large n and D, except when n is odd and D is even, and we derive a simple sufficient condition for graphs to yield such codes. For n odd and $D = 2$ we have an alternative and somewhat simpler method of producing nonadditive codes of the same size found in [16]. For both $D = 2$ and $D = 3$ we have studied nondegenerate codes on sequences of cycle and wheel graphs, in Secs. IV C and IV D. These include a number of cases which saturate the QS bound for $\delta = 2$ and 3, and others with $\delta = 3$ and 4 which are the largest possible additive codes for the given n, D, and δ. Section IV D contains results for a series of hypercube graphs with $n = 4$, 8, and 16, and in particular a $((16, 128, 4))_2$ additive code.

In Sec. V we show that what we call G-additive codes are stabilizer codes (hence "additive" in the sense usually employed in the literature), using a suitable generalization of the stabilizer formalism to general D. In this perspective the stabilizer is a dual representation of a code which is equally well represented by its codewords. The final Sec. VI has a summary of our results and indicates directions in which they might be extended.

II. PAULI OPERATORS AND GRAPH STATES

A. Pauli operators

Let $\{|j\rangle\}$, $j = 0, 1, \ldots D-1$ be an orthonormal basis for the D-dimensional Hilbert space of a qudit, and define

*Electronic address:slooi@andrew.cmu.edu

the unitary operators [17]

$$Z = \sum_{j=0}^{D-1} \omega^j \, |j\rangle\langle j| \,, \quad X = \sum_{j=0}^{D-1} |j\rangle\langle j \oplus 1| \,, \qquad (1)$$

with $\oplus$ denoting addition mod D. They satisfy

$$Z^D = I = X^D, \quad XZ = \omega ZX, \quad \omega := \mathrm{e}^{2\pi \mathrm{i}/D}. \qquad (2)$$

We shall refer to the collection of D^2 operators $\{X^\mu Z^\nu\}$, $\mu, \nu = 0, 1, \ldots, D-1$, as (generalized) *Pauli operators*, as they generalize the well known $I, X, Z, XZ\,(=-iY)$ for a qubit. Together they form the *Pauli basis* of the space of operators on a qudit.

For a collection of n qudits with a Hilbert space $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \cdots \mathcal{H}_n$ we use subscripts to identify the corresponding Pauli operators: thus Z_l and X_l operate on the space $\mathcal{H}_l$ of qudit l. An operator of the form

$$P = \omega^\lambda X_1^{\mu_1} Z_1^{\nu_1} X_2^{\mu_2} Z_2^{\nu_2} \cdots X_n^{\mu_n} Z_n^{\nu_n}, \qquad (3)$$

where λ, and μ_l and ν_l for $1 \le l \le n$, are integers in the range 0 to $D-1$, will be referred to as a *Pauli product*. If μ_l and ν_l are both 0, the operator on qudit l is the identity, and can safely be omitted from the right side of (3). The collection $\mathcal{Q}$ of all operators P of the form (3) with $\lambda = 0$, i.e., a prefactor of 1, forms an orthonormal basis of the space of operators on $\mathcal{H}$ with inner product $\langle A, B\rangle = D^{-n}\mathrm{Tr}(A^\dagger B)$; we call it the (generalized) *Pauli basis* $\mathcal{Q}$.

If P and Q are Pauli products, so is PQ, and hence the collection $\mathcal{P}$ of all operators of the form (3) for n fixed form a multiplicative group, the *Pauli group*. While $\mathcal{P}$ is not Abelian, it has the property that

$$PQ = \omega^\mu QP, \qquad (4)$$

where μ is an integer that depends on P and Q. (When $D = 2$ and $\omega = -1$ it is customary to also include in the Pauli group operators of the form (3) multiplied by i. For our purposes this makes no difference.)

The *base* of an operator P of the form (3) is the collection of qudits, i.e., the subset of $\{1, 2, \ldots n\}$, on which the operator acts in a nontrivial manner, so it is not just the identity, which is to say those j for which either μ_j or ν_j or both are greater than 0. A general operator R can be expanded in the Pauli basis $\mathcal{Q}$, and its base is the union of the bases of the operators which are present (with nonzero coefficients) in the expansion. The *size* of an operator R is defined as the number of qudits in its base, i.e., the number on which it acts in a nontrivial fashion. For example, the base of $P = \omega^2 X_1^2 X_4 Z_4$ (assuming $D \ge 3$) is $\{1, 4\}$ and its size is 2; whereas the size of $R = X_1 + 0.5 X_2 Z_2^2 Z_3 + i X_4$ is 4.

For two distinct qudits l and m the *controlled-phase* operation C_{lm} on $\mathcal{H}_l \otimes \mathcal{H}_m$, generalizing the usual controlled-phase for qubits, is defined by

$$C_{lm} = \sum_{j=0}^{D-1}\sum_{k=0}^{D-1} \omega^{jk} |j\rangle\langle j| \otimes |k\rangle\langle k| = \sum_{j=0}^{D-1} |j\rangle\langle j| \otimes Z_m^j. \ (5)$$

Of course, $C_{lm} = C_{ml}$, and it is easily checked that $(C_{lm})^D = I$. It follows from its definition that C_{lm} commutes with Z_l and Z_m, and thus with Z_p for any qudit p.

B. Graph states

Let $G = (V, E)$ be a graph with n vertices V, each corresponding to a qudit, and a collection E of undirected edges connecting pairs of distinct vertices (no self loops). Multiple edges are allowed, as in Fig.1 for the case of $D = 4$, as long as the multiplicity (weight) does not exceed $D - 1$, thus at most a single edge in the case of qubits. The lm element $\Gamma_{lm} = \Gamma_{ml}$ of the *adjacency matrix* Γ is the number of edges connecting vertex l with vertex m. The graph state

$$|G\rangle = \mathcal{U}\,|G^0\rangle = \mathcal{U}\left(|+\rangle^{\otimes n}\right), \qquad (6)$$

is obtained by applying the unitary operator

$$\mathcal{U} = \prod_{\{l,m\}\in E} (C_{lm})^{\Gamma_{lm}}. \qquad (7)$$

to the product state

$$|G^0\rangle := |+\rangle \otimes |+\rangle \otimes \cdots |+\rangle, \qquad (8)$$

where

$$|+\rangle := D^{-1/2} \sum_{j=0}^{D-1} |j\rangle \qquad (9)$$

is a normalized eigenstate of X, with eigenvalue 1. In (7) the product is over all distinct pairs of qudits, with $(C_{lm})^0 = I$ when l and m are not joined by an edge. Since the C_{lm} for different l and m commute with each other, and also with Z_p for any p, the order of the operators on the right side of (7) is unimportant.
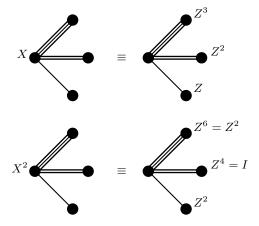

FIG. 1: Action of X and X^2 on graph state ($D = 4$).

Given the graph G we define the *graph basis* to be the set of D^n states

$$|\boldsymbol{a}\rangle := |a_1, a_2, \ldots, a_n\rangle = Z^{\boldsymbol{a}} |G\rangle$$
$$= Z_1^{a_1} Z_2^{a_2} \cdots Z_n^{a_n} |G\rangle \qquad (10)$$

where $\boldsymbol{a} = (a_1, \ldots a_n)$ is an n-tuple of integers, each taking a value between 0 and $D - 1$. The original graph state $|G\rangle$ is $|0, 0, \ldots, 0\rangle$ in this notation. That this collection forms an orthonormal basis follows from the fact that the Z_p operators commute with the C_{lm} operators, so can be moved through the unitary $\mathcal{U}$ on the right side of (6). As the states $Z^\nu |+\rangle$, $0 \leq \nu \leq D - 1$, are an orthonormal basis for a single qudit, their products form an orthonormal basis for n qudits. Applying the unitary $\mathcal{U}$ to this basis yields the orthonormal graph basis. The n-tuple representation in (10) is convenient in that one can define

$$|\boldsymbol{a} \oplus \boldsymbol{b}\rangle := |a_1 \oplus b_1, a_2 \oplus b_2, \ldots, a_n \oplus b_n\rangle,$$
$$|j\boldsymbol{a}\rangle := |ja_1, ja_2, \ldots, ja_n\rangle, \qquad (11)$$

where j is an integer between 0 and $D-1$, and arithmetic operations are mod D.

One advantage of using the graph basis is that its elements are mapped to each other by a Pauli product (up to powers of ω), as can be seen by considering the action of Z_l or X_l on a single qudit. The result for Z_l follows at once from (10). And as shown in App. A and illustrated in Fig. 1, the effect of applying X_l to $|G\rangle$ is the same as applying $(Z_m)^{\Gamma_{lm}}$ to each of the qudits corresponding to neighbors of l in the graph. Applying these two rules and keeping track of powers of ω resulting from interchanging X_l and Z_l, see (2), allows one to easily evaluate the action of any Pauli product on any $|\boldsymbol{a}\rangle$ in the graph basis.

III. CODE CONSTRUCTION

A. Preliminaries

Consider a quantum code corresponding to a K-dimensional subspace, with orthonormal basis $\{|\boldsymbol{c}_q\rangle\}$, of the Hilbert space $\mathcal{H}$ of n qudits. When the Knill-Laflamme [2] condition

$$\langle \boldsymbol{c}_q| Q |\boldsymbol{c}_r\rangle = f(Q)\delta_{qr} \qquad (12)$$

is satisfied for all q and r between 0 and $K - 1$, and every operator Q on $\mathcal{H}$ such that $1 \leq \text{size}(Q) < \delta$, but fails for some operators of size δ, the code is said to have *distance* δ, and is an $((n, K, \delta))_D$ code; the subscript is often omitted when $D = 2$. (See the definition of size in Sec. II A. The only operator of size 0 is a multiple of the identity, so (12) is trivially satisfied.) A code of distance δ allows the correction of any error involving at most $\lfloor (\delta - 1)/2 \rfloor$ qudits, or an error on $\delta - 1$ (or fewer) qudits if the location of the corrupted qudits is already known (e.g., they have been stolen).

It is helpful to regard (12) as embodying two conditions: the obvious off-diagonal condition saying that the matrix elements of Q must vanish when $r \neq q$; and the diagonal condition which, since $f(Q)$ is an arbitrary complex-valued function of the operator Q, is nothing but the requirement that all diagonal elements of Q (inside the coding space) be identical. The off-diagonal condition has a clear analog in classical codes, whereas the diagonal one does not. Both must hold for all operators of size up to and including $\delta - 1$, but need not be satisfied for larger operators.

In the coding literature it is customary to distinguish *nondegenerate* codes for which $f(Q) = 0$ for all operators of size between 1 and $\delta - 1$, i.e., for *all* q and r

$$\langle \boldsymbol{c}_q| Q |\boldsymbol{c}_r\rangle = 0 \text{ for } 1 \leq \text{size}(Q) < \delta, \qquad (13)$$

and *degenerate* codes for which $f(Q) \neq 0$ for at least one Q in the same range of sizes. See p. 444 of [4] for the motivation behind this somewhat peculiar terminology when δ is odd. In this paper our focus is on nondegenerate codes. For the most part they seem to perform as well as degenerate codes, though there are examples of degenerate codes that provide a larger K for given values of n, δ, and D than all known nondegenerate codes. Examples are the $((6, 2, 3))_2$ [18] and $((25, 2, 9))_2$ codes mentioned in [19].

B. Graph codes

When each basis vector $|\boldsymbol{c}_q\rangle$ is a member of the graph basis, of the form (10) for some graph G, we shall say that the corresponding code is a *graph code* associated with this graph. As noted in Sec. I, this differs from the definition employed in [7, 8, 9], but agrees with that in more recent $D = 2$ studies [13, 14], because we do not concern ourselves with the processes of encoding and decoding. In what follows we shall always assume $\delta \geq 2$, since $\delta = 1$ is trivial. As the left side of (12) is linear in Q, it suffices to check it for appropriate operators drawn from the Pauli basis $\mathcal{Q}$ as defined in Sec. II A. It is helpful to note that for any $Q \in \mathcal{Q}$, any pair $|\boldsymbol{c}_q\rangle$ and $|\boldsymbol{c}_r\rangle$ of graph basis states and any n-tuple $\boldsymbol{a}$,

$$\langle \boldsymbol{c}_q \oplus \boldsymbol{a}| Q |\boldsymbol{c}_r \oplus \boldsymbol{a}\rangle = \langle \boldsymbol{c}_q| Z^{-\boldsymbol{a}} Q Z^{\boldsymbol{a}} |\boldsymbol{c}_r\rangle$$
$$= \omega^\mu \langle \boldsymbol{c}_q| Q |\boldsymbol{c}_r\rangle \qquad (14)$$

for some integer μ depending on Q and $\boldsymbol{a}$; see (10), (11) and (4). Therefore, if (12) is satisfied for some Q and a collection $\{|\boldsymbol{c}_q\rangle\}$ of codewords, the same will be true for the same Q and the collection $\{|\boldsymbol{c}_q \oplus \boldsymbol{a}\rangle\}$ (with an appropriate change in $f(Q)$). Thus we can, and hereafter always will, choose the first codeword to be

$$|\boldsymbol{c}_0\rangle = |0, 0, \ldots, 0\rangle = |G\rangle. \qquad (15)$$

Analogous to Hamming distance in classical information theory we define the *Pauli distance* Δ between two

graph basis states as

$$\Delta(\boldsymbol{c}_q, \boldsymbol{c}_r) = \Delta(|\boldsymbol{c}_q\rangle, |\boldsymbol{c}_r\rangle) :=$$
$$\min\{\text{size}(Q) : \langle \boldsymbol{c}_q | Q | \boldsymbol{c}_r \rangle \neq 0\}, \quad (16)$$

where it suffices to take the minimum for $Q \in \mathcal{Q}$, the Pauli basis. (Ket symbols can be omitted from the arguments of Δ when the meaning is clear.) Also note the identities

$$\Delta(\boldsymbol{c}_q, \boldsymbol{c}_r) = \Delta(\boldsymbol{c}_r, \boldsymbol{c}_q) = \Delta(\boldsymbol{c}_q \oplus \boldsymbol{a}, \boldsymbol{c}_r \oplus \boldsymbol{a})$$
$$= \Delta(\boldsymbol{c}_0, \boldsymbol{c}_r \ominus \boldsymbol{c}_q), \quad (17)$$

where $\boldsymbol{a}$ is any n-tuple, and $\ominus$ means difference mod D, see (11). The second equality is a consequence of (14). Note that if in (16) we minimize only over Q operations which are tensor products of Z's (no X's), Δ is exactly the Hamming distance between the n-tuples $\boldsymbol{c}_q$ and $\boldsymbol{c}_r$, see (10).

For the case $q = r$, where (16) gives 0 (for $Q = I$), we introduce a special *diagonal distance* Δ' which is the minimum size of the right side of (16) when one restricts Q to be an element of $\mathcal{Q}$ of size 1 or more. The diagonal distance does not depend on the particular value of $q = r$, but is determined solely by the graph state $|G\rangle$—see (14) with $r = q$—and thus by the graph G. This has the important consequence that if we consider a particular G and want to find the optimum codes for a given δ that is no larger than Δ', the collection of operators $Q \in \mathcal{Q}$ for which (12) needs to be checked will all have zero diagonal elements, $f(Q) = 0$, and we can use (13) instead of (12). In other words, for the graph in question and for $\delta \leq \Delta'$, all graph codes are nondegenerate, and in looking for an optimal code one need not consider the degenerate case. Our computer results in Sec. IV are all limited to the range $\delta \leq \Delta'$ where no degenerate codes exist for the graph in question. Any code with $\delta > \Delta'$ will necessarily be degenerate, since there is at least one nontrivial Q for which (12) must be checked for the diagonal elements.

A code is *G-additive (graph-additive)* if given any two codewords $|\boldsymbol{c}_q\rangle$ and $|\boldsymbol{c}_r\rangle$ belonging to the code, $|\boldsymbol{c}_q \oplus \boldsymbol{c}_r\rangle$ is also a codeword. As shown in Sec. V, this notion of additivity implies the code is additive in the sense of being a stabilizer code. For this reason, we shall omit the G in G-additive except in cases where it is essential to make the distinction. Codes that do not satisfy the additivity condition are called nonadditive. The additive property allows one to express all codewords as "linear combinations" of k suitably chosen codeword generators. This implies an additive code must have $K = D^r$, r an integer, whenever D is prime. We will see an example of this in Sec. IV for $D = 2$.

The *quantum Singleton* (QS) bound [2]

$$n \geq \log_D K + 2(\delta - 1) \quad \text{or} \quad K \leq D^{n-2(\delta-1)} \quad (18)$$

is a simple but useful inequality. We shall refer to codes which saturate this bound (the inequality is an equality) as *quantum Singleton* (QS) codes. Some authors prefer the term MDS, but as it is not clear to us how the concept of "maximum distance separable," as explained in [20], carries over to quantum codes, we prefer to use QS.

C. Method

We are interested in finding "good" graph codes in the sense of a large K for a given n, δ, and D. The first task is to choose a graph G on n vertices, not a trivial matter since the number of possibilities increases rapidly with n. We know of no general principles for making this choice, though it is helpful to note, see App. A, that the diagonal distance Δ' cannot exceed 1 plus the minimum over all vertices of the number of neighbors of a vertex. Graphs with a high degree of symmetry are, for obvious reasons, more amenable to analytic studies and computer searches than those with lower symmetry.

Given a graph G and a distance δ, one can in principle search for the best nondegenerate code by setting $|\boldsymbol{c}_0\rangle = |G\rangle$, finding a $|\boldsymbol{c}_1\rangle$ with $\Delta(\boldsymbol{c}_0, \boldsymbol{c}_1) \geq \delta$, after that $|\boldsymbol{c}_2\rangle$ with both $\Delta(\boldsymbol{c}_0, \boldsymbol{c}_2) \geq \delta$ and $\Delta(\boldsymbol{c}_1, \boldsymbol{c}_2) \geq \delta$, and so forth, until the process stops. However, this may happen before one finds the largest K, because a better choice could have been made for $|\boldsymbol{c}_q\rangle$ at some point in the process. Exhaustively checking all possibilities is rather time consuming, somewhat like solving an optimal packing problem.

In practice what we do is to first construct a lookup table containing the $D^n - 1$ Pauli distances from $|G\rangle$ to all of the other graph basis states, using an iterative process starting with all $Q \in \mathcal{Q}$ of size 1, then of size 2, etc. This process also yields the diagonal distance Δ'. As we are only considering nondegenerate codes, we choose some $\delta \leq \Delta'$, so that (13) can be used in place of (12), and use the table to identify the collection S of all graph basis states with a distance greater than or equal to δ from $|\boldsymbol{c}_0\rangle = |G\rangle$. If S is empty there are no other codewords, so $K = 1$. However, if S is not empty then K is at least 2, and a search for the optimum code (largest K) is carried out as follows.

We produce a graph $\mathcal{S}$ (not to be confused with G) in which the nodes are the elements of S, and an edge connects two nodes if the Pauli distance separating them— easily computed from the lookup table with the help of (17)—is *greater than or equal to* δ. An edge in this graph signifies that the nodes it joins are sufficiently (Pauli) separated to be candidates for the code, and an optimal code corresponds to a largest complete subgraph or *maximum clique* of $\mathcal{S}$. Once a maximum clique has been found, the corresponding graph basis states, including $|\boldsymbol{c}_0\rangle$, satisfy (13) and span a coding space with the largest possible K for this graph G and this δ.

The maximum clique problem on a general graph is known to be NP-complete [21] and hence computationally difficult, and we do not know if $\mathcal{S}$ has special properties which can be exploited to speed things up. We used the relatively simple algorithm described in [22] for

finding a maximum clique, and this is the most time-consuming part of the search procedure.

The method just described finds additive as well as nonadditive codes. In fact one does not know beforehand whether the resultant code will be additive or not. If one is only interested in additive codes, certain steps can be modified to produce a substantial increase in speed as one only has to find a set of generators for the code.

IV. RESULTS

A. Introduction

Results obtained using methods described above are reported here for various sequences of graphs, each sequence containing graphs of increasing n while preserving certain basic properties. We used a computer search to find the maximum number K of codewords for each graph in the sequence, for distances $\delta \leq \Delta'$ and for $D = 2$ or 3, qubits and qutrits, up to the largest number n of qudits allowed by our resources (running time). Sometimes this revealed a pattern which could be further analyzed using analytic arguments or known bounds on the number of codewords.

In the case of distance $\delta = 2$ we can demonstrate the existence of QS codes for arbitrarily large values of n and D, except when n is odd and D is even, see Part A. In the later subsections we report a significant collection of $D = 2$ and 3 codes for $\delta = 2$, 3, and 4, including QS codes; codes which are the largest possible additive codes for that set of n, D and δ; and a new $((16, 128, 4))_2$ additive code.

Tables show the K found as a function of other parameters. The meaning of superscripts used in the tables is given below.

- a – Indicates the maximum clique search was terminated before completion. This means the code we found might not be optimal, i.e. there might be another code with larger K for this graph. We can only say the code is *maximal* in the sense that no codeword can be added without violating (13). Absence of this superscript implies no code with a larger K exists for this δ and this graph, either because the program did an exhaustive search, or because K saturates a rigorous bound.

- b – Indicates a nonadditive code. Codes without this superscript are additive.

- c – Indicates a QS code, one where K saturates the Singleton bound (18).

- d – Indicates this is not a QS code, but the largest possible *additive* (graph or other) code for the given n, δ and D, This follows from linear programming bounds in [23] for $D = 2$ and [24] for $D = 3$, along with the fact, Sec. III B, that for an additive code,
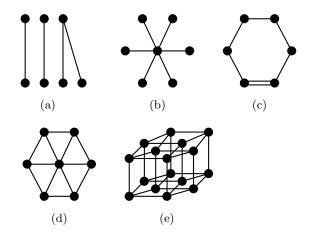

FIG. 2: Examples from different graph sequences: (a) bar (odd n), (b) star, (c) cycle, (d) wheel, (e) $n = 16$ hypercube.

K must be an integer power of D when D is prime. A larger *nonadditive* code for this graph might still be possible in cases flagged with a as well as d.

B. Distance $\delta = 2$; bar and star graphs

It was shown in [19] that for $D = 2$ one can construct $\delta = 2$ QS codes for any even n, and similar codes for larger D are mentioned, without giving details, in [5]. One way to construct graph codes with $\delta = 2$ is to use the method indicated in the proof, App. B, of the following result.

Partition theorem. *Suppose that for a given D the vertices of a graph G on n qudits can be partitioned into two nonempty sets V_1 and V_2 with the property that for each vertex in V_1 the sum of the number of edges (the sum of the multiplicities if multiple edges are present) joining it to vertices in V_2 is nonzero and coprime to D, and the same for the number of edges joining a vertex in V_2 to vertices in V_1. Then there is an additive QS code on G with distance $\delta = 2$.*

A *bar* graph is constructed by taking n vertices and dividing them into two collections V_1 and V_2, of equal size when n is even, and one more vertex in V_2 when n is odd, as in Fig. 2(a). Next pair the vertices by connecting each vertex in V_1 by a single edge to a vertex in V_2, with one additional edge when n is odd, as shown in the figure. (Multiple edges are possible for $D > 2$, but provide no advantage in constructing codes.) When n is even the conditions of the partition theorem are satisfied: 1 is always coprime to D. For odd n, the last vertex in V_1 has 2 edges joining it to V_2, which is coprime to D when D is odd. Hence bar graphs yield $\delta = 2$ QS codes for all n when D is odd, and for even n when D is even.

A *star* graph, Fig. 2(b), has a *central* vertex joined by single edges to every *peripheral* vertex, and no edges connecting pairs of peripheral vertices. Since the diagonal distance Δ' is 2, nondegenerate star codes cannot have δ

TABLE I: Maximum K for qubit and qutrit cycle graphs. See Sec. IV A for detailed meaning of superscripts.

n	$D = 2$		$D = 3$	
	$\delta = 2$	$\delta = 3$	$\delta = 2$	$\delta = 3$
4	4^c	0	9^c	1^c
5	6^b	2^c	27^c	3^c
6	16^c	1	81^c	9^c
7	22^b	2^d	243^c	27^c
8	64^c	8^d	729^c	81^c
9	96^{ab}	12^b	2187^c	243^c
10	256^c	18^b	6561^c	729^c
11	272^{ab}	32^{ad}	19683^c	729^{ad}
12	1024^c	64^{ad}	59049^c	2187^{ad}

[a]Non-exhaustive search
[b]Nonadditive code
[c]Code saturating Singleton bound (18)
[d]Largest possible additive code

larger than 2. As in the case of bar codes, one can construct additive QS codes for any n when D is odd, and for even n when D is even [25]. For odd n and $D = 2$ there are nonadditive codes with

$$K(n) = 2^{n-2} - \frac{1}{2}\binom{n-1}{(n-1)/2};\qquad (19)$$

see App. C for details. Codes with these parameters were discovered earlier by Smolin et al. [16] using a different approach. Computer searches show that for all odd $n \leq 7$ star graphs cannot yield a K larger than (19).

C. Cycle graphs

We used computer searches to look for graph codes based on cycle (loop) graphs, Fig. 2(c). Table I shows the maximum number K of codewords for codes of distance $\delta = 2$ and $\delta = 3$ for both $D = 2$ qubits and $D = 3$ qutrits. In the qutrit case the best codes were obtained by including one double edge (weight 2), as in Fig. 2(c), though when n is odd equally good codes emerge with only single edges. In the qubit case all edges have weight 1.

The $D = 2$ entries in Table I include for $n = 5$ the well known $((5, 2, 3))_2$, the nonadditive $((5, 6, 2))_2$ presented in [26], and, for larger n, a $((9, 12, 3))_2$ code similar to that in [27] and the $((10, 18, 3))_2$ of [13] based upon the same graph.

The $D = 3$, $\delta = 3$ entries are interesting because the QS bound is saturated for $4 \leq n \leq 10$ but *not* for $n = 11$. The $((11, 3^6 = 729, 3))_3$ code we found, the best possible *additive* code according to the linear programming bound in [24], falls short by a factor of 3 of saturating the $K = 3^7 = 2187$ QS bound, and even a nonadditive code based on this graph must have $K \leq 1990$ [28].

One can ask to what extent the results for $\delta = 2$ in Table I could have been obtained, or might be extended

to larger n, by applying the Partition theorem of Part A to a suitable partition of the cycle graph. It turns out—we omit the details—that when D is odd one can use the Partition theorem to produce codes that saturate the QS bound for any n, but when D is even the same approach only works when n is a multiple of 4. In particular, the $((6, 16, 2))_2$ additive QS code in Table I cannot be obtained in this fashion since the cycle graph cannot be partitioned in the required way.

D. Wheel graphs

If additional edges are added to a star graph so as to connect the peripheral vertices in a cycle, as in Fig. 2(d), the result is what we call a *wheel* graph. Because each vertex has at least three neighbors, our search procedure, limited to $\delta \leq \Delta'$, can yield $\delta = 4$ codes on wheel graphs, unlike cycle or star graphs. The construction of $\delta = 2$ codes for any D is exactly the same as for star graphs, so in Table II we only show results for $\delta = 3$ and 4, for both $D = 2$ and 3. The $((16, 128, 4))_2$ additive code appears to be new, and its counterpart in the hypercube sequence is discussed below.

TABLE II: Maximum K for qubit and qutrit wheel graphs. See Sec. IV A for detailed meaning of superscripts.

n	$D = 2$		$D = 3$	
	$\delta = 3$	$\delta = 4$	$\delta = 3$	$\delta = 4$
6	1	1^c	1	1^c
7	2^d	0	27^c	1
8	8^d	1^d	27	9^c
9	8^d	1^d	243^c	9
10	20^c	4^d	243^a	27
11	32^{ad}	4^d	729^{ad}	81
12	64^{ad}	8	2187^{ad}	81^a
13	128^{ad}	16	6561^{ad}	243^a
14	256^{ad}	32^a	19683^{ad}	729^a
15	512^{ad}	64^{ad}	59049^{ad}	2187^a
16	1024^{ad}	128^{ad}		

[a]Non-exhaustive search
[b]Nonadditive code
[c]Code saturating Singleton bound (18)
[d]Largest possible additive code

E. Hypercube graphs

Hypercube graphs, Fig. 2(e), have a high symmetry, and as n increases the coordination bound, App. A, allows Δ' to increase with n, unlike the other sequences of graphs discussed above. We have only studied the $D = 2$ case, with the results shown in Table III. Those for $\delta = 2$ are an immediate consequence of the Partition

theorem: each hypercube is obtained by adding edges between two hypercubes of the next lower dimension, and these are the V_1 and V_2 of the theorem. The generators for the $((16, 128, 4))_2$ additive code are given in Table IV. The $2^7 = 128$ codewords are of the form, see (11), $|\alpha_1\mathbf{g}_1 \oplus \alpha_2\mathbf{g}_2 \oplus \cdots \alpha_7\mathbf{g}_7\rangle$, where each α_j can be either 0 or 1.

TABLE III: Maximum K for qubit hypercube graphs. See Sec. IV A for detailed meaning of superscripts.

n	$D = 2$		
	$\delta = 2$	$\delta = 3$	$\delta = 4$
4	4^c	0	0
8	64^c	8^d	1^d
16	16384^c	512^a	128^{ad}

[a]Non-exhaustive search
[c]Code saturating Singleton bound (18)
[d]Largest possible additive code

V. G-ADDITIVE CODES AS STABILIZER CODES

The stabilizer formalism introduced by Gottesman in [29] for $D = 2$ (qubits) provides a compact and powerful way of generating quantum error correcting codes. It has been extended to cases where D is prime or a prime power in [6, 24, 30]. In [8] stabilizer codes were extended in a very general fashion to arbitrary D from a point of view that includes encoding. However, our approach to graph codes is somewhat different, see Sec. I, and hence its connection with stabilizers deserves a separate discussion. We will show that for any $D \geq 2$ a G-additive (as defined near the end of Sec. III B) code is a stabilizer code, and the stabilizer is effectively a dual representation of the code.

The Pauli group $\mathcal{P}$ for general n and D was defined in Sec. II A. Relative to this group we define a *stabilizer* code (not necessarily a graph code) $\mathcal{C}$ to be a $K \geq 1$-dimensional subspace of the Hilbert space satisfying three conditions:

C1. There is a subgroup $\mathcal{S}$ of $\mathcal{P}$ such that for *every* T in

TABLE IV: Generators of $((16, 128, 4))_2$ additive code for hypercube graph

Generator	Bit notation
$|\mathbf{g}_1\rangle$	$|0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1\rangle$
$|\mathbf{g}_2\rangle$	$|0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1\rangle$
$|\mathbf{g}_3\rangle$	$|0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1\rangle$
$|\mathbf{g}_4\rangle$	$|0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0\rangle$
$|\mathbf{g}_5\rangle$	$|0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1\rangle$
$|\mathbf{g}_6\rangle$	$|0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0\rangle$
$|\mathbf{g}_7\rangle$	$|1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1\rangle$

$\mathcal{S}$ and *every* $|\psi\rangle$ in $\mathcal{C}$

$$T|\psi\rangle = |\psi\rangle \tag{20}$$

C2. The subgroup $\mathcal{S}$ is maximal in the sense that every T in $\mathcal{P}$ for which (20) is satisfied for all $|\psi\rangle \in \mathcal{C}$ belongs to $\mathcal{S}$.

C3. The coding space $\mathcal{C}$ is maximal in the sense that any ket $|\psi\rangle$ that satisfies (20) for every $T \in \mathcal{S}$ lies in $\mathcal{C}$.

If these conditions are fulfilled we call $\mathcal{S}$ the *stabilizer* of the code $\mathcal{C}$. That it is Abelian follows from (4), since for $K > 0$ there is some nonzero $|\psi\rangle$ satisfying (20). One can also replace (20) with

$$T|c_q\rangle = |c_q\rangle \tag{21}$$

where the $\{|c_q\rangle\}$ form an orthonormal basis of $\mathcal{C}$. Note that one can always find a subgroup $\mathcal{S}$ of $\mathcal{P}$ satisfying C1 and C2 for any subspace $\mathcal{C}$ of the Hilbert space, but it might consist of nothing but the identity. Thus it is condition C3 that distinguishes stabilizer codes from non-additive codes. A stabilizer code is uniquely determined by $\mathcal{S}$ as well as by $\mathcal{C}$, since $\mathcal{S}$ determines $\mathcal{C}$ through C3.

As we shall see, the stabilizers of G-additive graph codes can be described in a fairly simple way. Let us begin with one qudit, $n = 1$, where the trivial graph G has no edges, and the graph basis states are of the form $\{Z^c|+\rangle\}$ for c in some collection C of integers in the range $0 \leq c \leq D - 1$. The subgroup $\mathcal{S}$ of $\mathcal{P}$ satisfying C1 and C2 must be of the form $\{X^s\}$ for certain values of s, $0 \leq s \leq D - 1$, belonging to a collection S. This is because Z and its powers map any state $Z^c|+\rangle$ to an orthogonal state, and hence T in (21) cannot possibly contain a (nontrivial) power of Z. Furthermore, since

$$X^s Z^c|+\rangle = \omega^{cs} Z^c|+\rangle, \tag{22}$$

see (2), X^s will leave $\{Z^c|+\rangle\}$ unchanged only if $\omega^{cs} = 1$, or

$$cs \equiv 0 \pmod{D}. \tag{23}$$

Thus for $\mathcal{S}$ to satisfy C1, it is necessary and sufficient that (23) hold for every $c \in C$, as well as every $s \in S$. Further, $\mathcal{S} = \{X^s\}$ is maximal in the sense of C2 only if S contains every s satisfying (23) for each $c \in C$. As shown in App. D, such a collection S must either (depending on C) consist of $s = 0$ alone, or consist of the integer multiples νs_1, with $\nu = 0, 1, \ldots (D/s_1 - 1)$, of some $s_1 > 0$ that divides D. In either case, S is a subgroup of the group $\mathbb{Z}_D$ of integers under addition mod D, and indeed any such subgroup must have the form just described.

We now take up C3. Given the maximal collection S of solutions to (23), we can in turn ask for the collection of C' of integers c in the range 0 to $D - 1$ that satisfy (23) for every s in S. Obviously, C' contains C, but as shown in App. D, $C' = C$ if and only if C is a subgroup of $\mathbb{Z}_D$, i.e., $\mathcal{C}$ is G-additive. Next note that every T in $\mathcal{S}$, as it is

a power of X and because of (22), maps every graph basis state to itself, up to a phase. Thus when (and only when) $\mathcal{C}$ is G-additive, the codewords are just those graph basis states for which this phase is 1 for every $T \in \mathcal{S}$. To check C3, expand an arbitrary $|\psi\rangle$ in the graph basis. Then $T|\psi\rangle = |\psi\rangle$ for all $T \in \mathcal{S}$ means that all coefficients must vanish for graph basis states that do not belong to $\mathcal{C}$. Hence C3 is satisfied if and only if $\mathcal{C}$ is G-additive.

The preceding analysis generalizes immediately to $n > 1$ in the case of the trivial graph G^0 with no edges. A graph code $\mathcal{C}$ has a basis of the form $\{Z^{\boldsymbol{c}}|G^0\rangle\}$ for a collection C of integer n-tuples $\boldsymbol{c} \in \mathbb{Z}_D^n$, and is G-additive when the collection $C = \{\boldsymbol{c}\}$ is closed under component-wise addition mod D, i.e., is a subgroup of $\mathbb{Z}_D^n$. Whether or not $\mathcal{C}$ is G-additive, the subgroup $\mathcal{S}$ of $\mathcal{P}$ satisfying C1 and C2 consists of all operators of the form $X^{\boldsymbol{s}} = X_1^{s_1} X_2^{s_2} \cdots$ with the n-tuple $\boldsymbol{s}$ satisfying

$$\boldsymbol{c} \cdot \boldsymbol{s} := \sum_{l=1}^{n} c_l s_l \equiv \boldsymbol{0} \pmod{D} \qquad (24)$$

for every $\boldsymbol{c} \in C$. Just as for $n = 1$, $\mathcal{S}$ cannot contain Pauli products with (nontrivial) powers of Z operators. Let S denote the collection of all such $\boldsymbol{s}$. The linearity of (24) means S is an additive subgroup of $\mathbb{Z}_D^n$.

One can also regard (24) as a set of conditions, one for every $\boldsymbol{s} \in S$, that are satisfied by certain $\boldsymbol{c} \in \mathbb{Z}_D^n$. The set C' of all these solutions is itself an additive subgroup of $\mathbb{Z}_D^n$, and contains C. In App. D we show that $C' = C$ if and only if C (the collection we began with) is an additive subgroup of $\mathbb{Z}_D^n$, and when this is the case the sizes of C and S are related by

$$|C| \cdot |S| = D^n. \qquad (25)$$

Just as for $n = 1$, any $X^{\boldsymbol{s}}$ maps a graph basis state for the trivial graph G^0—they are all product states—onto itself up to a multiplicative phase, and the same argument used above for $n = 1$ shows that C3 is satisfied for all $T \in \mathcal{S}$ if and only if $\mathcal{C}$ is G-additive.

To apply these results to a general graph G on n qubits, note that the unitary $\mathcal{U}$ defined in (7) provides, through (6) and (10), a one-to-one map of the graph basis states of the trivial G^0 onto the graph basis states of G. At the same time the one-to-one map $\mathcal{U}P\mathcal{U}^\dagger$ carries the $\mathcal{S}$ satisfying C1 and C2 (and possibly C3) for the G^0 code to the corresponding $\mathcal{S}$, satisfying the same conditions for the G code. (The reverse maps are obtained by interchanging $\mathcal{U}^\dagger$ and $\mathcal{U}$.) Consequently, the results obtained for G^0 apply at once to G, and the transformation allows the elements of the stabilizer for the G graph code to be characterized by integer n-tuples $\boldsymbol{s}$ satisfying (24). Thus we have shown that G-additive codes are stabilizer codes, and for these the coding space and stabilizer group descriptions are dual, related by (24): each can be derived from the other.

VI. CONCLUSION AND DISCUSSION

In this paper we have developed an approach to graph codes which works for qudits with general dimension D, and employs graphical methods to search for specific examples of such codes. It is similar to the approaches developed independently in [13, 14, 15]. We have used it for computer searches on graphs with a relatively small number n of qudits, and also to construct certain families of graphs yielding optimum distance $\delta = 2$ codes for various values of D and n which can be arbitrarily large. It remains a challenging problem to do the same for codes with distance $\delta > 2$.

In a number of cases we have been able to construct what we call quantum Singleton (QS) codes that saturate the quantum Singleton bound [2]: these include the $\delta = 2$ codes for arbitrarily large n and D mentioned above, and also a number of $\delta = 3$ codes in the case of $D = 3$ (qutrits), see Tables I and II. The results for cycle graphs for $D = 3$ and $\delta = 3$ in Table I are interesting in that the QS bound is saturated for $n \leq 10$, but fails for $n = 11$, as it must for nondegenerate codes; see the discussion in Sec. IV C. Our results are consistent with the difficulty of finding QS codes for larger δ [23], but suggest that increasing D may help, as observed in [7]. It is worth noting that we have managed to construct many of the previously known nonadditive codes, or at least codes with the same $((n, K, \delta))_D$, using simple graphs. Some other nonadditive codes not discussed here, such as the $((10, 24, 3))_2$ code in [14], can also be obtained from suitably chosen graphs. While all these results are encouraging, they represent only a beginning in terms of understanding what properties of graphs lead to good graph codes, and how one might efficiently construct such codes with arbitrarily large n and δ, for various D.

As noted in Sec. III B, all graph codes with distance $\delta \leq \Delta'$, where Δ' is the diagonal distance of the graph, are necessarily nondegenerate, and our methods developed for such codes will (in principle) find them all. All codes with $\delta > \Delta'$ are necessarily degenerate codes, and their systematic study awaits further work. It should be noted that our extension of graph codes to $D > 2$ is based on extending Pauli operators in the manner indicated in [31]. Though the extension seems fairly natural, and it is hard to think of alternatives when D is prime, there are other ways to approach the matter when D is composite (including prime powers), which could yield larger or at least different codes, so this is a matter worth exploring.

The relationship between stabilizer (or additive) codes and G-additive (as defined in Sec. III B) graph codes has been clarified by showing that they are dual representations, connected through a simple equation, (24), of the same thing. One might suspect that such duality extends to nongraphical stabilizer codes, but we have not studied the problem outside the context of graph codes. Nonadditive codes, which—if one uses our definition, Sec. V—do not have stabilizers, are sometimes of larger size than additive codes, so they certainly need to be taken into

account in the search for optimal codes. The graph formalism employed here works in either case, but computer searches are much faster for additive codes.

APPENDIX A: THE X-Z RULE AND RELATED

X-Z Rule. *Acting with an X operator on the i'th qudit of a graph state $|G\rangle$ produces the same graph basis state as the action of Z operators on the neighbors of qudit i, raised to the power given by the edge multiplicities Γ_{im}.*

The operator X_i commutes with C_{lm} when $i \neq l$ and $i \neq m$, but if $i = l$ (or similarly $i = m$) one can show using (5) and (1) that

$$X_l C_{lm} = C_{lm} Z_m X_l = Z_m C_{lm} X_l. \tag{A1}$$

That is, an X_i operator can be pushed from left to right through a C_{lm} with at most the cost of producing a Z operator associated with the *other* qudit: if $i = l$ one gets Z_m, if $i = m$ one gets Z_l. Since all Z commute with all C, one can place the resulting Z_m either to the left or to the right of C_{lm}.

Now consider pushing X_i from the left to the right through $\mathcal{U}$, the product of C_{lm} operators defined in (7). Using (A1) successively for those C_{lm} that do not commute with X_i, one sees that this can be done at the cost of generating a Z_m for every edge of the graph connecting i to another vertex m. Let the product of these be denoted as $\hat{Z} := \prod_{(l=i,m)\in E} Z_m^{\Gamma_{lm}}$. Then, with definition (6), we can show

$$\begin{aligned} X_i |G\rangle = X_i \mathcal{U} |G^0\rangle &= \hat{Z}\mathcal{U}X_i |G^0\rangle \\ &= \hat{Z}\mathcal{U} |G^0\rangle = \hat{Z} |G\rangle, \end{aligned} \tag{A2}$$

which completes the proof of the X-Z Rule.

For graph codes satisfying (13), the X-Z Rule leads to the:

Coordination bound. *The diagonal distance Δ' for a graph G cannot exceed $\nu+1$, where ν is the minimum over all vertices of the number of neighbors of a vertex, this being the number of vertices joined to the one in question by edges, possibly of multiplicity greater than 1.*

To make the counting absolutely clear consider Fig. 1, where the vertex on the left has 3 neighbors, and each of the others has 1 neighbor, so that in this case $\nu = 1$. To derive the bound, apply X to a vertex which has ν neighbors. By the X-Z rule the result is the same as applying appropriate powers of Z to each neighbor. Let P be this X tensored with appropriate compensating powers of Z at the neighboring vertices in such a way that $P |G\rangle = |G\rangle$. The size of P is $\nu + 1$, and Δ' can be no larger.

Another useful result follows from the method of proof of the X-Z Rule:

Paulis to Paulis. *Let P be a Pauli product* (3), *and for $\mathcal{U}$ defined in* (7) *let*

$$P' = \mathcal{U}^\dagger P \mathcal{U}, \quad P'' = \mathcal{U}P\mathcal{U}^\dagger. \tag{A3}$$

Then both P' and P'' are Pauli products.

To see why this works, rewrite the first equality as $\mathcal{U}P' = P\mathcal{U}$, and imagine pushing each of the single qudit operators, of the form $X_j^{\mu_j} Z_j^{\nu_j}$, making up the product P through $\mathcal{U}$ from left to right. This can always be done, see the discussion following (A1), at the cost of producing some additional Z operators, which can be placed on the right side of $\mathcal{U}$, to make a contribution to P'. At the end of the pushing the final result can be rearranged in the order specified in (3) at the cost of some powers of ω, see (2). The argument for P'' uses pushing in the opposite direction.

APPENDIX B: PARTITION THEOREM PROOF

Given the partition of the n qudits into sets V_1 and V_2 containing n_1 and n_2 elements, the code of interest consists of the graph basis states $|\boldsymbol{c}\rangle = |c_1, c_2, \ldots, c_n\rangle$ satisfying the two conditions

$$\sum_{i\in V_1} c_i \equiv 0 \pmod{D} \tag{B1}$$

$$\sum_{j\in V_2} c_j \equiv 0 \pmod{D} \tag{B2}$$

This code is additive and contains $K = D^{n_1-1} \times D^{n_2-1} = D^{n-2}$ codewords. (The counting can be done by noting that (B1) defines a subgroup of the additive group $\mathbb{Z}_D^{n_1}$, and its cosets are obtained by replacing 0 with some other integer on the right side of (B1).)

We first demonstrate that this code has $\delta \geq 2$ by showing that any Pauli operator, except the identity, applied to a single qudit maps a codeword into a graph basis state not in the code. If Z^ν for $0 < \nu < D$ is applied to a qudit in V_1, the effect will be to replace 0 on the right side of (B1) with ν, so this graph state is not in the code. If X^μ, $0 < \mu < D$ is applied to a qudit in V_1 the result according to the X-Z Rule, App. A, will be the same as placing Z operators on neighboring qudits in V_2 (as well as V_1) in such a way that 0 on the right side of (B2) is replaced by $g\mu$, where g is the total number of edges (including multiplicities) joining the V_1 qudit with qudits in V_2. But as long as g is coprime to D, as specified in the condition for the theorem, $g\mu$ cannot be a multiple of D, and (B2) will no longer be satisfied. The same is true if $Z^\nu X^\mu$ is a applied to a qudit in V_1. Obviously the same

arguments work for Pauli operators applied to a single qudit in V_2. Thus we have shown that $\delta \geq 2$.

But $\delta > 2$ is excluded by the QS bound, so we conclude that we have an additive code of $K = D^{n-2}$ elements and distance $\delta = 2$ that saturates the QS bound.

APPENDIX C: CONSTRUCTION OF QUBIT STAR GRAPH CODES

As noted in Sec. IV B a star graph for n-qubits consists of a central vertex joined by edges to $n-1$ peripheral vertices. Let V_1 be the central vertex and V_2 the set of peripheral vertices. When n is even and $D = 2$ the conditions of the Partition theorem, Sec. IV B, are satisfied, and the $\delta = 2$ code constructed in App. B consists of the 2^{n-2} graph basis states with no Z on the central qubit and an even number r of Z's on the peripheral qubits, thus satisfying (B1) and (B2), and yielding an additive QS code.

When n is odd the central vertex is connected to an even number $n - 1$ of vertices in V_2, so the conditions of the Partition theorem no longer hold. A reasonably large $\delta = 2$ nonadditive code can, however, be constructed by again assuming no codeword has Z on the central qubit, and that the code contains all graph basis states with r Z's on the peripheral qubits *for a certain selected set R of r values.*

The set R must satisfy two conditions. First, it cannot contain both r and $r + 1$, because applying an additional Z to a codeword with r Z's yields one with $r+1$, and one cannot have both of them in a code of distance $\delta = 2$. Second, applying X to the central vertex and using the X-Z rule, App. A, maps a codeword with r Z's to one with $r' = n - 1 - r$; hence R cannot contain both r and $n-1-r$. For example, when $n = 7$ ($n-1 = 6$ peripheral qubits) the set $R = \{0, 2, 5\}$ satisfies both conditions, as does $R = \{1, 4, 6\}$, whereas $R = \{1, 2, 6\}$ violates the first condition and $R = \{1, 3, 5\}$ the second.

By considering examples of this sort, and noting that the number of such graph basis states with r Z's is $\binom{n-1}{r}$ which is equal to $\binom{n-1}{n-1-r}$, one sees that for n odd one can construct in this way a nonadditive code with

$$\sum_{i=0}^{(n-3)/2} \binom{n-1}{i} = 2^{n-2} - \frac{1}{2}\binom{n-1}{(n-1)/2} \qquad \text{(C1)}$$

codewords.

APPENDIX D: SOLUTIONS TO $\boldsymbol{c} \cdot \boldsymbol{s} \equiv 0$ (mod D)

Let $\mathcal{A}$ be the collection of all n-component integer vectors (i.e., n-tuples) of the form $\boldsymbol{a} = (a_1, a_2, \ldots a_n)$, $0 \leq a_j \leq D - 1$, with component-wise sums and scalar multiplication defined using arithmetic operations mod D. In particular, $\mathcal{A}$ is a group of order D^n under component-wise addition mod D. We shall be interested in subsets C and $\mathcal{S}$ of $\mathcal{A}$ that satisfy

$$\boldsymbol{c} \cdot \boldsymbol{s} := \sum_{l=1}^{n} c_l s_l \equiv 0 \pmod{D} \qquad \text{(D1)}$$

for all $\boldsymbol{c} \in C$ and $\boldsymbol{s} \in S$. Given some collection C, we shall say that S is *maximal* relative to C if it includes *all* solutions $\boldsymbol{s}$ that satisfy (D1) for every $\boldsymbol{c} \in C$. It is easily checked that a maximal S is an additive subgroup of $\mathcal{A}$: it includes the zero vector and $-\boldsymbol{s}$ mod D whenever $\boldsymbol{s} \in S$. A similar definition holds for C being maximal relative to a given S. We use $|C|$ to denote the number of elements in a set or collection C.

Theorem. *Let C be an additive subgroup of $\mathcal{A}$, and let S be maximal relative to C, i.e., the set of all $\boldsymbol{s}$ that satisfy* (D1) *for every $\boldsymbol{c} \in C$. Then C is also maximal relative to S, and*

$$|C| \cdot |S| = D^n. \qquad \text{(D2)}$$

The proof is straightforward when D is a prime, since $\mathbb{Z}_D$ is a field, and one has the usual rules for a linear space. The composite case is more difficult, and it is useful to start with $n = 1$:

Lemma. Let C be a subgroup under addition mod D of the integers lying between 0 and $D - 1$, and S all integers in the same range satisfying

$$cs \equiv 0 \pmod{D} \qquad \text{(D3)}$$

for every $c \in C$. Then C consists of *all* integers c in the range of interest which satisfy (D3), and $|C| \cdot |S| = D$.

When $C = \{0\}$ the proof is obvious, since $|C| = 1$ and $|S| = D$. Otherwise, because it is an additive subgroup of $\mathbb{Z}_D$, C consists of the multiples $\{\mu c_1\}$ of the smallest positive integer c_1 in C, necessarily a divisor of D, when μ takes the values $0, 1, \ldots s_1 - 1$, where $s_1 = D/c_1$. One quickly checks that all integer multiples $s = \nu s_1$ of this s_1 satisfy (D3) and are thus contained in S. But S is also an additive subgroup, and s_1 is its minimal positive element (except in the trivial case $c_1 = 1$), for were there some smaller positive integer s' in S we would have $0 < c_1 s' < D$, contradicting (D3). Similarly there is no way to add any additional integers to C while preserving the subgroup structure under addition mod D without including a positive c less than c_1, which will not satisfy (D3) for $s = s_1$.

For $n > 1$ it is helpful to use a *generator matrix F*, with components F_{rl}, each between 0 and $D - 1$, with the property that $\boldsymbol{c} \in C$ if and only if it can be expressed as linear combinations of rows of F, i.e.,

$$c_l \equiv \sum_r b_r F_{rl} \pmod{D} \qquad \text{(D4)}$$

for a suitable collection of integers $\{b_r\}$. This collection will of course depend on the $\boldsymbol{c}$ in question, and for a given

c need not be unique, even assuming (as we shall) that $0 \leq b_r \leq D - 1$. In particular the matrix F for which each row is a distinct c in C, with r running from 1 to $|C|$, is a generator matrix. It is straightforward to show that if F is any generator matrix for C, S consists of all solutions s to the equations

$$\sum_{l=1}^{n} F_{rl} s_l \equiv 0 \pmod{d} \text{ for } r = 1, 2, \ldots. \quad \text{(D5)}$$

The collections C and S, vectors of the form (D4) and those satisfying (D5), remain the same if F is replaced by another generator matrix F' obtained by one of the following *row operations*: (i) permuting two rows; (ii) multiplying (mod D) any row by an *invertible* integer, i.e., an integer which has a multiplicative inverse mod D; (iii) adding (mod D) to one row an *arbitrary* multiple (mod D) of a different row; (iv) discarding (or adding) any row that is all zeros, to get a matrix of a different size. Of these, (i) and (iv) are obvious, and (ii) is straightforward. For (iii), consider what happens if the second row of F is added to the first, so that $F'_{rl} = F_{rl}$ except for

$$F'_{1l} \equiv F_{1l} + F_{2l} \pmod{D}. \quad \text{(D6)}$$

Then setting

$$b'_1 = b_1, \; b'_2 \equiv b_2 - b_1 \pmod{d}, \; b'_l = b_l \text{ for } l \geq 3 \quad \text{(D7)}$$

leads to the same c in (D4) if b and F are replaced by b' and F' on the right side. Likewise, any c that can be written as a linear combination of F' rows can be written as a combination of those of F, so the two matrices generate the same collection C, and hence have the same solution set S to (D5). Since adding to one row a different row can be repeated an arbitrary number of times, (iii) holds for an arbitrary (not simply an invertible) multiple of a row.

The corresponding column operations on a generator matrix are (i) permuting two columns; (ii) multiplying a column by an invertible integer; (iii) adding (mod D) to one column an arbitrary multiple (mod D) of a different column. Throwing away (or adding) columns of zeros is *not* an allowed operation. When column operations are carried out to produce a new F' from F, the new collections C' and S' obtained using (D4) and (D5) will in general be different, but C' is an additive subgroup of the same size (order), $|C'| = |C|$, and likewise $|S'| = |S|$. The argument is straightforward for (i) and (ii), and for (iii) it is an easy exercise to show that if the second column of F is added to the first to produce F', the collection C is mapped into C' by the map

$$c'_1 \equiv c_1 + c_2 \pmod{D}; \quad c'_l = c_l \text{ for } l \geq 2 \quad \text{(D8)}$$

whose inverse will map C' into C when one generates F from F' by subtracting the second column from the first. Thus $|C| = |C'|$. The same strategy shows that $|S'| = |S|$; instead of (D8) use $s'_2 \equiv s_2 - s_1 \pmod{D}$, and $s'_l = s_l$ for $l \neq 2$.

The row and column operations can be used to transform the generator matrix to a (non unique) diagonal form, in the following fashion. If each F_{rl} is zero the problem is trivial. Otherwise use row and column permutations so that the smallest positive integer f in the matrix is in the upper left corner $r = 1 = l$. Suppose f does not divide some element, say F_{13}, in the first row. Then by subtracting a suitable multiple of the first column from the third column we obtain a new generator F' with $0 < F'_{13} < f$, and interchanging the first and third columns we have a generator with a smaller, but still positive, element in the upper left corner. Continue in this fashion, considering both the first row and the first column, until the upper left element of the transformed generator divides *every* element in both. When this is the case, subtracting multiples of the first column from the other columns, and multiples of the first row from the other rows, will yield a matrix with all zeros in the first row and first column, apart from the nonzero upper left element at $r = 1 = l$, completing the first step of diagonalization.

Next apply the same overall strategy to the sub matrix obtained by ignoring the first row and column. Continuing the process of diagonalization and discarding rows that are all zero (or perhaps adding them back in again), one arrives at a diagonal $n \times n$ generator matrix

$$\hat{F}_{rl} = f_l \delta_{rl}, \quad \text{(D9)}$$

where some of the f_l may be zero. The counting problem is now much simplified, because for each l c_l can be any multiple mod D of f_l, and s_l any solution to $f_l s_l \equiv 0$ (mod D), independent of what happens for a different l. Denoting these two collections by C_l and S_l, the lemma implies that $|C_l| \cdot |S_l| = D$ for every l, and taking the product over l from 1 to n yields (D2). This in turn implies that C consists of *all possible* c that satisfy (D1) for all the $s \in S$. To see this, note that the size $|C|$ of C is $D^n/|S|$. If we interchange the roles of C and S in the above argument (using a generator matrix for S, etc.), we again come to the result (D2), this time interpreting $|C|$ as the number of solutions to (D1) with S given. Thus since it cannot be made any larger, the original additive subgroup C we started with is maximal relative to S. This completes the proof.

[1] P. W. Shor, Phys. Rev. A **52**, R2493 (1995).

[2] E. Knill and R. Laflamme, Phys. Rev. A **55**, 900 (1997).

[3] A. M. Steane, Phys. Rev. Lett. **77**, 793 (1996).

[4] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000), 5th ed.

[5] E. M. Rains, IEEE Trans. Inf. Theory **45**, 1827 (1999).

[6] A. Ashikhmin and E. Knill, IEEE Trans. Inf. Theory **47**, 3065 (2001).

[7] D. Schlingemann and R. F. Werner, Phys. Rev. A **65**, 012308 (2002).

[8] D. Schlingemann, Quantum Info. Comp. **2**, 307 (2002), arXiv:quant-ph/0111080.

[9] D. Schlingemann, Quantum Info. Comp. **3**, 431 (2003), arXiv:quant-ph/0202007.

[10] M. Grassl, T. Beth, and M. Roetteler, Int. J. Quantum Inf. **2**, 55 (2004).

[11] V. Arvind, P. P. Kurur, and K. R. Parthasarathy, arXiv:quant-ph/0210097.

[12] R. Raussendorf and H. J. Briegel, Phys. Rev. Lett. **86**, 5188 (2001).

[13] A. Cross, G. Smith, J. A. Smolin, and B. Zeng, arXiv:0708.1021 [quant-ph].

[14] S. Yu, Q. Chen, and C. H. Oh, arXiv:0709.1780 [quant-ph].

[15] D. Hu, W. Tang, M. Zhao, Q. Chen, S. Yu, and C. Oh, arXiv:0801.0831 [quant-ph].

[16] J. A. Smolin, G. Smith, and S. Wehner, Phys. Rev. Lett. **99**, 130505 (2007), arXiv:quant-ph/0701065.

[17] See [31] for a list of references to work that employs operators of this type.

[18] While there seems to be no proof that the $((6, 2, 3))_2$ degenerate code has a larger K than any nondegenerate code with $n = 6$ and $\delta = 3$, some support comes from the fact that we performed an exhaustive search of all graphs with 6 vertices and did not find a nondegenerate graph code with $\delta = 3$ and $K > 1$. But the notion that this degenerate code is superior to nondegenerate codes is undercut by the observation that the well known non-degenerate $((5, 2, 3))_2$ code uses only 5 instead of 6 qubits to achieve equal values of K and δ.

[19] A. R. Calderbank, E. M. Rains, P. M. Shor, and N. J. A. Sloane, IEEE Trans. Inf. Theory **44**, 1369 (1998), arXiv:quant-ph/9608006.

[20] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes* (North Holland, 1977).

[21] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W. H. Freeman, 1979).

[22] R. Carraghan and P. M. Pardalos, Operations Research Letters **9**, 375 (1990).

[23] M. Grassl, *Bounds on the minimum distance of linear codes*, Available online at http://www.codetables.de (2007).

[24] A. Ketkar, A. Klappenecker, S. Kumar, and P. K. Sarvepalli, IEEE Trans. Inf. Theory **51**, 4892 (2006).

[25] We omit the details. In some but not all cases one can use the Partition theorem with V_1 and V_2 the center and the peripheral vertices. Allowing some double edges when $D > 2$ extends the range of n values where the Partition theorem can be employed.

[26] E. M. Rains, R. H. Hardin, P. W. Shor, and N. J. A. Sloane, Phys. Rev. Lett. **79**, 953 (1997), arXiv:quant-ph/9703002.

[27] S. Yu, Q. Chen, C. H. Lai, and C. H. Oh, arXiv:0704.2122 [quant-ph].

[28] Since the distance $\delta = 3$ does not exceed the diagonal distance $\Delta' = 3$ for this graph, a graph code is necessarily nondegenerate, see Sec. III B, and hence the quantum Hamming bound—see p. 444 of [4]—extended to $D = 3$ applies, and this yields an upper bound of $K \leq 1990$.

[29] D. Gottesman, arXiv:quant-ph/9705052.

[30] M. Bahramgiri and S. Beigi, arXiv:quant-ph/0610267.

[31] E. Hostens, J. Dehaene, and B. De Moor, Phys. Rev. A **71**, 042315 (2005), arXiv:quant-ph/0408190.