Dimensional jump in quantum error correction

Bombin Palomo, Hector

Published in:
New Journal of Physics

DOI:
10.1088/1367-2630/18/4/043038

Publication date:
2016

Document version
Publisher's PDF, also known as Version of record

Citation for published version (APA):
Bombin Palomo, H. (2016). Dimensional jump in quantum error correction. *New Journal of Physics*, *18*(4), [043038]. https://doi.org/10.1088/1367-2630/18/4/043038

Dimensional jump in quantum error correction

2016 New J. Phys. 18 043038

(http://iopscience.iop.org/1367-2630/18/4/043038)

New Journal of Physics
The open access journal at the forefront of physics

Published in partnership with: Deutsche Physikalische Gesellschaft and the Institute of Physics

OPEN ACCESS

RECEIVED
13 November 2015

REVISED
8 March 2016

ACCEPTED FOR PUBLICATION
8 April 2016

PUBLISHED
27 April 2016

PAPER

Dimensional jump in quantum error correction

Héctor Bombín

Yukawa Institute for Theoretical Physics, Kitashirakawa Oiwakecho, Sakyo-ku, 606-8502 Kyoto, Japan
Deparment of Mathematical Sciences, University of Copenhagen, Universitetsparken 5, DK-2100 Copenhagen Ø, Denmark

Keywords: topological stabilizer code, gauge color code, single-shot error correction, gauge fixing, transversal gate, quantum-local operation

Abstract

Topological stabilizer codes with different spatial dimensions have complementary properties. Here I show that the spatial dimension can be switched using gauge fixing. Combining 2D and 3D gauge color codes in a 3D qubit lattice, fault-tolerant quantum computation can be achieved with constant time overhead on the number of logical gates, up to efficient global classical computation, using only local quantum operations. Single-shot error correction plays a crucial role.

1. Introduction

Quantum error correction methods [1] that emphasize locality [2] constitute nowadays the most promising approach for the practical implementation of a quantum computer. In particular, topological stabilizer codes [3] receive a good deal of attention due to their flexibility and relative simplicity. 2D topological stabilizer codes are potentially easiest to implement, but low dimensionality severely constrains the operations that can be performed locally [4]. 3D codes do not suffer from such obstructions [5], but require many more qubits, among other drawbacks. The purpose of this work is to bring together the best of the two worlds by providing a bridge between them: a procedure to switch back and forth between 2D and 3D codes.

Among 2D topological stabilizer codes color codes [6] are optimal in terms of the local implementation of gates. Namely, all Clifford gates are transversal, i.e. act individually on the physical qubits composing the code (or pair-wise for two-qubit logical gates). See [7] for a recent single-qubit implementation. Unfortunately Clifford gates are not enough for universal computation, but this is all that 2D topological stabilizer codes can offer [4, 8]. The way out is to either resort to complementary techniques that increase the amount of resources needed [9], to consider more complicated codes [3], or to increase the spatial dimension.

3D (gauge) color codes [10] are 3D topological stabilizer codes with many remarkable characteristics that, put together, enable fault-tolerant quantum computation with *quantum-local* elementary operations, i.e. involving only a finite depth local quantum circuit aided with global classical information processing [11]. This comes at a cost: spatial locality can only be attained in 4D due to two-qubit logical gates. In addition, 3D color codes require $O(n^{3/2})$ qubits to correct the same number of errors as an n-qubit 2D color code.

Dimensional jumps solve these problems, at least to a large extent. As the name suggests, in a dimensional jump the spatial dimension of a local code is switched *in constant time*, or more precisely via a quantum-local operation, where locality refers to a 3D layout. In particular, the procedure allows to switch back and forth fault-tolerantly between 3D and 2D color codes.

Dimensional jumps make use of the principles of *single-shot error correction* [11, 12], i.e. quantum-local fault-tolerant error correction. Single-shot error correction plays a key role in the quantum-locality of operations in 3D color codes. This is again the case for dimensional jumps, which involve error correction.

Equipped with dimensional jumps one can envision the 3D-local fault-tolerant quantum computing layout of figure 1. The starting point is a stack of 2D color codes, analogous to the one proposed for toric codes in [2]. Each layer encodes a single logical qubit, and all Clifford gates can be perfomed transversally. On one extreme of the stack sits a 3D color code lattice, and the 2D color code sitting next to it can be converted back and forth into a 3D color code. As a 2D code it can be part of the Clifford gates occurring in the stack, and as a 3D code it

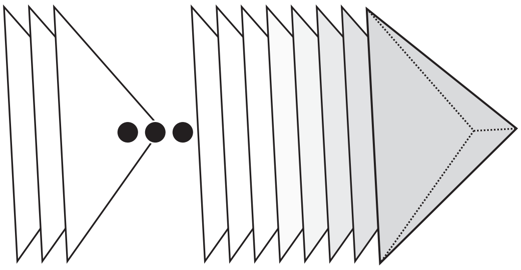

Figure 1. A 3D layout for fault-tolerant quantum computation. Each layer of the stack is a 2D color code encoding a logical qubit. On one extreme sits a 3D color code lattice. The stack acts as a memory. Computations happen on the 3D end.

becomes isolated from the other 2D codes but a non-Clifford gate can be implemented, achieving universality [5].

An advantage of the layout is that all logical qubits but one are encoded in 2D, dramatically reducing the resources when compared with an all-3D encoding. Also important is that all elementary operations are quantum-local. As a drawback, the 3D capabilities are only available in one location, and therefore parallel computation is lost[1]. The time overhead is still constant on the number of logical gates, and for this it is enough to be able to perform swap gates in the stack, with computations confined to neighbors of the 3D code. Finally, it is worth noting that, for the 2D and 3D constructions of [10], the required elementary measurements involve at most six physical qubits (plus any ancillas used).

2. Background

This section summarizes and rephrases previous results that will be needed later.

2.1. Stabilizer codes

A stabilizer subsystem code [13] on n physical qubits is defined by two subgroups $\mathcal{S}$, $\mathcal{G}$ of the Pauli group of operators on n qubits. The stabilizer group $\mathcal{S}$ defines the code subspace where quantum information is encoded: encoded states are eigenstates, with eigenvalue $+1$, of all the elements of $\mathcal{S}$. The gauge group $\mathcal{G}$ generates the algebra of operators that do not disturb encoded information. The groups $\mathcal{S}$ and $\mathcal{G}$ satisfy

$$-1 \not\in \mathcal{S}, \qquad \mathcal{S} \propto \mathcal{G} \bigcap \mathcal{Z}(\mathcal{G}), \tag{1}$$

where $\mathcal{Z}(\mathcal{A})$ denotes the centralizer of $\mathcal{A}$ in the Pauli group. The operators in the group $\mathcal{Z}(\mathcal{G})$ are (bare) logical (Pauli) operators: they transform encoded states while preserving the code subspace. Logical operators that are equivalent up to stabilizers have the same action on encoded qubits. Therefore it is convenient to choose a representative group of logical operators $\mathcal{L} \subseteq \mathcal{Z}(\mathcal{G})$ such that each element of $\mathcal{L}$ belongs to a different class of the quotient $\mathcal{Z}(\mathcal{G})/\mathcal{S}$.

2.2. Error correction

Error correction is the procedure that attempts to remove the errors that a code has suffered. *Ideally* it amounts to

(a) measure a set of generators of $\mathcal{S}$ (the result is the *syndrome* σ of $\mathcal{S}$), and

(b) apply a Pauli operator E that yields an encoded state (such E is said to have syndrome σ).

The operator E anticommutes with the generators with negative eigenvalue outcome. Its choice should minimize residual logical errors.

When the stabilizer generators are local the above ideal process is quantum-local. However, in practice error correction is itself noisy, and often in making the process fault-tolerant quantum-locality is lost. Surprisingly, for some codes quantum-locality can be preserved: they allow single-shot error correction [11].

[1] If the number of physical qubits is less of a concern, one can always consider a network of 3D codes connected by 2D codes. In such a setting parallel gates are again possible, subject to the locality constraints imposed by the lattice structure.

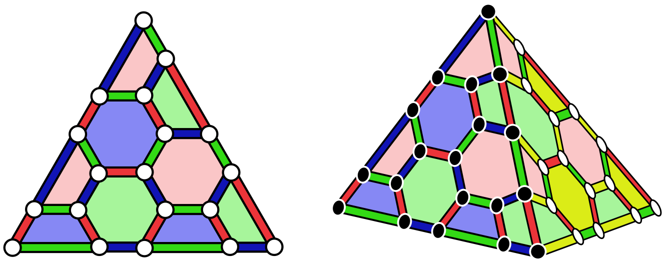

Figure 2. (Left) A triangular two-colex. Plaquettes appear in their complementary color, i.e. *rg*-plaquettes are colored blue. (Right) A tetrahedral three-colex. Cells appear in their complementary color, i.e. *rgb*-plaquettes are colored yellow. Every facet is a triangular two-colex. The outer vertices (*rgb*-facet) are marked in black, the rest are inner vertices.

2.3. Gauge fixing

Gauge fixing is a procedure [14] that allows to switch back and forth between two codes $\mathcal{S}$, $\mathcal{G}$ and $\mathcal{S}'$, $\mathcal{G}'$ if [10] they share a representative group of logical operators $\mathcal{L}$ and

$$\mathcal{S} \subseteq \mathcal{S}' \tag{2}$$

or, equivalently (up to a choice of signs for $\mathcal{S}$ and $\mathcal{S}'$)

$$\mathcal{G}' \subseteq \mathcal{G}. \tag{3}$$

Any encoded state of $\mathcal{S}'$ is also an encoded state for $\mathcal{S}$. Transforming an encoded state of $\mathcal{S}$ into an encoded state of $\mathcal{S}'$ is called gauge fixing. The procedure is similar to error correction: *ideally* it amounts to

(a) extract the syndrome σ of $\mathcal{S}'$, and

(b) apply some operator $E \in \mathcal{G}$ with syndrome σ.

The syndrome σ is trivial for elements of $\mathcal{S}$, and E is unique up to elements of $\mathcal{G}'$.

2.4. Splitting

As a particular case of gauge fixing, consider that the code $\mathcal{S}'$, $\mathcal{G}'$ actually consists of several codes with gauge groups $\mathcal{S}_i$, $\mathcal{G}_i$ defined on *disjoint* sets of n_i qubits each. Together they form the code on $n = \sum_i n_i$ qubits

$$\mathcal{S}' = \prod_i \mathcal{S}_i, \qquad \mathcal{G}' = \prod_i \mathcal{G}_i, \tag{4}$$

where it is implicitly assumed that all operators have been suitably tensored with identities to act on the n qubits. If the codes have logical representative groups $\mathcal{L}_i$, we might choose for the n qubit code the group

$$\mathcal{L} := \prod_i \mathcal{L}_i. \tag{5}$$

Any code $\mathcal{S}$, $\mathcal{G}$ on the n qubits that has logical group $\mathcal{L}$ and satisfies conditions (2, 3) can be gauge fixed to $\mathcal{S}'$, $\mathcal{G}'$ (at least ideally). In this case gauge fixing amounts to *splitting* the n-qubit code into several pieces, each with some of the original physical and logical qubits. Conversely, putting together the pieces yields an encoded state of $\mathcal{S}$, $\mathcal{G}$.

2.5. Colexes

2D color codes [6] are defined on *two-colexes*. These are 2D trivalent lattices with three-colored edges in which *plaquettes* (two-cells) have edges of two colors. The two-colexes considered here are triangular. In particular, each side of the triangle has edges in different combinations of two colors, see figure 2. A plaquette with red and green edges is a *rg*-plaquette, a blue edge is a *b*-edge, etc.

3D color codes [5] are defined on *three-colexes*. These are 3D tetravalent lattices with four-colored edges, in which plaquettes have edges of two colors and *cells* (three-cells) have edges of three colors. The three-colexes considered here as a starting point are tetrahedral. In particular, each facet of the tetrahedron has edges in different combinations of three colors, see figure 2.

In the present work the specific choice of two- and three-colexes is not relevant. For detailed constructions and pictures, the reader is referred to the literature [5, 10, 12, 15, 16].

2.6. Color codes

2D color codes and 3D gauge color codes [10] are self-dual CSS topological stabilizer codes, i.e. the generators of the stabilizer and gauge group are products either exclusively of bit-flip X operators or exclusively of phase-flip Z operators, with the same geometry for X- and Z-type generators. Therefore, the code is completely defined by the support of the generators.

Both in 2D and 3D there is one physical qubit per vertex of the colex. Denote the respective stabilizer and gauge groups $\mathcal{S}_2$, $\mathcal{G}_2$ and $\mathcal{S}_3$, $\mathcal{G}_3$. Let the support of an edge, plaquette or cell operator be the set of vertices of a given edge, plaquette or cell, respectively. E.g. a plaquette operator X_p flips the qubits of the plaquette p, and so on. Both $\mathcal{G}_2$ and $\mathcal{G}_3$ are generated by the set of all plaquette operators, something that will be highly relevant below. The stabilizers in general depend on the geometry of the code. In 2D triangular codes plaquette operators generate $\mathcal{S}_2$, and in 3D tetrahedral codes cell operators generate $\mathcal{S}_3$. In both cases the support of X and Z logical operators can be chosen to be the set of all qubits.

It is interesting to mention that error correction has been substantially explored for 2D color codes [17–24], whereas for 3D little is known [12].

3. Dimensional jumps

This section contains the main results of the paper. Namely, (i) that a tetrahedral 3D gauge color code can be split into a triangular 2D color code and another 3D gauge color code with no encoded qubits and (ii) that switching fault-tolerantly back and forth between the 2D and 3D codes only requires quantum-local operations. The generalization to higher dimensions is also briefly discussed.

3.1. Outer and inner codes

The starting point is a geometrical observation: each triangular facet of a tetrahedral three-colex is a triangular two-colex. Assume that such a three-colex and a distinguished facet are given. This facet will be denoted the *outer* (two-)colex. Conversely, the *inner* (three-)colex is composed of those vertices, edges, and cells not in contact with the outer colex. Cells/plaquettes with both inner and outer vertices are called *interface* cells/plaquettes.

The three-colex yields a 3D gauge color code $\mathcal{S}_3$, $\mathcal{G}_3$, and the outer colex yields a 2D color code $\mathcal{S}_2$, $\mathcal{G}_2$. Crucially for the results below, the 3D code admits logical operator representatives with support the set of all outer qubits, see appendix A. That is, the 3D gauge color code and the 2D color code have a common group $\mathcal{L}$ of logical operator representatives.

A 3D gauge color code $\mathcal{S}_\mathrm{in}$, $\mathcal{G}_\mathrm{in}$ can also be defined for the inner colex. Plaquette operators generate $\mathcal{G}_\mathrm{in}$ (by definition) and $\mathcal{S}_\mathrm{in}$ is generated by (i) cell operators and (ii) the restriction to the inner qubits of interface cell operators. The resulting code encodes no logical qubits, see appendix A.

3.2. Dimensional collapse

As observed above, (i) the inner code has no logical qubits, (ii) the outer 2D code and the 3D code share representative logical operators and (iii)

$$\mathcal{G}_2\,\mathcal{G}_\mathrm{in} \subseteq \mathcal{G}_3. \tag{6}$$

According to section 2.4 (with $\mathcal{G} = \mathcal{G}_3$ and $\mathcal{G}' = \mathcal{G}_2\mathcal{G}_\mathrm{in}$) the 3D code splits via gauge fixing in two pieces: the outer 2D code (that keeps the logical qubit) and the inner code. Moreover, due to the CSS structure (2) holds exactly, not up to a choice of signs:

$$\mathcal{S}_3 \subseteq \mathcal{S}_2\,\mathcal{S}_\mathrm{in}. \tag{7}$$

Denote by $\mathcal{G}_3|_2$ the group formed by the operators in $\mathcal{G}_3$ constrained to the outer qubits. Its generators are edge operators, the restriction of interface plaquettes to the outer code. Ideally, the dimensional jump from the 3D code to the 2D code amounts to

(1) discard inner qubits,

(2) extract the syndrome σ of $\mathcal{S}_2$, and

(3) apply some $E \in \mathcal{G}_3|_2$ with syndrome σ.

The operator E is unique up to elements of $\mathcal{S}_2$ and is a product of *string operators*. E.g. a b-string s is composed of outer b-edges e_i and might have endpoints at outer rg-plaquettes, see figure 3. The string operator X_s flips the qubits of the edges e_i and anticommutes with a plaquette operator Z_p if and only if p is an endpoint of s.

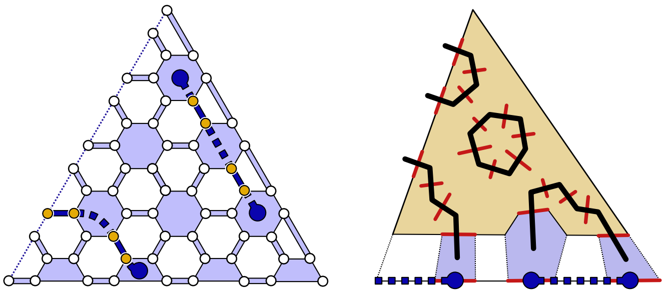

Figure 3. The relationship between b-strings (dotted-blue lines), rg-plaquette syndromes (blue circles) and flux lines (thick black). (Left) Two-colex with its b-edges and rg-plaquettes emphasized. A X_s string operator has support on the qubits along its string s. It anticommutes with Z plaquette operators at its endpoints. (Right) Schematic three-colex with the outer colex on the bottom and the inner colex shaded. Flux lines are composed of edges dual to rg-plaquettes (in red) and can have endpoints at inner rgy-cells (blue). Each b-string has the syndrome of a certain flux line.

3.3. Flux

The above procedure is not fault-tolerant, no matter how the syndrome σ is extracted: even if the gauge fixing process is perfect, most pre-existing single-qubit errors on outer qubits yield a residual logical error. To achieve fault-tolerance the key is to extract σ *indirectly* from the inner qubits. The resulting gauge fixing process is not only fault-tolerant but also quantum-local.

Let the outer colex be the rgb-facet of the tetrahedron, i.e. the facet with r-, g- and b-edges on it. Suppose that inner rg-plaquette Z_{p} operators are measured on an *encoded state* of $\mathcal{S}_3$. Each plaquette has a *dual* edge that pierces the plaquette connecting the centers of the cells meeting at the plaquette. The result of the measurement is codified as the set γ of edges dual to the inner rg-plaquettes with eigenvalue -1.

Given a cell c with rg-plaquettes p_i, every vertex in c belongs to exactly one of the p_i. Therefore

$$Z_{\mathrm{c}} = \prod_i Z_{\mathrm{p}_i}. \tag{8}$$

For the initial state $Z_{\mathrm{c}} = 1$ and thus γ has no *inner endpoints*, i.e. inner cells at which an odd number of edges meet. Therefore γ is a disjoint union of paths γ_i, or 'flux-lines', that are either closed or have endpoints (i) at the rgy-facet or (ii) at an interface rgy-cell, see figure 3.

Every interface rgy-cell c has a unique outer rg-plaquette p_{c} (the rest are inner, unless the colex is pathological), see figure 3. A flux-line is said to have an endpoint at p_{c} when it has an endpoint at c. According to (8) $Z_{\mathrm{p}_{\mathrm{c}}}$ has eigenvalue -1 if and only if p_{c} is the endpoint of and odd number of flux-lines γ_i. Thus the syndrome of $\mathcal{S}_2$ can be recovered from the measurement of inner rg-, gb- and rb-plaquette X and Z operators, which commute[2], see appendix A. The steps (1) and (2) above can be substituted by:

1+2 Obtain a syndrome σ from the destructive measurement of the plaquette operators in $\mathcal{G}_{\mathrm{in}}$ with colors matching the outer plaquettes.

3.4. Measurement errors

Consider again the measurement of inner rg-plaquette Z_{p} operators. Suppose that the original encoded state is noiseless but measurements can fail: instead of the correct dual edge set γ they yield $\gamma + \delta$, with $+$ the symmetric difference of sets: plaquette operators corresponding to edges in δ are assigned the wrong eigenvalue. The set $\gamma + \delta$ can have inner endpoints. Let δ_0 be a set of dual edges of minimal cardinality with the same inner endpoints (efficiently computable using perfect matching [2]). It provides an 'effective' set

$$\gamma_{\mathrm{eff}} \coloneqq \gamma + \delta + \delta_0 = \gamma + \bigsqcup_i \omega_i, \tag{9}$$

where $\delta + \delta_0$ decomposes as a disjoint union of flux-lines ω_i (because $\delta + \delta_0$ has, like γ_{eff}, no inner endpoints).

For every outcome γ of rg-flux-lines there is some operator E_γ that is a product of b-string X operators and has the syndrome corresponding to γ; it is unique up to stabilizers. By using γ_{eff} as input for the third gauge fixing step, instead of the correct operator E_γ we apply

$$E_{\gamma_{\mathrm{eff}}} \sim E_\gamma\, E_{\delta+\delta_0} \sim E_\gamma \prod_i E_{\omega_i}, \tag{10}$$

where the equivalence is up to stabilizers: measurement noise δ translates into errors E_{ω_i} at the final stage.

[2] It is worth emphasizing that the gauge fixing discussed here differs from the of one considered in [10], which involved all plaquette operators of either X or Z type, which do not commute.

How bad are these errors? Each E_{ω_i} is a b-string operator X_{s_i} that we can *choose* subject to the constraint that its endpoints on outer plaquettes should match those of ω_i. It follows by inspection of the different geometries, depicted in figure 3, that the number of qubits in the support of E_{ω_i} is, up to a constant depending on the lattice structure, smaller than the length $|\omega_i|$ of the flux-line. Moreover, at least half of the edges of ω_i belong to δ, rather than δ_0:

$$|\delta_0 + \omega_i| = |\delta_0| - |\omega_i \bigcap \delta_0| + |\omega_i \bigcap \delta| \geqslant |\delta_0| \tag{11}$$

because $\delta_0 + \omega_i$ has the same inner endpoints as δ_0. Thus an error E_{ω_i} of large support requires a large ω_i within which at least $|\omega_i|/2$ measurements fail. This suggests that local noise in the measurement process will yield local residual noise. Indeed, a standard argument [2, 11] shows that if the noise is local and below a threshold, so will be the residual noise, see appendix B.

3.5. Fault tolerance

In general the original 3D state will be noisy, and so will be the measurements and the application of $E \in \mathcal{G}_3|_2$. Local errors affecting outer qubits at any time will remain local, because the application of E is local. Local errors affecting inner qubits and previous to measurements can be absorbed as local measurement errors. If all the noise is local and below a threshold, so will be the residual noise after the 'dimensional collapse'.

3.6. Blowing up

The inverse dimensional jump only requires initializing the inner code. Since it encodes no logical qubits, it suffices to apply error correction to an arbitrary state of the inner qubits. Moreover, since it is a 3D gauge color code with local stabilizer and gauge generators it admits single-shot error correction, see appendix A, and thus the process is quantum-local.

3.7. Higher dimensions

Colexes and gauge color codes can be defined for arbitrary dimensions [10]. For a given D-colex it is possible to build different color codes with labels (d, e) that indicate the dimension of the gauge generators: d and e are positive integers with $d + e \leqslant D$, X-type generators are $(e + 1)$-cell operators, and Z-type generators are $(d + 1)$-cell operators.

The most interesting class of color codes is that constructed out of *simplicial* colexes, which generalize the triangular and tetrahedral colexes considered above. They encode a single qubit, and the logical X and Z operators can be chosen to have as support the whole colex. For these codes gauge fixing can be used to change, within a given colex, the parameters (d, e) at will [10].

The dimensional jump described above switches between a given $(1,\ 1)$ tetrahedral color code and a $(1,\ 1)$ triangular color code defined on any of the facets of the tetrahedron. Analogously, one can switch between a (d, e) D-simplicial color code and a (d, e) $(D - 1)$-simplicial color code defined on any of the facets of the D-simplex[3]. The lesson is that gauge color codes with different values of D or (d, e) are more than just separate codes. Altogether they form a *system* of topological stabilizer codes, and much more is possible by making them work together than by using them separately.

Finally, for every dimension D there exists a minimal simplicial colex with $2^{D+1} - 1$ vertices [10]. The corresponding color codes are quantum Reed–Muller codes and are known to be related via gauge fixing [25].

4. 3D-local computation

Dimensional jumps open the door to a 3D-local fault-tolerant quantum computer in which all operations are quantum local. This sections describes a particular approach to achieve this this. The time overhead is constant on the number of logical gates, and only a single 3D color code lattice is required. As a drawback, any parallelism of the original circuit is lost, so no higher-level fault-tolerance is possible.

5. Layout

The general layout is described in figure 1: a stack of 2D color codes with a 3D color code at one end of the stack. Computations are performed at the end where the 3D code lives, whereas the stack acts as a memory. For the logical qubits encoded on the 2D color codes next to the three-colex structure, quantum-local initialization and

[3] Indeed, the gauge generators trivially match, and logical operators can be chosen to have as support the whole facet. To check that such operators commute with any cell operator of dimension at least two, observe that there exists only one color κ such that κ-edges might have a single vertex in the facet, and the cell operator contains edges of more than one color.

universal gates are available via dimensional jumps (quantum-local measurements do not require them [2]). Moreover, for those 2D codes single-shot error correction is available via dimensional jumps, making fault-tolerant CNot gates quantum-local. The only operation required in the stack is the (trivially transversal) swapping of the 2D codes. This allows to move logical qubits so that they are available at the computing end of the stack when needed. Single-shot error correction is not available in the stack, and it is not needed. For 2D codes in the stack error correction amounts to keeping track of errors by repeatedly measuring the stabilizer generators [2].

Only one obstacle is left: ensuring that the required logical qubits are always available at the computing end for each step of the computation, without incurring in waiting times. A procedure to achieve this is given next.

5.1. Computing at the end of a stack

Consider, as described above, a quantum computer where logical qubits are placed on a stack and non-trivial computations are limited to the end of the stack: the rest of qubits can only be swapped with their neighbors in the stack. It is possibly not entirely obvious that the number of rounds of parallel gates can be, up to a constant, equal to the number of gates in the circuit model. This section provides a simple algorithm for swapping the stacked logical qubits that achieves this.

At the end of the stack there are a number of (logical) qubits on which non-swap gates can be performed. At least there should be two of them, since two-qubit gates are necessary. These qubits can be regarded as the 'internal' memory of a 'processing unit', while the rest form an 'external' memory: the stack. On the stack two kinds of operations are allowed: the swap of neighboring qubits and the swap of the topmost qubit with an internal qubit.

The computation is divided in steps. At each step a certain external qubit is required to be at the end of the stack, so that it can be accessed by the processing unit. The challenge is to perform on the stack, after each computational step, a finite depth circuit composed of nearest neighbor swaps that places the next qubit to be processed at the end of the stack.

The proposed algorithm is the following. The positions in the stack are labeled with integers: position 1 is the topmost. At the sth step the position i has an integer label m_i: it indicates that the qubit currently at position i should be at position 1 on the step $m_i \geqslant s$, but not before. In particular at the sth step

$$m_1 = s. \tag{12}$$

When a qubit will not be used again this label can take any value larger than the number of steps, with the condition that $m_i \neq m_j$ for $i \neq j$.

At start, before the first step, qubits are ordered according to their first use, i.e.

$$m_i < m_{i+1}, \qquad m_1 = 1. \tag{13}$$

After the step s is performed the label m_1 is updated to its new value (the rest stay clearly the same) and the following operations are performed, each consisting of a round of parallel swaps:

(1) For every $n \geqslant 0$, if

$$m_{2n+1} > m_{2n+2} \tag{14}$$

swap the qubits at positions $2n + 1$ and $2n + 2$ (and the labels m_{2n+1} and m_{2n+2} accordingly).

(2) For every $n \geqslant 1$, if

$$m_{2n} > m_{2n+1} \tag{15}$$

swap the qubits at positions $2n$ and $2n + 1$ (and the labels m_{2n} and m_{2n+1} accordingly).

To check that (12) is satisfied at every step, it suffices to show that at every step and for all positions i

$$m_1 \leqslant m_i. \tag{16}$$

Suppose that after the sth step and the update of m_1

$$m_{2n} < m_{2n+k}, \qquad n,\, k > 0. \tag{17}$$

Then, after the first round of swaps is performed

$$m_{2n+1} < m_{2n+1+k}, \qquad n \geqslant 0,\, k > 0 \tag{18}$$

and after the second round of swaps is performed both (16) and (17) hold again. In particular $m_1 = s + 1$ as required. Thus the inequalities (17) are an invariant of the two-round procedure. Since they are initially satisfied, the algorithm works.

Acknowledgments

I thank the MINECO grant FIS2012-33152, the CAM grant QUITEMAD+, the Sapere Aude grant of the Danish Council for Independent Research, the ERC Starting Grant QMULT and the CHIST-ERA project CQC. This work was supported by the International Research Unit of Advanced Future Studies at Kyoto University.

Appendix A. Boundaries in 3D color codes

The main text makes use of some basic properties of gauge color codes that were presented in [11]. For easier reference they are gathered in this appendix.

A.1. Geometry

The tetrahedral colexes of the main text are just an example of a larger class of geometries for color codes. As in [11], of interest here are three-colexes Λ that are topological balls and are obtained by removing some vertices (and all cells in contact with them) of a larger colex $\bar{\Lambda}$ without boundary. The boundary of Λ is a topological sphere divided in *regions* (discs), that meet at *borders* (open lines), that meet at *corners* (points). They are defined as follows. Consider a *rgb*-cell of $\bar{\Lambda}$ that is not part of Λ but is in contact with Λ. The *rgb*-cell and Λ share a set of plaquettes R, which can only contain *rg*-, *rb*- or *gb*-plaquettes. Typically these plaquettes form a topological disc in the boundary of Λ, which is called a *rgb*-region (otherwise there are several discs, each a region). The *r*- and *g*-edges that separate a *rgb*-region and a *rgy*-region are said to form a *rg*-border. For each *rg*-border there is exactly one *rg*-plaquette in $\bar{\Lambda}$ (and not in Λ) that contains its edges. Finally, when a *rg*-border, a *rb*-border and a *gb*-border meet at a vertex, this is said to be a *y*-corner of the colex. Again, for each *y*-corner there is exactly one *y*-edge in $\bar{\Lambda}$ (and not in Λ) that contains it. This description of the boundary of Λ in terms of regions, borders and corners does not depend on $\bar{\Lambda}$: it is intrinsic to Λ.

In the case of the tetrahedral colex each facet is a region with a different color combination: rgb, rgy, rby and gby. The simplest $\bar{\Lambda}$ is obtained by adding a single additional vertex together with a single edge, plaquette and cell for each color combination. The resulting manifold is a three-sphere. Due to this construction, color codes in tetrahedral colexes have been called 'punctured'.

The inner colex described in the main text is another example of the above general class of geometries. It is obtained from a tetrahedral colex by erasing all the cells in contact with one of the facets/regions. Each of the removed cells contributes a region in the inner colex. E.g. in the main text the outer set of vertices is the *rgb*-facet, which is in contact with *rgy*-, *rby*- and *gby*-cells. To obtain the inner colex all the outer vertices are removed, together with those cells. The remaining inner colex has *rgy*-, *rby*- and *gby*-regions, which are of two kinds. They can correspond to one of the original facets of the tetrahedron, or they can correspond to one of the erased cells.

The regions of the tetrahedral and inner colex turn out to have very different properties at the level of the code. This motivates some definitions. A border is *odd* if it connects two corners with different colors. A region without odd borders is said to be *frozen*. A region with an odd number of odd borders of any given color is said to be *free*. In a tetrahedral colex all regions are free, e.g. the *rgb*-facet has a single odd *rg*-border, connecting the *r*- and the *b*-corner. In an inner colex all regions are frozen, because all corners have the same color (in the above example they are *y*-corners).

A.2. Stabilizer and logical operators

As stated in the main text, a 3D gauge color code is obtained by placing a qubit at each vertex of the three-colex and attaching gauge X and Z generators to plaquettes. A bit-flip plaquette operator X_{p} has support on the qubits belonging to the plaquette p, and similarly for a phase-flip plaquette operator Z_{p}. If p is a *rg*-plaquette then X_{p} can only anticommute with $Z_{\mathrm{p}'}$ if p$'$ is a *by*-plaquette: e.g. if p' is a *gb*-plaquette then it shares with p a certain number x of *b*-edges, and then exactly $2x$ qubits (a qubit belongs at most to one *b*-edge).

For the above class of geometries, it is shown in [11] that the generators of $\mathcal{Z}(\mathcal{G})$ are (i) cell operators and (ii) region operators, i.e. operators of the form

$$X_R \coloneqq \prod_{i\in R} X_i, \qquad Z_R \coloneqq \prod_{i\in R} Z_i, \tag{A1}$$

where X_i, Z_i are the Pauli X, Z, operators on the ith qubit and R is a region, regarded here as set of vertices/qubits. Moreover, (i) a region R is free if and only if X_R and Z_R anticommute, and (ii) given two different regions R and R', X_R and $Z_{R'}$ anticommute if and only if they share an odd number of odd borders. From this result it follows that to obtain a representative group $\mathcal{L}$ of bare logical operators

$$\mathcal{L} \subseteq \mathcal{Z}(\mathcal{G}), \qquad \mathcal{L} \simeq \mathcal{Z}(\mathcal{G})/\mathcal{S}, \tag{A2}$$

it suffices to choose a minimal generating set for $\mathcal{L}$ among X and Z region operators. It follows also that for frozen regions R the operators X_R and Z_R are stabilizer elements. In particular, if R is a frozen *rgy*-region with *y*-corners X_R can be obtained as a product of plaquette operators X_p as follows (and similarly for Z operators)

$$X_R = \prod_{\text{p}\in rg-\text{plaquettes } in\ R} X_p. \tag{A3}$$

Indeed, every vertex/qubit belongs to exactly one *rg*-plaquette of $\bar{\Lambda}$, and thus either (i) it belongs to exactly one *rg*-plaquette of Λ or (ii) it is part of an *rg*-border. The second options is never true for a frozen *rgy*-region with *y*-corners: it only shares vertices with *ry*-, *gy*- and *by*-borders. Moreover, R only borders with *rby*- and *gby*-regions, which cannot contain *rg*-plaquettes: if a vertex belongs to R, it belongs to a unique *rg*-plaquette contained in R.

When all the regions are free region operators cannot contribute any stabilizer generators: a product of X and Z region operators can only belong to the stabilizer if it is trivial. When all the regions are frozen, on the other hand, all region operators belong to the stabilizer. Region operators are, however, not independent. In a colex where all corners are *y*-corners the following identity holds (and similarly for Z operators)

$$\prod_{c\in rgb-\text{cells}} X_c \prod_{c\in rgy-\text{cells}} X_c = \prod_{R\in rgy-\text{regions}} X_c, \tag{A4}$$

where X_c are bit-flip cell operators and X_p are bit-flip plaquette operators. This is because each vertex/qubit belongs (i) exactly to one *rgb*-cell and (ii) exactly to one *rgy*-cell, *unless* it belongs to a *rgy*-region. Ulternatively, this lack of independence of region operators is a trivial consequence of the flux picture of the main text and the expression (A3): a flux incoming at a certain *rg*-plaquette of a given *rgy*-region has to exit at another *rg*-plaquette of another (or the same) *rgy*-region.

In tetrahedral codes all the regions are free and thus the stabilizer $\mathcal{S}$ is generated by cell operators alone. In inner codes all the regions are frozen and thus the stabilizer has as generators (i) cell operators and (ii) region operators, except those coming from the facets of the original tetrahedral colex (facet operators are redundant due to (A4)). The take home message is that for all the geometries considered in the main text there exist a set local generators of the stabilizer.

A.3. Single-shot error correction

Gauge color codes with local stabilizer and gauge generators exhibit single-shot error correction [11]. In particular, this is true for the above geometries when (i) all the regions are free or (ii) all the regions are frozen. The quantum-local process for fault-tolerant error correction can be split in two analogous processes that correct separately bit-flip and phase-flip errors. Let $\mathcal{G}_Z$, $\mathcal{S}_Z$ denote the gauge and stabilizer subgroups generated by Z operators. Single-shot error correction of bit-flip errors amounts to

(1) measure Z plaquette operators (generators of $\mathcal{G}_Z$),

(2) process the measurements (classically) to obtain a syndrome σ for $\mathcal{S}_Z$ (a subgroup of $\mathcal{G}_Z$),

(3) choose a bit-flip operator E with syndrome σ, and

(4) apply E.

In the gauge color code families considered in the main text both the stabilizer and gauge group have local generators. Single-shot error correction is both possible for tetrahedral codes (because all regions are free) and for inner codes (because all regions are frozen).

Appendix B. Single-shot gauge fixing

The purpose of this appendix is to show that the dimensional collapse procedure of the main text is fault-tolerant. The derivation will be entirely parallel to the one used in [11] to show that single-shot error correction is feasible with 3D gauge color codes. We will need in particular the following definition and lemma from [11].

Definition 1. Let $p(A)$ be a probability distribution over subsets $A \subseteq B$ of some set B. Given some $\alpha > 0$ the distribution p is α-bounded if for every $A \subseteq B$

$$\tilde{p}(A) \coloneqq \sum_{A'\supseteq A} p(A') \leqslant \alpha^{|A|}. \tag{B1}$$

Lemma 2. *Consider a family of graphs with bounded maximum degree and some $k > 0$. Each graph has a set of nodes Γ and comes equipped with (i) two subsets of nodes $\Gamma_i \subseteq \Gamma$, $i = 1, 2$, and (ii) a function $f\colon \Gamma_1 \longrightarrow \Gamma$ such* *that for every cluster (set of nodes)* $V \subseteq \Gamma_1$ *and for every connected component* V_c *of the cluster* $f(V)$

$$|V \bigcap V_c| \geqslant k|V_c|. \tag{B2}$$

There exists some $\alpha_0 > 0$ *such that for every* α *with* $0 < \alpha < \alpha_0$ *the following holds. If a probability distributions* $p_1(V)$ *over clusters* $V \subseteq \Gamma_1$ *is* α*-bounded, then the probability distribution* $p_2(V')$ *over clusters* $V' \subseteq \Gamma_2$ *defined by*

$$p_2(V') := \sum_{V \,|\, f(V) \bigcap \Gamma_2 = V'} p_1(V). \tag{B3}$$

is β*-bounded with*

$$\beta := \frac{(\alpha/\alpha_0)^k}{1-(\alpha/\alpha_0)^k}, \tag{B4}$$

As argued in the main text, it is enough to show that local measurement noise in the syndrome extraction step gives rise to local residual noise after the gauge fixing step. We will do this for some simple but compelling enough model of noise. The starting point is some family of 3D tetrahedral gauge color codes that is regular enough, i.e. the lattice is translationally invariant up to boundaries and locally identical across the family, and the shape of the tetrahedron just scales in size across the family, without other changes. As in the text, we will focus on the gauge fixing of *rg*-plaquette *Z* operators in $\mathcal{S}_2$. Other color combinations and *X* operators are analogous.

Let $\mathcal{E}$ be the set of edges dual to inner *rg*-plaquettes. Recall that each edge $e \in \mathcal{E}$ has two endpoints, each of which can be (i) in an inner cell, in which case it is called an *inner endpoint*, (ii) in an interface *rgy*-cell, in which case it is regarded as an endpoint at the corresponding outer *rg*-plaquette, and called *outer endpoint*, and (iii) at the *rgy*-facet (this case is uninteresting). The endpoints of some $\gamma \subseteq \mathcal{E}$ are those inner cells or outer plaquettes that are endpoints of an odd number of edges $e \in \gamma$. Let $\mathcal{E}_0 \subseteq \mathcal{E}$ be the subset of elements without inner endpoints. Given $\gamma \in \mathcal{E}_0$ let $\partial\gamma$ denote the set of outer *rg*-plaquettes where γ has an endpoint.

Let $\mathcal{B} \subset \mathcal{G}_3|_2$ be the set of bit-flip operators that are products of *b*-string operators (in the outer 2D color code). A given $E \in \mathcal{B}$ can only anticommute with an outer plaquette operator $Z_{\rm p}$ if p is an *rg*-plaquette. Let Synd E denote the set of such *rg*-plaquettes, so that Synd E characterizes the syndrome of E in $\mathcal{S}_2$. The geometry of a 2D triangular color code is such that Synd E can take any value, it is unconstrained. In particular, for every $\gamma \in \mathcal{E}_0$ there exists some $E_\gamma \in \mathcal{B}$ with

$$\mathrm{Synd}\ E_\gamma = \partial\gamma. \tag{B5}$$

The operator E_γ is unique up to stabilizers, and we choose it so that it has minimal support, i.e. for every $E \in \mathcal{B}$

$$\mathrm{Synd}\ E = \partial\gamma \quad \Longrightarrow \quad |\mathrm{Supp}\ E_\gamma| \leqslant |\mathrm{Supp}\ E|. \tag{B6}$$

As discussed in the text, it follows by inspection of the geometry of the problem that there exists some constant $K > 0$ such for every set of dual inner edges γ without inner endpoints

$$|\mathrm{Supp}\ E_\gamma| \leqslant K|\gamma|. \tag{B7}$$

In particular, K is constant across the family of codes.

We model local noise in the measurements through a probability distribution $p(\delta)$ over sets $\delta \subseteq \mathcal{E}$. Recall that plaquette operators from *rg*-plaquettes dual to the elements in δ give a wrong measurement outcome. To model the locality of the measurement noise, we impose that $p(\delta)$ is α-bounded for some $\alpha > 0$.

The residual error when the wrong set of measurement outcomes is given by $\delta \subseteq \mathcal{E}$ is

$$E_{\delta+\delta_0}, \tag{B8}$$

where $\delta_0 \subseteq \mathcal{E}$ has the same inner endpoints as δ and minimal cardinality among sets with that property, i.e. for any $\delta' \in \mathcal{E}$,

$$\partial\delta = \partial\delta' \quad \Longrightarrow \quad |\delta_0| \leqslant |\delta'|. \tag{B9}$$

Let $\mathcal{Q}$ be the set of physical qubits in the outer 2D code. To quantify the residual noise we construct a distribution $p'(\mathrm{Q})$ over sets of qubits $Q \subseteq \mathcal{Q}$, with $p'(\mathrm{Q})$ the probability that the support of $E_{\delta+\delta_0}$ is Q, i.e.

$$p'(\mathrm{Q}) := \sum_{\delta|\mathrm{Supp}\ E_{\delta+\delta_0}=\mathrm{Q}} p(\delta). \tag{B10}$$

The following result quantifies the locality of the residual noise. It shows in particular that it can be made as small as desired by improving the quality of the measurements.

Proposition 3. *Given a family of 3D gauge color codes as described above and satisfying in particular condition (B7) for some* $K > 0$*, there exists some* $\alpha_0 > 0$ *such that for every* α*-bounded distribution* $p(\delta)$ *the distribution* $p'(\mathrm{Q})$ *of (B10) is* β*-bounded with* β *as in (B4) and*

$$k = \frac{1}{2(1+K)}. \tag{B11}$$

Proof. Construct for each code in the family a graph with node set

$$\Gamma = \mathcal{E} \sqcup \mathcal{Q} \tag{B12}$$

and such that two nodes a, b are linked if

- $a,\, b \in \mathcal{E}$ and a and b share an inner or outer endpoint,
- $a,\, b \in \mathcal{Q}$ and a and b are both in the same b-edge or in the same rg-plaquette, or
- $a \in \mathcal{Q},\, b \in \mathcal{E}$ and a is in a rg-plaquette that is an outer endpoint of b.

This definition is designed so that trivially (i) the family of graphs has bounded maximum degree and (ii) the following is satisfied: given a connected component $\gamma_c \sqcup Q_c$ of a cluster $\gamma \sqcup Q$, where $\gamma_c,\, \gamma \subseteq \mathcal{E}$ and $Q_c,\, Q \subseteq \mathcal{Q}$, and given $E \in \mathcal{B}$ with Supp $E = Q$,

$$\delta \in \mathcal{E}_0 \quad \Longrightarrow \quad \delta_c \in \mathcal{E}_0, \tag{B13}$$

$$\text{Synd } E = \partial\delta \quad \Longrightarrow \quad \text{Synd } E_c = \partial\delta_c, \tag{B14}$$

where E_c is the restriction of E to Q_c.

The result will follow by applying lemma 2 to the above family of graphs, taking $\Gamma_1 = \mathcal{E}, \Gamma_2 = \mathcal{Q}$ and

$$f(\delta) = (\delta + \delta_0) \bigcup \text{Supp } E_{\delta+\delta_0}. \tag{B15}$$

In this case (B2) reads

$$2(1+K)|\delta \bigcap \delta_c| \geqslant |\delta_c| + |Q_c|, \tag{B16}$$

where $\delta_c \sqcup Q_c$ is any connected component of $f(\delta)$. Notice that

$$|\delta \bigcap \delta_c| = |\delta_c| - |\delta_0 \bigcap \delta_c|. \tag{B17}$$

According to (B13) $\delta_c \in \mathcal{E}_0$ and thus

$$\delta + \delta_0' \in \mathcal{E}_0, \qquad \delta_0' := \delta_0 + \delta_c. \tag{B18}$$

By the minimality of δ_0 (B9)

$$0 \leqslant |\delta_0'| - |\delta_0| = |\delta_c| - 2|\delta_c \bigcap \delta_0| = 2|\delta \bigcap \delta_c| - |\delta_c|. \tag{B19}$$

Thus

$$2|\delta \bigcap \delta_c| \geqslant |\delta_c|. \tag{B20}$$

If E_c is the restriction of $E_{\delta+\delta_0}$ to Q_c, according to (B14) Synd $E_c = \partial\delta_c$. Then

$$\text{Synd } E' = \text{Synd } E_{\delta+\delta_0}, \qquad E' := E_{\delta+\delta_0} E_c E_{\delta_c} \tag{B21}$$

and

$$\begin{aligned} &|\text{Supp } (E_{\delta+\delta_0} E_c)| + |\text{Supp } E'_{\delta_c}| \geqslant |\text{Supp } E'| \\ &\geqslant |\text{Supp } E_{\delta+\delta_0}| = |\text{Supp } (E_{\delta+\delta_0} E_c)| + |Q_c|, \end{aligned} \tag{B22}$$

where the second inequality is by the minimality of $E_{\delta+\delta_0}$ (B6). Using (B7)

$$K|\delta_c| \geqslant |\text{Supp } E'_{\delta_c}| \geqslant |Q_c|. \tag{B23}$$

The inequalities (B20) and (B23) imply (B16).

References

[1] Lidar D and Brun T (ed) 2013 *Quantum Error Correction* (New York: Cambridge University Press)
[2] Dennis E, Kitaev A, Landahl A and Preskill J 2002 *J. Math. Phys.* **43** 4452
[3] Kitaev A 2003 *Ann. Phys.* **303** 2
[4] Bravyi S and König R 2013 *Phys. Rev. Lett.* **110** 170503
[5] Bombin H and Martin-Delgado M A 2007a *Phys. Rev. Lett.* **98** 160502
[6] Bombin H and Martin-Delgado M A 2006 *Phys. Rev. Lett.* **97** 180501
[7] Nigg D *et al* 2014 *Science* **345** 302
[8] Pastawski F and Yoshida B 2015 *Phys. Rev.* A **91** 012305
[9] Bravyi S and Kitaev A 2005 *Phys. Rev.* A **71** 22316
[10] Bombin H 2015 *New J. Phys.* **17** 083002
[11] Bombin H 2015 *Phys. Rev.* X **5** 031043
[12] Brown B J, Nickerson N H and Browne D E 2015 arXiv:1503.08217

[13] Poulin D 2005 *Phys. Rev. Lett.* **95** 230504

[14] Paetznick A and Reichardt B W 2013 *Phys. Rev. Lett.* **111** 90505

[15] Bombin H and Martin-Delgado M A 2007 *Phys. Rev.* B **75** 75103

[16] Kubica A and Beverland M E 2015 *Phys. Rev.* A **91** 032330

[17] Katzgraber H, Bombin H and Martin-Delgado M 2009 *Phys. Rev. Lett.* **103** 90501

[18] Katzgraber H, Bombin H, Andrist R and Martin-Delgado M 2010 *Phys. Rev.* A **81** 12319

[19] Andrist R, Katzgraber H, Bombin H and Martin-Delgado M 2010 *New J. Phys.* **13** 083006

[20] Landahl A J, Anderson J T and Rice P R 2011 arXiv:1108.5738

[21] Bombin H, Andrist R, Ohzeki M, Katzgraber H and Martin-Delgado M 2012 *Phys. Rev.* X **2** 21004

[22] Sarvepalli P and Raussendorf R 2012 *Phys. Rev.* A **85** 022317

[23] Delfosse N 2014 *Phys. Rev.* A **89** 012317

[24] Stephens A M 2014 arXiv:1402.3037

[25] Anderson J T, Duclos-Cianci G and Poulin D 2014 *Phys. Rev. Lett.* **113** 080501