Studies of quantum error correction

Leonid P. Pryadko

UCR

(Dated: April 7, 2020)

Study questions on quantum codes

I. MEASUREMENT CIRCUITS FOR SIMPLE CODES

1. **Rotated surface codes**: Consider a family of cyclic n-qubit codes, with stabilizer generators of the form
$$g_i = X_i Z_{i+a} Z_{i+a+b} X_{i+2a+b}, \quad i \in \{0, 1, 2, \ldots, n-1\},$$
where the qubit indices are taken $\pmod n$ as appropriate for a cyclic code. For $n = 5$ and $a = b = 1$ this gives exactly the 5-qubit code.
   (a) Show that all of these commute with each other for any a and b (we got a valid stabilizer group).
   (b) Find the number of *independent* generators for different a, b, and $n \leq 30$ such that $2a + b < n$, and calculate the corresponding numbers of encoded qubits k.
   (c) Try to find layout of these operators on a square lattice with periodic boundary conditions [think about lattice periodicity vectors $u = (u_x, u_y)$ and $v = (v_x, v_y)$ such that, e.g., the pair of points $r = (x, y)$ and $r + u$ are identified – this is similar to wrapping vectors for carbon nanotubes]. For which values of n, a, b can you get such a square lattice layout with stabilizer generators forming a simple repeating pattern, e.g., around a single plaquette? (With qubits on vertices of the square lattice.)
   (d) Using a representation of the Pauli operators in terms of a pair of binary vectors, write a little program for calculating a distance of such a code corresponding to a stabilizer generator matrix $G = (A|B)$. For example, for the 5-qubit code, this matrix has the form (only non-zero elements are shown):
$$G = \left(\begin{array}{ccccc|ccccc} 1 & . & . & 1 & . & . & 1 & 1 & . & . \\ . & 1 & . & . & 1 & . & . & 1 & 1 & . \\ 1 & . & 1 & . & . & . & . & . & 1 & 1 \\ . & 1 & . & 1 & . & 1 & . & . & . & 1 \\ . & . & 1 & . & 1 & 1 & 1 & . & . & . \end{array}\right).$$
   For the simplest (but not the most efficient program), given G, you can just go over all 4^n Pauli operators, calculating the corresponding syndromes, and finding the non-trivial undetectable operators of minimum weight to calculate the code distance d.
   This can be done, e.g., using Mathematica, Gap, C, or C++. For example, in Mathematica, given the matrix `G` above, the number of independent generators can be counted simply as `MatrixRank[G,Modulus->2]` — this is for the item (b).
   (e) Whether with such a program or by hand, try to find a pattern in k and d as a function of n, a, and b, and try to generalize your findings to construct interesting

families of such codes (square lattice layout would help here). In particular, I remember that one such sequence with $n = 5, 13, 25, 41, \ldots$ exists.

2. For a given set of n, a, b, assuming you also have n ancillary qubits, construct a circuit for simultaneous measurement of all n stabilizer generators. Assume that two-qubit `CNOT` gates between any pair of qubits are allowed, and simultaneous non-overlapping gates are also allowed. You can also use single-qubit Clifford gates in your circuit, the `H` (Hadamard) and `P` (Phase) gates.
   (a) Start with constructing a circuit for measuring a single generator of the 5 qubit code.
   (b) Come up with the schedule to measure all 5 generators simultaneously (including the redundant generator). Write the circuit as a QASM program, e.g. (there are variants of QASM, you may need to adjust this to work):

```
# declare qubits
qubits 1,2,3,4,5,6,7,8,9,10
# t=0 (comment to mark the time step)
Prep_z 6, 7, 8, 9, 10
# t=1
CNOT 1 6
CNOT 2 7
CNOT 3 8
CNOT 4 9
CNOT 5 10
# t=2
...
```

   (c) Using my shell scripts as examples, write a little program to output a sequence of gates in QASM format for measuring generators of a code with given a, b.
   (d) Write two versions of the program: one measuring $X_1Z_2Z_3X_4$ operators in linear order 1-2-3-4, and the other in order 1-3-2-4.
   (e) Also write variants of your programs for the generators in the form $X_1Y_2Y_3X_4$, with Z replaced by Y — this should be more efficient when phase noise is dominant.
   (f) Typeset a few of your programs as circuits (e.g., using `qasm2circ` program—google for it).
3. **Surface codes with smooth and rough boundaries:** Consider a family of square lattice surface codes with smooth and rough boundaries, see Fig. 1.
   (a) What are the logical operators in this code?
   (b) What is the total number of data qubits for an $a \times b$ lattice?
   (c) What is the X and Z distance for such a code on an $a \times b$ lattice? What are the smallest values of a and b for a distance-3 code?
   (d) Compare the parameters of these codes with those for rotated surface codes in item 1.
   (e) Assuming there is an ancillary qubit for each generator (in each vertex and in the middle of each square), for the smallest distance-3 code in this class, come up with a circuit of depth 4 CNOT gates (plus some Hadamard gates) to measure the stabilizer generators.
   (f) Using the shell scripts I sent you as examples, come up with a general script to generate measurement circuits for arbitrary values of a and b.

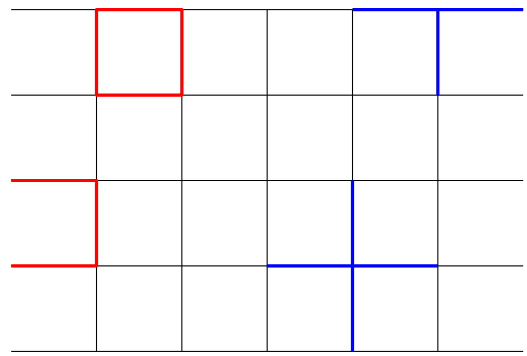

FIG. 1. Square lattice surface code of size 5×6 with smooth boundaries on top/bottom and rough boundaries on left/right. Qubits are located on the edges. Stabilizer generators shown as red (Z-type) and blue (X-type) shapes have 3 qubits on the boundaries and 4 qubits in the bulk.

4. **Gottesman-Knill theorem**: Learn how error propagation in Clifford circuits works.
   (a) Propagate all three distinct single-qubit errors in the two circuits for the 5 qubit code, by hand, to the output of the circuit, so that you can see what syndrome is measured and what is the resulting error.
   (b) Propagate all possible single-qubit errors through the circuit and bin them by syndrome values and error equivalence classes. This gives the error model to linear order in error probability p. Compare the results for the two circuit versions in item 2(d).
5. For small codes, given the stabilizer generator matrix as in 1(d) and a similar matrix for the logical generators, construct a (non-efficient) maximum-likelihood (ML) decoder which computes explicitly the partition functions corresponding to a given error e (in binary format) and the errors $e + c$, where c if one of the $4^k - 1$ non-trivial codewords corresponding to logical operators of the code. Here we will focus on small codes with $k = 1$ and $k = 2$. See Dennis et al[1] and also my more recent paper Ref. 2 for discussion of ML decoding.
6. Use the results above to analyze the structure of errors to leading order in small error probability p.
7. Use errors generated by the circuit and the ML decoder constructed above, based on the qubit error probabilities $p_X = p_Y$, $p_X + p_Y + p_Z = p$ (with the addition of syndrome measurement error q) to simulate error correction in your circuits.
8. **Subsystem codes.** Refs. 3 and 4. These are characterized by length n, number of encoded qubits k, number of gauge qubits κ, and distance d.
9. Start with a formally defined gauge generator matrix $G = (A|B)$, where A and B are binary matrices of identical size. Denote $C = AB^T + BA^T$. For a stabilizer code, $C = 0$ (all generators commute), while in a subsystem code it is not necessarily so. Given the size and binary ranks of the matrices G and C, calculate the parameters n, k, κ?
10. Given the gauge group $\mathcal{G}$ of a subsystem code, its stabilizer group $\mathcal{S}$ is defined as the center of $\mathcal{G}$ (up to a phase), that is, the maximal subgroup of $\mathcal{G}$ with phases removed, whose elements commute with all elements of $\mathcal{G}$. Come up with prescription to construct the stabilizer generator matrix given matrices G and C.

[1] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, "Topological quantum memory," J. Math. Phys. **43**, 4452 (2002).

[2] L. P. Pryadko, "On maximum-likelihood decoding with circuit-level errors," (2019), unpublished, arXiv:1909.06732.

[3] David Poulin, "Stabilizer formalism for operator quantum error correction," Phys. Rev. Lett. **95**, 230504 (2005).

[4] Dave Bacon, "Operator quantum error-correcting subsystems for self-correcting quantum memories," Phys. Rev. A **73**, 012340 (2006).