My talk - topological phases for robust quantum memories

Friday, July 20, 2018 2:43 PM

The following is crude note that guided my talks. For each talk there is Plan and Actual, where the latter contains keywords we have covered.

Here is some reference information:

- Homology:
  Kitaev's original paper mentions this, assuming familiarity with cellular homology. A thorough reference for this subject is Hatcher's book "Algebraic Topology" which you can download from http://pi.math.cornell.edu/~hatcher/AT/ATpage.html . This is a long book; in the lecture, we tried to develop intuition based on refinement of triangulation.
- Equivalent conditions for error correction:
  This is mainly based on https://arxiv.org/abs/1610.06169 . The paper treats approximate error correction from which one can read off the conditions stated in the lectures. Note that Knill-Laflamme condition is weaker than the others since it pulls a possibly huge Hilbert dimension factor. The "erasure method" to check the Knill-Laflamme condition is used in https://arxiv.org/abs/1101.1962
- 4D toric code, the (2,2)-toric code, is defined in https://arxiv.org/abs/quant-ph/0110143. The notation of (p,q)-toric code has appeared in https://arxiv.org/abs/1309.2680 .
- The existence of logical operators in hyperplanes of codimension 1, under certain situations, is proved in https://arxiv.org/abs/0810.1983 and https://arxiv.org/abs/1011.3529 . In the lectures, we discussed it as a consequence of the anticipation that the bulk is correctable, but the existence of codimension-1 logical operator can be shown under much milder conditions.
- The self-correcting property of the 4D toric code is proved in https://arxiv.org/abs/0811.0033 . More readable reference is https://arxiv.org/abs/0907.2807 .
- Polynomial methods for translation-invariant Pauli stabilizer code Hamiltonians is introduced in https://arxiv.org/abs/1204.1063 . A lecture note based on this is https://arxiv.org/abs/1607.01387 . The papers https://arxiv.org/abs/1505.02576 and https://arxiv.org/abs/1603.04442 also have appendices on polynomial representations.
- The proof of the no-strings rule for the cubic code is found in Sec. 5.3 of my thesis https://arxiv.org/abs/1305.6973. My thesis has an appendix on elements of commutative algebra that are used routinely in the polynomial method. The content is a subset of a terse yet excellent book "Introduction to Commutative Algebra" by Atiyah and MacDonald.

Lecture 1:

I've heard that we have learned basics of fault-tolerance, and one of the key constructions, namely the stabilizer formalism. So, I assume that you are familiar with it. For that matter, here are two exercises:

- Write down the projector on to the code space of a Pauli stabilizer code with careful attention to the normalization factor. Take the trace to prove that "every independent stabilizer halves the Hilbert space."
- Write down a stabilizer state's density matrice (a projector) as a stabilizers. Take partial trace, and show that the entanglement spectrum is flat, and compute the entanglement entropy.

It will not make sense if you heard an introductory lecture on topological phases, but you didn't hear of toric code. Let us do it here.

- Toric code definition: lattice, Hamiltonian. Logical qubit counting by stabilizers. Heuristic.
- Mention two string operators that wraps the torus, and look at the irrep.
- Now it is time to talk about homology:
    - Write the stabilizer map and syndrome as binary matrices. Symplectic form.
    - Convince that the logical operators are in one-one correspondence with the homology (algebraic)
    - Translate this into the toric code. Look at the plaquettes and e-particles to derive the chain maps.
    - Explain topological homology. Simplicial homology. Give intuition that homology is topological invariant.
- With this machinery we can define 3D toric code. Historically much well known as discrete gauge theory.
    - Surface operators, and string operators. First Betti number = Second Betti number = the number of logical qubits.
- Although the code distance is of the linear system size, the error correction is applicable for much larger set of qubits.
    - Let us check the Knill-Laflamme condition, by "erasing method."

Having confirmed the Knill-Laflamme condition, it is time for us to investigate some other equivalent conditions. Some of which will be useful to understand limits of lattice quantum codes.

- KL condition implies decoupling in the density matrix. (Schmidt decompose the AR state and use KL condition.)
- Decoupling applied to general state implies the decoupling with fixed state in the correctable region. (KL already implies that rho^A is fixed. As an exercise, prove it directly from the decoupling.)
- The stronger decoupling condition implies the recovery map. (Exercise.)
- Recovery map implies cleaning criterion. (Consider the dual channel on operators.)
- Cleaning condition implies KL condition. (Look at the <psi_i | M_A | psi_i > by fixing a reference state and a logical operator. And then using the linearity to remove the off-diagonal term.)

Actual Lecture 1:

Started with two exercises.
Due to silence from the audience for the nontriviality of the problem, I solved the problem on the board.
Toric code revisited. Counted logical operators.
Homology, simplicial homology.
Boundary maps
Homology group with Z2 coefficients.
Intuitive explanation of the invariance of the homology group to the triangulation by resorting to refinements of triangulation.
Made connection to homology by interpreting stabilizer map and excitation map. Was asked to repeat, and repeated.
Ended with brief remarks on higher dimensions. Promised to go to higher dimensions.

Lecture 2 plan

- With this machinery we can define 3D toric code. Historically much well known as discrete gauge theory.
  - Surface operators, and string operators. First Betti number = Second Betti number = the number of logical qubits.
- Although the code distance is of the linear system size, the error correction is applicable for much larger set of qubits.
  - Let us check the Knill-Laflamme condition, by "erasing method."

Having confirmed the Knill-Laflamme condition, it is time for us to investigate some other equivalent conditions. Some of which will be useful to understand limits of lattice quantum codes.

- KL condition implies decoupling in the density matrix. (Schmidt decompose the AR state and use KL condition.)
- Decoupling applied to general state implies the decoupling with fixed state in the correctable region. (KL already implies that rho^A is fixed. As an exercise, prove it directly from the decoupling.)
- The stronger decoupling condition implies the recovery map. (Exercise.)
- Recovery map implies cleaning criterion. (Consider the dual channel on operators.)
- Cleaning condition implies KL condition. (Look at the <psi_i | M_A | psi_i > by fixing a reference state and a logical operator. And then using the linearity to remove the off-diagonal term.)

- Request for robust quantum memory. Now that we have investigated examples and criteria of error correction, let us think of quantum memory more generally.
  - First we need some error correcting code realized as a ground state subspace. It doesn't have to be a ground state subspace, but let us restrict ourselves in that setting. Other possibilities are your opportunities.
  - Encoding of information: this is more or less your protocol to use a physical medium as a quantum memory.
  - Time evolution under noise. Thermal noise is of fundamental interest.
  - Read-off the memory. Decoding algorithm.
  - The combination of the whole is important. Do not underestimate the importance of the decoding

algorithm.

- Good and bad quantum memories.
  - 1D Ising model. Degenerate ground state, one bit of classical information. Why not quantum ?
  - Metropolis evolution; stochastic, detailed balance condition.
  - Gas of domain walls -- thermalize quickly, useless as a memory.
  - 2D Ising model - Peierl's argument. Energy vs entropy.
  - Quantum: 2D toric code vs 4D toric code (Emphasize the decoding algorithm, in this case cooling or simulated annealing).
    - Why do people then study 2D toric code, and some considers it as the best? It is used as software. There is no temperature or heat bath that is natural. The errors are random, and we actively error correct.
  - Energy barrier appears to be important.
  - Bravyi-Terhal no-go result on 2D. + Existence of logical operators on hyperplanes. Essentially 1D argument. Suppose every strip is correctable, then they lack any logical operator. By locality the union cannot contain any logical operator, and by the converse of the cleaning lemma, the union is correctable. By the cleaning lemma, the complement should contain all logical operators, but that cannot be the case by symmetry of the argument.

Lecture 2 Actual:
Reviewed Lecture 1.

Defined (1,2) (1,3), (2,2) toric code. Mentioned the correspondence of the logical operators to homology groups.

Mentioned that any stabilizer code has homological interpretation.

Listed Error correction criteria, didn't have time to prove it.
Left it as exercise.

Every logical operator resides in hyperplane.

Memory as combination of encoding/evolution/decoding.

Discussed 1D and 2D Ising model as classical memory, and pointed out that the energy barrier appears to be important.

Ended with a question what to do with 2D toric code.

Lecture 3 Plan: (Draw the fractal before I begin on the board.)

Review Lecture 2: constructed various toric codes. Listed Error correcting criteria with respect to erasure channel. Discussed memory in general, and differentiated 1D and 2D Ising models.

2D topologically ordered system, if the bulk is a good error correcting code, then all information is accessed on a union of two strings. The dynamics of 1D Ising model becomes relevant.

Ex) Hexagonal lattice, qubits on vertices, (X,Z)^{otimes 12} term on every hexagon with legs. Sagar Vijay brought up this example, to wonder about possibility of fractons in 2D. Find the elementary excitations and show that they are not fractons. If bold, find disentangling circuit to map it to the toric code. It is equivalent to 8 copies of the toric code, the number 12 is obtained by looking at the coker of excitation map.

Returning to the q memory problem, can we get rid of strings? If we can, great. If we cannot, then it is interesting to understand why.

What is a string, to begin thinking? In the continuum theory, no problem. A continuous map from an interval to the space is a string. But, then we might have to resort to "homology"-like construction, and the string might be inevitable. In a discrete lattice, there is no such thing as a continuous map from an interval, other than a constant map. Get back to propagating particles; the trace of such particle is a string. Two anchors to wrap excitations, and connecting bridge of bounded width. Compare to the "hexagonal insect model" (since an insect has six legs.)

Error correcting criteria does not rule out the possibility of no-string model. Let us develop some language, and simplify computation.

Polynomial method, and the cubic code.

- Classical coding theory. Label the position of 1's into coefficients. Polynomial representation of stabilizer code Hamiltonians.
- Quantum is "square" or "double" of classical.
- So, quantumly one gets a sequence of length 2.
- Present the polynomial, and lay out a diagram in the checkerboard cubic lattice. That would "feel" different. I would call it "dandelion" model as it looks like the seed with fluffy parachutes. Note the similarity to the checkerboard model.
- What are the logical operators? Construction of fractals in terms of a zero-divisor on the virtual excitation module. Mention linear cellular automata. Local construction of the logical operators, and energy barrier.
- Fusion rule of the topological charges. Coker eps.
- The no-strings rule as two-term zero divisor condition.

Decoding algorithm: Look at the spatial distribution of errors, cluster them mentally, test the neutrality and if neutral, annihilate the cluster. RG type, decoding algorithm.

Graph the memory time, which is upper bounded by relaxation time to thermal state, as a function of the system size.

Where are we headed?

- Still the q memory problem stable against thermal errors is open. Energy/Entropic barrier has been considered, but not too satisfactory to compete with (2,2)-toric code. Is the decoding algorithm bad, or is the cubic code fundamentally not self-correcting. Can it be better?
- Big question on "phase of matter". Prevailing definition of adiabatic path in the Hamiltonian space can be applied, but when it comes to sequences of finite systems, it appears to be too narrow in some sense. Would one call the dandelion model to represent different phase of matter for each system size?
    - A related matter is in the entanglement RG. No usual field theory would generate more field components as we go to longer distances. UV cutoff insensitive theory might not exist at all.

    - May be we have to equate a lot more, to make progress.
- Anything that can be asked about translation invariant stabilizer code can be phrased in terms of polynomial maps. Find invariants. Is the dandelion model different from cubic code? There are many cubic codes, are they equal or different?
- Subsystem symmetry enriched models.
- In this talk completely missing is the gauging or duality map. In some type I fracton models, the gauging appears to make sense. But in type II, the number of symmetry generators, up to translation, does not appear to match the "gauge" field degrees of freedom. What are we gauging?

Lecture 3 Actual:

Reviewed previous lecture on homology-based construction of toric codes
Clarified the notation (a,n-a)-toric code.
Defined string operators on lattices.
Explained the polynomial representation.
Expressed commutativity and TQO condition using complex
Suggested how to prove the no-strings rule using the polynomial method.
Ended with suggestive remarks.