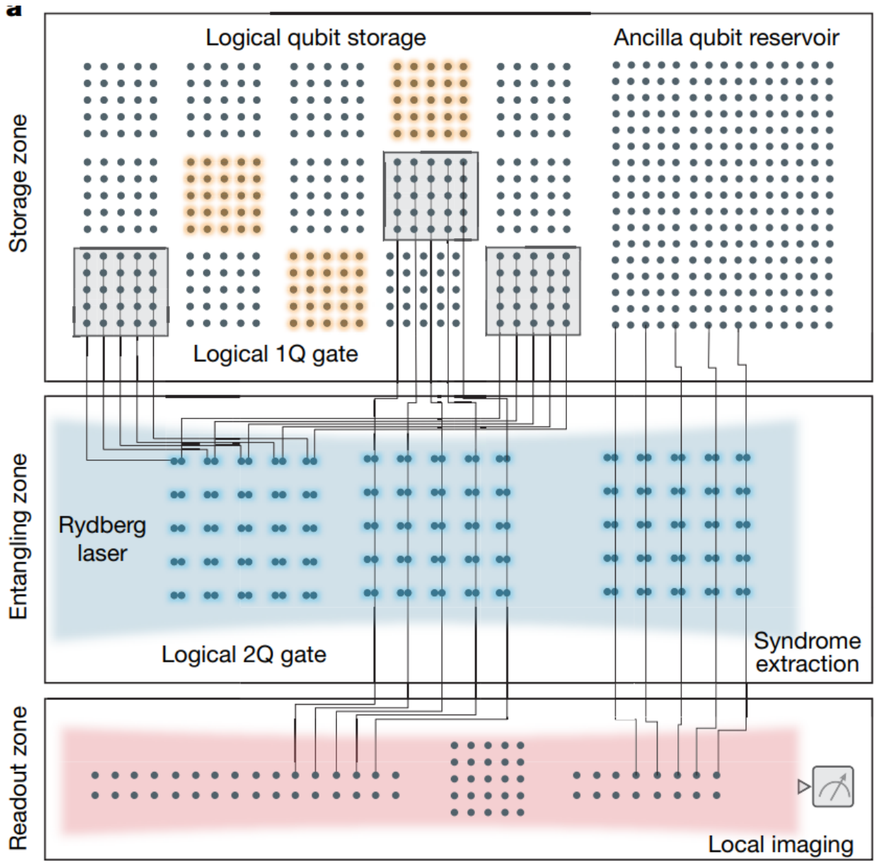
Quantum error correction gets real



Shor

Steane

INSTITUTE FOR QUANTUM INFORMATION AND MATTER

John Preskill
CIQC Annual Meeting
29 June 2026

How Peter Shor Changed the World

1994: “These algorithms take a number of steps polynomial in the input size, for example, the number of digits of the integer to be factored.”

1995: “It is shown how to reduce the effects of decoherence for information stored in quantum memory, assuming that the decoherence process acts independently on each of the bits stored in memory.”

1996: “This paper shows both how to correct errors in encoded qubits using noisy gates and also how to compute on these encoded qubits without ever decoding the qubits.”

Unruh, Physical Review A,
Submitted June 1994

PHYSICAL REVIEW A VOLUME 51, NUMBER 2 FEBRUARY 1995

Maintaining coherence in quantum computers

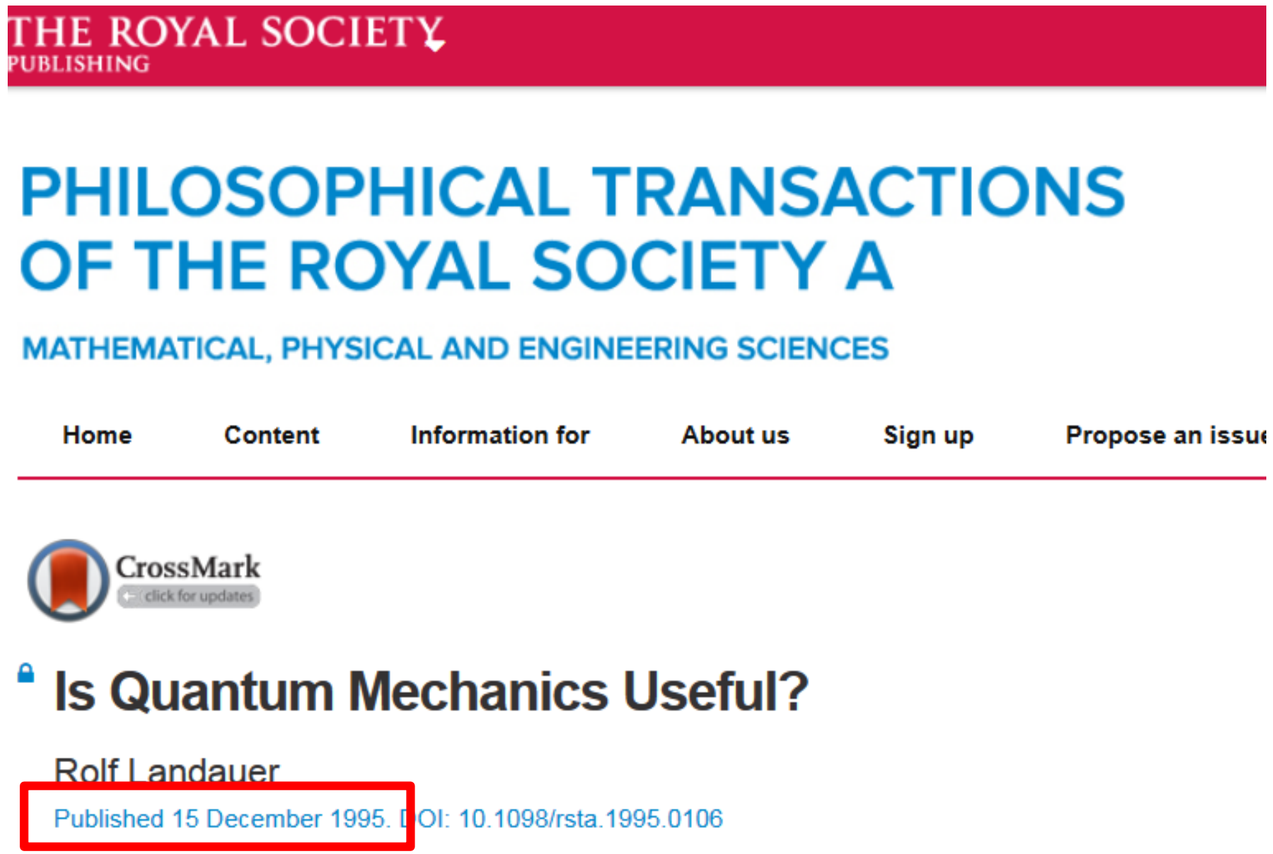
W. G. Unruh*
Canadian Institute for Advanced Research, Cosmology Program, Department of Physics, University of British Columbia, Vancouver, Canada V6T 1Z1
(Received 10 June 1994)

The effects of the inevitable coupling to external degrees of freedom of a quantum computer are examined. It is found that for quantum calculations (in which the maintenance of coherence over a large number of states is important), not only must the coupling be small, but the time taken in the quantum calculation must be less than the thermal time scale $\hbar/k_BT$. For longer times the condition on the strength of the coupling to the external world becomes much more stringent.

PACS number(s): 03.65.−w

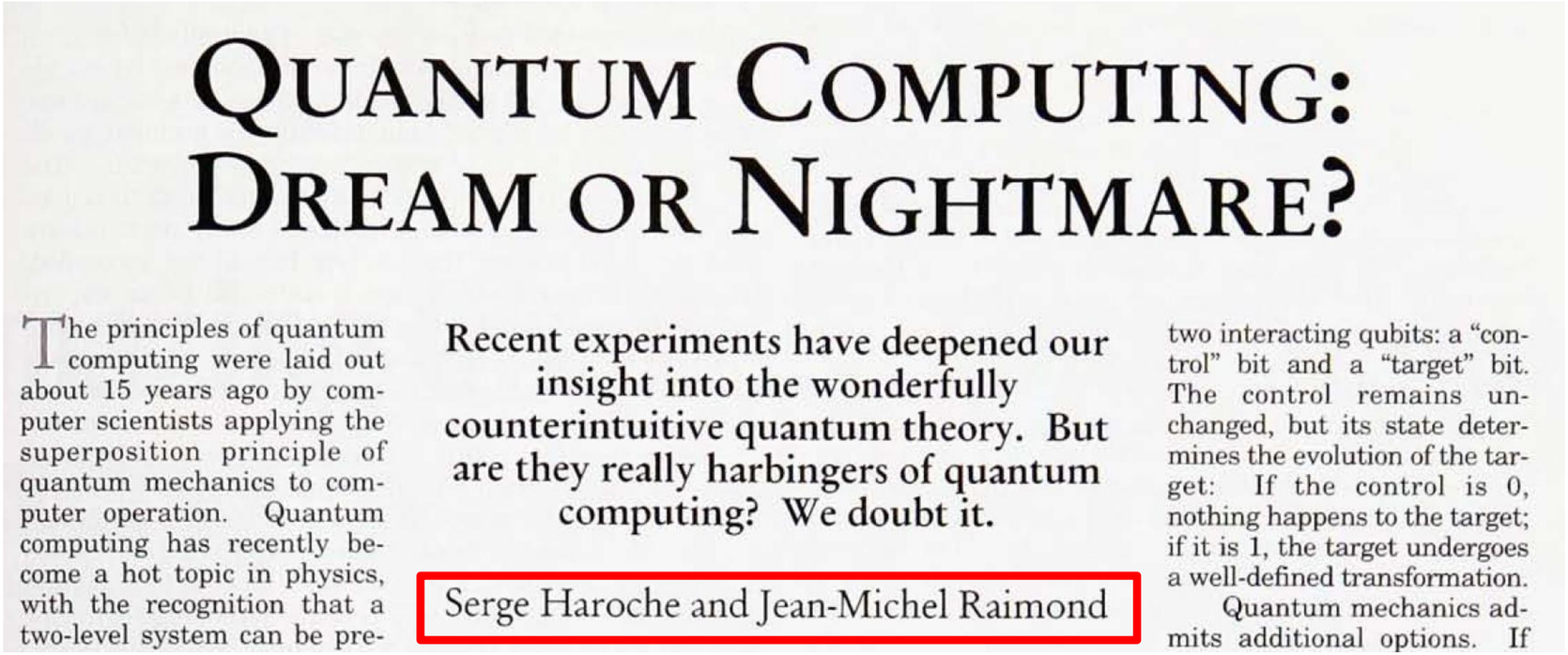
"The thermal time scale thus sets a (weak) limit on the length of time that a quantum calculation can take."

Landauer, Philosophical Transactions, Published December 1995



"…small errors will accumulate and cause the computation to go off track."

Haroche and Raimond, Physics Today, Published August 1996



QUANTUM COMPUTING: DREAM OR NIGHTMARE?

Recent experiments have deepened our insight into the wonderfully counterintuitive quantum theory. But are they really harbingers of quantum computing? We doubt it.

Serge Haroche and Jean-Michel Raimond

The principles of quantum computing were laid out about 15 years ago by computer scientists applying the superposition principle of quantum mechanics to computer operation. Quantum computing has recently become a hot topic in physics, with the recognition that a two-level system can be pre-

two interacting qubits: a "control" bit and a "target" bit. The control remains unchanged, but its state determines the evolution of the target: If the control is 0, nothing happens to the target; if it is 1, the target undergoes a well-defined transformation.

Quantum mechanics admits additional options. If

Therefore we think it fair to say that, unless some unforeseen new physics is discovered, the implementation of error-correcting codes will become exceedingly difficult as soon as one has to deal with more than a few gates. In this sense the large-scale quantum machine, though it may be the computer scientist's dream, is the experimenter's nightmare.

Why quantum computing is hard

We want qubits to interact strongly with one another.

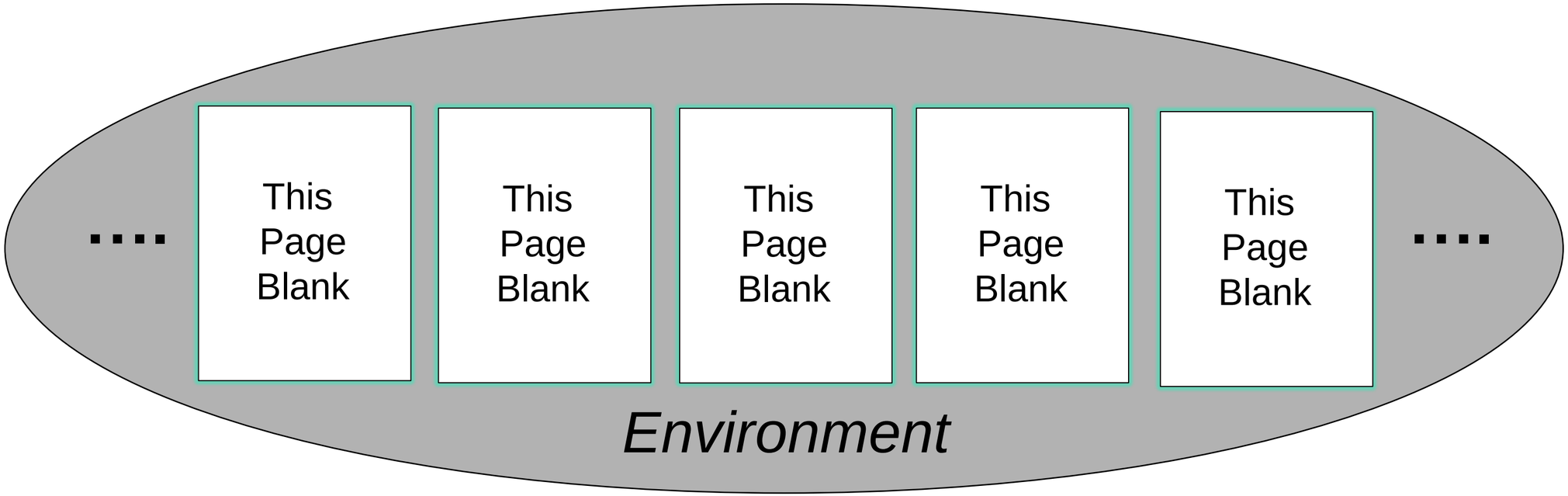
We don’t want qubits to interact with the environment.

Except when we control or measure them.

ERROR!

To resist decoherence, we must prevent the environment from "learning" about the state of the quantum computer during the computation.

Quantum error correction



The protected “logical” quantum information is encoded in a highly entangled state of many physical qubits.

The environment can't access this information if it interacts locally with the protected system.

From NISQ to FASQ

What we have now:

-- *Noisy Intermediate-Scale Quantum (NISQ) machines*.

-- Capable of performing thousands of two-qubit operations.

-- Becoming useful for scientific exploration.

-- Limited commercial value.

What we want to have:

-- *Fault-Tolerant Application-Scale Quantum (FASQ) machines*.

-- Capable of performing billions or trillions of two-qubit operations.

-- Opening a wide variety of scientific and commercial applications.

-- Need to improve error rates by many orders of magnitude!

-- Quantum error correction is essential for crossing from NISQ to FASQ.

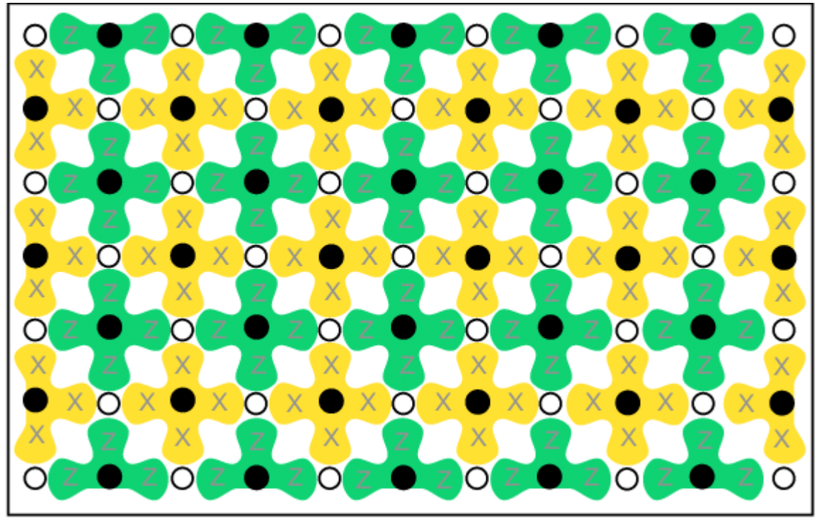
Overhead cost of fault tolerance

$$P_{\text{logical}} \approx C\left(P_{\text{physical}} / P_{\text{threshold}}\right)^{(d+1)/2}$$

$$d = \sqrt{n}, \quad C \approx 0.1, \quad P_{\text{threshold}} \approx .01$$

surface code

Suppose $P_{\text{physical}} = .001, P_{\text{logical}} = 10^{-11}$
$\Rightarrow d = 19, n = 361$ physical qubits per logical qubit,
plus a comparable number of ancilla qubits for syndrome measurement. (Improves to $d = 9$ for $P_{\text{physical}} = 10^{-4}$.)



white = data
black = check

Quantum low-density parity-check codes

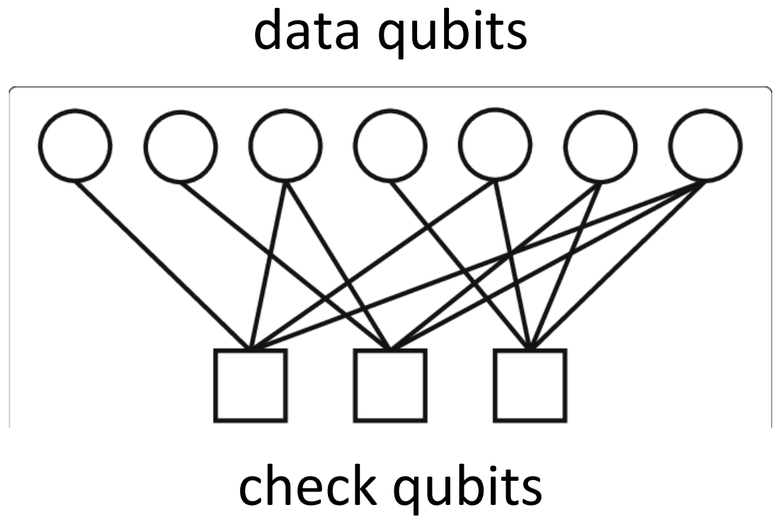
The n physical qubits obey n-k constraints ("checks"), leaving k "logical" qubits.

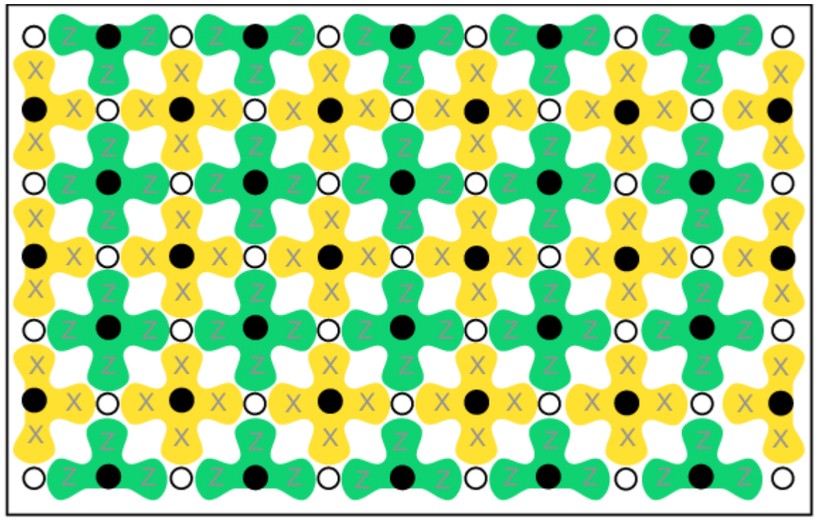
Each logical operator acts on at least d physical qubits ("distance" d).

Each constraint acts on few physical qubits, and each physical qubit is involved in few constraints.

Surface code: checks are geometrically local in 2 dimensions. k=1 logical qubits per block.

High-rate codes: large k and d via nonlocal connectivity.





surface code

The joy of nonlocal connectivity

Long-range connections might require complex fabrication in solid state devices.

More natural for movable qubits in ion traps and neutral atom arrays.

Higher encoding rates with quantum low-density parity-check (qLDPC) codes. Reduced number of physical qubits, high threshold, feasible decoders.

Universal gate gadgets for qLDPC codes with reduced spacetime cost.

These advantages partially compensate for the slow clock speed.

What *are* these things?

Bivariate Bicycle (BB) Codes
Hypergraph Product (HGP) Codes
Quasi-Cyclic Lifted Product (LP) Codes
Quantum Tanner Codes
Syndrome decoding by Belief Propagation (BP)
Pauli-Based Computation (PBC)
Surgery
Algorithmic Fault Tolerance (AFT)
And lots of other stuff!

Quantum Physics

[Submitted on 9 Jun 2026]

Handbook of Error-Correcting Codes

Victor V. Albert, Philippe Faist

Barcode scans, clear phone calls, reliable data storage, satellite communication, and large-scale quantum computation are all made possible by error correction. We present a handbook version of The Error Correction Zoo, a curated reference of methods for protecting classical or quantum information from errors during storage and transmission. The handbook includes descriptions of these error-correcting codes and a classification according to the symbols they use. It also catalogues relations among codes and related objects such as sphere packings, lattices, designs, groups, and classical and quantum phases of matter. The collection is intended both as a rigorous reference and as a practical aid for tracing the web of code relationships and uncovering new connections.

Comments: 440 classical codes, 619 quantum codes, 15 c-q codes. Online zoo at this https URL. Notify zookeeper of errors or issue a pull request at this https URL

Subjects: **Quantum Physics (quant-ph)**; Strongly Correlated Electrons (cond-mat.str-el); Information Theory (cs.IT); Combinatorics (math.CO); Metric Geometry (math.MG)

https://preskill.caltech.edu/ph219/ph219_2026.html

Ph219/CS219
Quantum Computation
Spring 2026

Course description: Ph/CS 219C is the third term in a three-term course on quantum computation and quantum information science. Topics covered in 219A included density operators, quantum operations, quantum entanglement, quantum circuits, and quantum algorithms. Ph/CS 219B covered quantum error correction and fault-tolerant quantum computing.

Ph/CS 219C will build on 219B by covering additional topics related to error correction and fault tolerance as well as other topics to be decided. But the course should be accessible to students who did not take 219B.

Instructor:
John Preskill, 206 Annenberg, X-6691, email: preskill(at)caltech(dot)edu

Teaching assistants:
Kyle Gulshen, email: kgulshen(at)caltech(dot)edu
Preksha Naik, email: pnaik(at)caltech(dot)edu
TA Office hours: Tuesdays on weeks where problem sets are due. Tuesday 2-3pm in Downs 107. Please also feel free to email TAs any time to request additional office hours.

Class meetings:
Monday and Wednesday 2:30 – 3:55 pm in 269 Lauritsen, starting March 30. The lectures are in-person only and will not be recorded.
There will be no lecture on May 6.

Homework assignments and grading:
The course is graded pass-fail. Homework will be submitted, and graded homework will be returned, using Canvas.

You may receive partial credit if you describe a thoughtful approach to the problem, even if you are unable to solve it completely.

References:
The primary reference for most of the lectures will be these lecture notes (JP). Other useful books are *Quantum Computation and Quantum Information* by Nielsen and Chuang (NC), *Classical and Quantum Computation* by Kitaev, Shen, and Vyalyi (KSV), *Quantum Computing Since Democritus* by Aaronson, *The Theory of Quantum Information* by Watrous, and *Quantum Information Theory* by Wilde.
Other recommended lecture notes: John Watrous, Umesh Vazirani, Andrew Childs, Scott Aaronson, Ronald de Wolf
The first few weeks of the course will draw on material in Chapter 7 of the lecture notes. Notes and references for subsequent lectures will be posted.

Lectures:
March 30 and April 1: Lectures 1 and 2. Error correction review, erasure and correctability. 7.3, 7.10
April 6 and 8: Lectures 3 and 4. Cleaning lemma, bounds on local codes, string operators. 7.14, 7.19, 7.20
April 13: Lecture 5. Codes from circulant matrices and polynomials. Notes.
April 15: Lecture 6. Bivariate bicycle codes. Notes. Reference.
April 20: Lecture 7. Hypergraph product codes. Notes. Reference.
April 22: Lecture 8. Chain complexes and CSS codes. Notes.
April 27: Lecture 9. Lifted product codes. Notes. Reference.
April 29: Lecture 10. Syndrome decoding. Notes. Reference 1. Reference 2.
May 4: Lecture 11. Belief propagation. Notes.
May 6: No Lecture.
May 11: Lecture 12. Quantum computing by measurement. Notes.
May 13: Lecture 13. Pauli-based computation. Notes. Reference.
May 18: Lecture 14. One-way quantum computer. Notes.
May 20: Lecture 15. Lattice surgery. Notes. Reference 1. Reference 2.
May 27: Lecture 16. Gauging logical operators. Notes. Reference.
June 1: Lecture 17. Algorithmic fault tolerance. Notes. Reference 1. Reference 2.
June 3: Lecture 18. Clifford hierarchy. Notes. Reference.

See here for notes and references

Lecture 17
1 June 2026
Ph/CS 219

Algorithmic Fault Tolerance

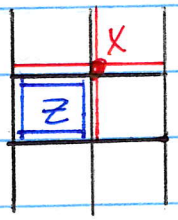
We've discussed Pauli-based computation (PBC): Universal quantum computation via $|T\rangle$ prep and Pauli-product measurement (PPM), and we've discussed how to do PPM via surgery on either many surface code blocks or blocks of high-rate codes (the former with 2D connectivity, the latter with nonlocal connectivity). In surgery, we introduce ancillas and deform the code. For fault-tolerance, we consider qLDPC codes with distance d, and formulate the deformed code so that it, too, is qLDPC with old) distance, and in addition the target PP is in the stabilizer of the deformed code. Therefore we can perform PPM by measuring

Calderbank-Shor-Steane (CSS) Codes

Quantum code from 2 classical linear codes.

Parity checks: H_X = "X-type"

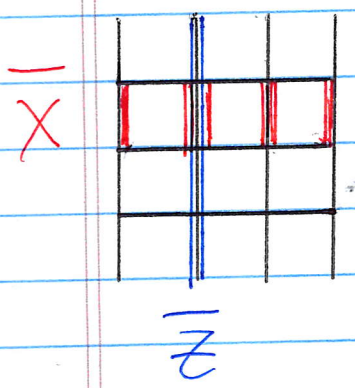
H_Z = "Z-type"



Checks commute: $H_Z H_X^T = 0 = H_X H_Z^T$

Logical Pauli Operators: $\mathcal{L}_X = \frac{\ker(H_Z)}{\mathrm{Im}(H_X^T)}$

$\mathcal{L}_Z = \frac{\ker(H_X)}{\mathrm{Im}(H_Z^T)}$



Distance:

d_X = Lowest weight of logical X

d_Z = Lowest weight of logical Z

Cyclic Symmetry and Polynomials

Repetition code on a cycle: translation invariant

Toric code: Translation inv. in hor and vert directions

Cyclic Shift: $X = \begin{pmatrix} 0&0&0&1 \\ 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \end{pmatrix}$, $X^T = \begin{pmatrix} 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \\ 1&0&0&0 \end{pmatrix} = X^{-1}$

X: $1 \to 2 \to 3 \to 4 \to 1$, X^{-1}: $1 \to 4 \to 3 \to 2 \to 1$

$X^n = I$ if X is $n \times n$ matrix.

Circulant matrices = polynomials.

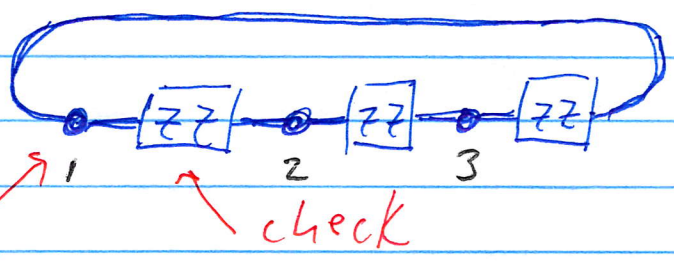
Binary matrix, determined by first row (or column) = element of polynomial ring $R = \mathbb{F}_2[x]/(x^n - 1)$.

Matrix $A = c_0 I + c_1 X + c_2 X^2 + \cdots + c_{n-1} X^{n-1}$

Matrix multiplication $\equiv$ polynomial mult.

Classical repetition code:

$H = 1 + X = \begin{pmatrix} 1&0&1 \\ 1&1&0 \\ 0&1&1 \end{pmatrix}$



Toric Code from Polynomials

Two cyclic symmetries: x (hor) and y (vert)

Two "registers": | vert. edge, — hor. edge.

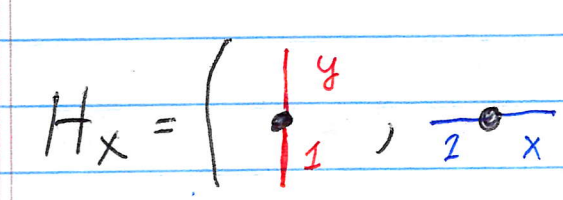
$L \times M$ torus $\Rightarrow x^L = 1 = y^M$.

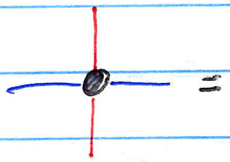
$$H_x = (1+y, 1+x)$$

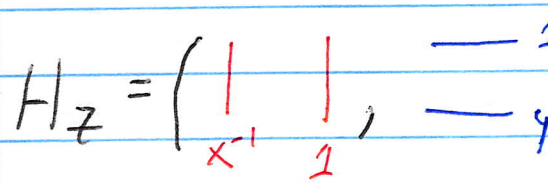
$$H_z = (1+x^{-1}, 1+y^{-1}) \Rightarrow H_z^T = (1+x, 1+y)$$

Reg 1: vert Reg 2: hor

Checks commute: $H_x H_z^T = (1+y)(1+x) + (1+x)(1+y) = 0$

$H_x =$ 

=  = site operator

$H_z =$ 

=  = plaquette operator

Bivariate Bicycle (BB) Codes

$H_X = \left(A(x,y), B^T(x,y) \right)$

$H_Z = \left(B(x,y), A^T(x,y) \right)$

Two cyclic symmetries.
Two registers.

Reg. 1 Reg. 2

checks commute: $H_X H_Z^T = AB^T + B^T A = 0$

Check weight = number of terms

K logical qubits: number of relations among checks.

Example: Gross code

$$[[n, K, d]] = [[144, 12, 12]]$$

Torus $12 \times 6 \Rightarrow x^{12} = 1 = y^6$

$$A' = y^{-1} A = 1 + y + y^{-1} x^3$$

$$B'^T = x^{-1} B^T = 1 + x + x^{-1} y^3$$

Check weight 6

As for toric code

longer reach

X check includes:

Register 1 3 steps right, 1 step down.

Register 2 1 step left, 3 steps up.

Hypergraph Product (HGP) Codes

A is $r_A \times n_A$ matrix; B is $r_B \times n_B$ matrix.

$$H_X = \left(A \otimes I_{n_B},\ I_{r_A} \otimes B^T \right) \quad r_A n_B \text{ checks}$$

$$H_Z = \left(I_{n_A} \otimes B,\ A^T \otimes I_{r_B} \right) \quad n_A r_B \text{ checks}$$

Reg 1: $n_A n_B$ qubits

Reg. 2: $r_A r_B$ qubits

A acts "horizontally", B acts "vertically."

checks commute: $H_X H_Z^T = A \otimes B^T + A \otimes B^T$

Reg. 1 Reg. 2

$K \geq$ #qubits − #checks

$$= n_A n_B + r_A r_B - r_A n_B - n_A r_B = (n_A - r_A)(n_B - r_B)$$

In fact, $K = K_A K_B + K_{A^T} K_{B^T}$ Künneth Formula

HGP of two high-rate classical codes → High-rate quantum codes.

$K_A = O(n_A)$, $K_B = O(n_B) \Rightarrow K = O(n)$

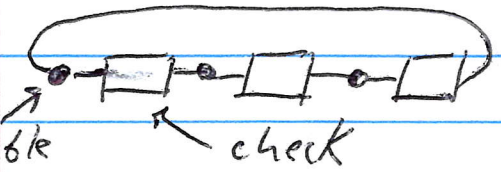
HGP of two classical LDPC codes → quantum LDPC

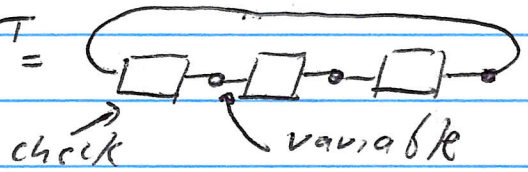
X check wt. = wt. of A row + wt. of B column.

Z check wt. = wt. of B row + wt. of A column.

Toric Code as HGP of two classical rep. codes

Repetition code on a cycle and its transpose code.

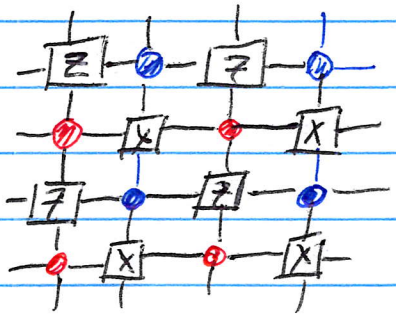
$H =$ 

$H^T =$ 

same code!

$H_X = (A \otimes I, I \otimes B^T)$ A acts hor. on reg 1, B^T acts vert on reg. 2

$H_Z = (I \otimes B, A^T \otimes I)$ A^T acts hor on reg 2, B acts vert on reg. 1



checks are weight 2+2=4.

It's the Toric code again!

distance $d = \min(d_A, d_B, d_A^T, d_B^T)$

$$d_{toric} = d_{rep}$$

If classical codes and their transpose codes have $d = O(n)$, then quantum code has $d = O(\sqrt{n})$.

Can also consider "higher dimensional" products.

useful for non-Clifford gates and "metachecks".

Lifted Product (LP) codes

$$H_X = (A \otimes I_{n_B},\ I_{r_A} \otimes B^T) \qquad A \text{ is } r_A \times n_A$$
$$H_Z = (I_{n_A} \otimes B,\ A^T \otimes I_{r_B}) \qquad B \text{ is } r_B \times n_B$$

Now matrix entries are not $\{0,1\}$ but rather elements of polynomial ring $R = \mathbb{F}_2[x]/(x^\ell - 1)$, e.g. monomials x^m, $m \in \{0,1,2,\cdots,\ell-1\}$.

"Lift" the "seed matrices" A and B by adding a "fiber" above each point in a 2D "base space".

Compared to seed code, number of qubits and number of checks increases by factor ℓ, without increasing check weight if entries are monomials.

Example (Cain, Xu et al. 2026): $A = B$, $\ell = 33$.

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & x^{14} & x^{19} & x^{11} & x^{26} \\ 1 & x^{13} & x^{2} & x^{15} & x^{21} \end{pmatrix}$$

X and Z check matrices have:

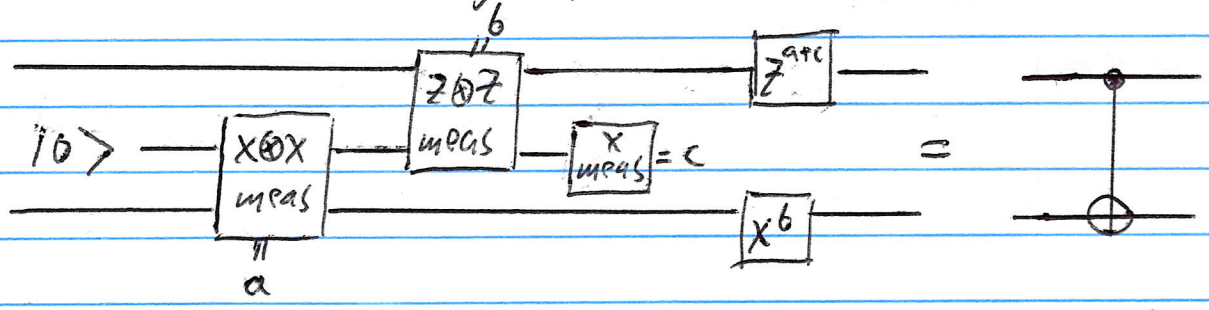
$n = (5^2 + 3^2)33 = 1122$ columns.

$5 \times 3 \times 33 = 495$ rows.

$K \geq 1122 - 2(495) = 132$ (actually $K = 148$)

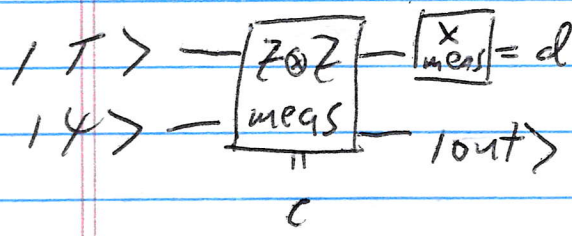
$d \leq 20$ and check weight $5+3 = 8$

Quantum Computing by Nondestructive Measurement



Heisenberg picture: Track how Pauli operators transform. CNOT up to a "Pauli frame" update.

For universality, a non-Clifford gate, e.g. $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$. Done by measurement and T-state prep: $|T\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\pi/4}|1\rangle\right)$



$$|\text{out}\rangle = \begin{cases} c=0: T|\psi\rangle \\ c=1: T^{-1}|\psi\rangle \end{cases} \times \text{Pauli}$$

= "clean up" with $S = T^2$.

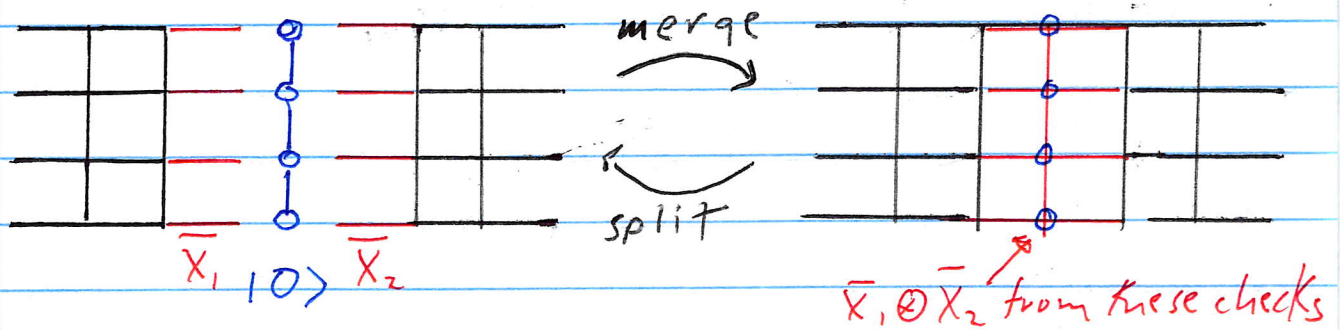
$|T\rangle$ preparation and nondestructive Pauli product measurement are all you need!

$|T\rangle$ prep by e.g. magic state distillation.

Pauli product measurement by surgery.

Pauli Product Measurement by Surgery

Measuring $X_1 \otimes X_2$ for two surface code blocks:



- Deform to a new code such that target Pauli product is a product of low-weight checks
- Measure old) times to determine checks robustly.
- Deform back to original code.

Generalize: measure high-weight Pauli products in surface code

Generalize: extend to deformation of high-rate codes.

What *are* these things?

Bivariate Bicycle (BB) Codes
Hypergraph Product (HGP) Codes
Quasi-Cyclic Lifted Product (LP) Codes
Quantum Tanner Codes
Syndrome decoding by Belief Propagation (BP)
Pauli-Based Computation (PBC)
Surgery
Algorithmic Fault Tolerance (AFT)
And lots of other stuff!

The most important ideas in physics in the past 40 years?

1. The holographic principle (1994)
2. Topological quantum order (1989)
3. Quantum error correction (1995)

All three ideas are closely related!

The common thread: many-particle quantum entanglement.