Quantum Error Correction Codes

Boyu Zhou
University of Arizona, Department of Physics
(Dated: December 6, 2020)

Quantum error correction will play an essential role in the realisation of quantum computing. So understanding quantum error correction codes is basic to understand and be familiar with the current and future of the quantum information processing and the quantum computation. In this review paper, we first claim why developing the quantum error correction codes are critical in Sec. I. Then it's also necessary to brief the classical error correction theory to better understand the key points of quantum error correction theory in Sec. II. Starting from the Sec. III, IV, and V, we provide an introductory guide to the fundamental of the theory by using the simplest but important example codes. Finally, we discuss the issues that are the big challenges we need to overcome.

I. INTRODUCTION

In the past thirty years, the development of traditional computers has increased by 100,000 times due to the development of silicon chips. This is the so called Moore's Law[1], which predicts that the number of transistors on a microchip doubles every two years. However, experts agree that computers should reach the physical limits of Moore's Law at some point in the 2020s[2]. The second problem for traditional computers is that, as Feynman had pointed out, there seemed to be essential difficulties in simulating quantum mechanical systems on classical computers. Thus, we need to design quantum computers urgently.

But noise is always a great bane of information processing systems. It's quite important to protect information against the effects of noise, especially the noise has more influence on quantum computers than the classical computers. There is already a quite complete theory of classical error correction[3]. But the existing classical methods for quantum error correction is not enough. Due to the no-cloning theorem[4] and the wave function collapse, quantum information cannot be duplicated in the same way as classical information. All these difficulties and challenges require us to construct a new theory, quantum error correction theory.

II. FROM CLASSICAL TO QUANTUM ERROR CORRECTION

A. Classical error correction

We will start from the theory of classical error correction. As a complex computing system, the classical computer has inevitable noise during operation, so it faces problems such as gate operation errors and inaccurate calculations. But for classical computers, we have developed and widely used classical error correction theory. By applying error correction technology, we can obtain reliable calculation results if the noisy does not exceed its threshold. The key ideal to protect a message against noise, we should encode the message by adding some redundant information to the message[5].

The simplest example of an error correction code is the three-bit repetition code, that is, replace one bit with its three copies:

$$\begin{aligned} 0 &\longrightarrow 000 \\ 1 &\longrightarrow 111 \end{aligned} \tag{1}$$

In this way, if at most one bit has an error, like, 000 becomes 001. Then by observing the value of each bit and comparing it in pairs , we can find out the third bit has flipped, then restore the third bit, via a majority vote[6].

This three-repetition code belongs to linear codes, which is the most widely used classical error-correcting codes. Suppose we wish to encode k bits using n bits. The data can be represented as a k-dimensional binary vector v. Because we are dealing with binary vectors, all the arithmetic is mod two. For a linear code, the encoded data is then Gv for some nk matrix G, which is independent of v. G is called the generator matrix for the code. Its columns form a basis for the k-dimensional coding subspace of the n-dimensional binary vector space, and represent basis codewords. The most general possible codeword is an arbitrary linear combination of the basis codewords; thus the name "linear code"[7]. So the three-bit repetition code mapping a single bit to three copied is specified by the generator matrix:

$$G = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{2}$$

$G[0] = (0,0,0)$, $G[1] = (1,1,1)$, so this is a $[3,1]$ code. The $[n,k]$ code means a code using n bits to encode k bits of information.

To make error-correction and recovery, we need the parity check matrix P,which is an $(n-k) \times n$ matrix of 0,1 of maximal rank $n-k$ with $PG = 0$. Since any codeword x has the form Gv, $Px = PGv = 0v = 0$, and P annihilates any codeword. If an error e happens, the corrupted codeword now is $x' = x + e$. It is easy to check $Px' = Pe$, which is called the error syndrome. This tells what and where the error e is, then we can correct this error easily.

B. Basics of Quantum Error Correction

In the classical systems, bit is the fundamental unit. A bit can only take a binary number like 0 or 1. But in the quantum systems, the fundamental unit is qubit, which can exist in coherent superpositions of $|0\rangle$ and $|1\rangle$ state. The general qubit state form is as follows:

$$|\psi\rangle = a|0\rangle + b|1\rangle \quad (3)$$

where a and b are complex numbers that satisfy $|a|^2 + |b|^2 = 1$.

The effect of superposition cause two difference between classical and quantum information, one is the quantity. For a classical memory composed of N bits, it can only store one of 2^n binary numbers; however,for a quantum memory composed of N qubits, it can store all these 2^n binary numbers at the same time. The second difference is the error types. In classical information, the only error-type to be considered is the bit-flip that takes $0 \leftrightarrow 1$. But in quantum information, it can happen phase-flip that takes $1 \leftrightarrow -1$ which cause the state $|\psi\rangle$ becomes the state $|\psi'\rangle = a|0\rangle - b|1\rangle$.

Any error process of a single qubit can be represent by a sum of the Pauli set $\{X,Y,Z,I\}$, where

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (4)$$

Pauli X-type errors can be thought of as quantum bit-flips that map $X|0\rangle = |1\rangle$, and $X|1\rangle = |0\rangle$; Pauli Z-type errors can be thought of as quantum phase-flips that map $Z|0\rangle = |0\rangle$, and $Z|1\rangle = -|1\rangle$; Pauli Y gates are the combination of X and Z, since $Y = iXZ$; I gate means no error since $I|0\rangle = |0\rangle$, and $I|1\rangle = |1\rangle$.

With the help of Pauli gates, we can try to understand what properties are essential for a more general quantum error-correcting code. In order to correct two errors E_a and E_b, it requires we are always able to distinguish the two different errors in any possible code states $|\psi\rangle$

$$\langle\psi| E_a^\dagger E_b |\psi\rangle = 0 \quad (5)$$

In fact, this should always be distinguishable for any different states. And more importantly, we cannot learn anything about the actual state of the code within the coding space which is the requirement of the quantum physics to protect the superposition. Thus, this quantity must be the same for all the basis codewords:

$$\langle\psi_i| E_a^\dagger E_b |\psi_j\rangle = C_{ab}\delta_{ij} \quad (6)$$

where C_{ab} is a Hermitian matrix independent of i and j. This is a necessary condition for all the codes to correct the errors which was found by Knill and Laflamme[8] and Bennett et al[9]. As illustrated in FIG. 1, the left part is "good" coding space with orthogonal error operators, and the right part is "bad" coding space with non-orthogonal error operators.

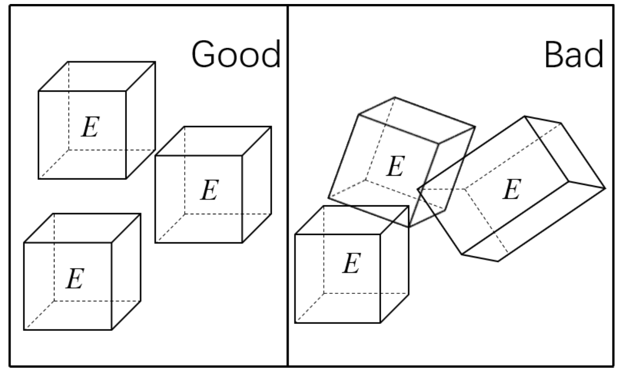

FIG. 1. "Good" and "bad" coding space.

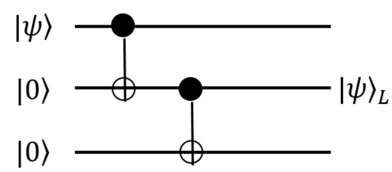

FIG. 2. Encode stage of the three-qubit error correction code

III. SIMPLE CODE EXAMPLES

A. The three-qubit error correction code

In fact, we should emphasize that the three-qubit error correction code is not a "good" code for quantum information, but this is a good start for all the reasonable codes. The general state for three-qubit error correction code is:

$$|\psi\rangle_L = a|0\rangle_L + b|1\rangle_L \quad (7)$$

where $|0\rangle_L$ and $|1\rangle_L$ are logical basis states, that means $|0\rangle_L = |000\rangle$ and $|1\rangle_L = |111\rangle$.

FIG. 2 shows how to use a single qubit $|\psi\rangle$ entangled with two ancilla qubits and two CNOT gates to encode a logical qubit state. After the encoding stage, we are able to correct errors using this code logical qubit $|\psi\rangle_L$. FIG. 3 shows the correction circuit. Each logical qubit will first have one kind of error from the error set $E = \{X_1, X_2, X_3\}$, then the syndrome extraction process by introducing two ancilla qubits and performing CNOT gatess will be applied. As shown in Table. I, by measuring ancilla qubits, we can find what errors happened in the logical qubit and then we can correct it.

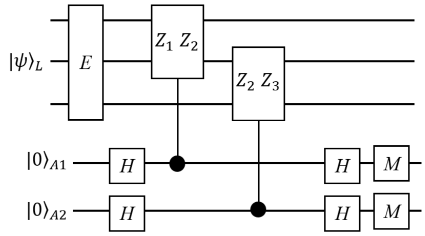

FIG. 3. Correction circuit

B. The nine-qubit error correction code

Since the three-qubit code can only correct bit-filp errors, we need a code can correct both bit-flip and phase-flip error. This was first to be successfully realized by Shor[10] in 1995 and is based largely on the three-qubit code. According to Shor's theory, the logical qubits states for the code are:

$$|0\rangle_L = \frac{1}{2\sqrt{2}}(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) \tag{8}$$

$$|1\rangle_L = \frac{1}{2\sqrt{2}}(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) \tag{9}$$

Each logical qubit contains three blocks of the same three-qubit state. The circuit to perform the encoding is shown in FIG. 4. This redundancy is enough to correct both bit-flip error and phase-error. Let's see how this works.

For bit-filp errors, for example, the state $|000\rangle \pm |111\rangle$ in the first block changes to the state $|010\rangle \pm |101\rangle$, which is also the error gate X_2. Likely what we did for the three-qubit code earlier, by performing the correction circuit shown in FIG. 3, the syndrome on the ancilla qubits will give 11, we can know the bit-filp error happens in the second qubit and then corrected it. For phase-filp error, this will change the sign in one block,$|000\rangle \pm |111\rangle$ becomes $|000\rangle \mp |111\rangle$. By comparing the phase in blocks, instead of measuring the relative phase in each block (This will change the coding information), you can find out the block whose phase has changed. The result only allows you to determine which block has a sign different from the other two, then the unitary phase transformation can be applied to correct the error.

The nine-qubit code is easy to understand, however it is not the most effective code, because it cost too much to use 9 qubits to protect just 1 qubit information. In 1996, Steane[11] proposed a quantum code that uses 7 qubits to encode 1 qubit. Daniel and Gottesman[12] first introduced the concept of ”stabilizer” to make the theory of quantum error correction more systematic and perfect.

TABLE I. All bit-flip errors on the three qubit code.

Error	Syndrome on ancilla qubits
X_1	10
X_2	11
X_3	01

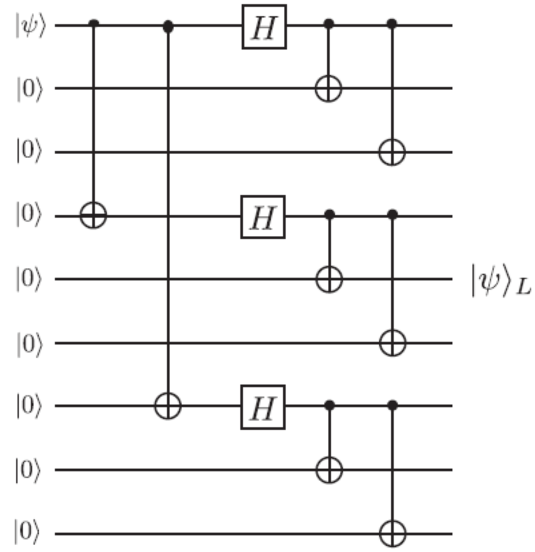

FIG. 4. Encode stage of the nine-qubit error correction code

IV. STABILIZER CODES

A. Properties of stabilizer codes

We first introduce stabilizer formalism. This formalism requires to describe quantum states by using operators. For any state $|\psi\rangle$, if the operator P satisfies $P|\psi\rangle = |\psi\rangle$, or equivalently, $|\psi\rangle$ has eigenvectors with +1 eigenvalues, we say the operator P stabilizes the state $|\psi\rangle$. For example, for single qubit $|0\rangle$:

$$Z|0\rangle = |0\rangle \tag{10}$$

Z is the stabilizer of the single-qubit state $|0\rangle$. As for multi-qubit states, like EPR state, $|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ has:

$$X_1X_2|\phi^+\rangle = |\phi^+\rangle \tag{11}$$

$$Z_1Z_2|\phi^+\rangle = |\phi^+\rangle \tag{12}$$

We say the state $|\phi^+\rangle$ is stabilized by the operators X_1X_2 and Z_1Z_2. In fact, $|\phi^+\rangle$ is the unique state (up to a globe phase) which is stablized by the two operators.

Above all, the stabilizers P_i must satisfy the following properties:
(1) They must be Pauli-group elements, $P_i \in \mathcal{G}_n$, where $\mathcal{G}_n$ is the Pauli group over n-qubits. Here the n-qubits Pauli group is N fold tensor product of single-qubit Pauli group $\mathcal{G}$.

$$\mathcal{G} \equiv \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\} \tag{13}$$

(2) They must stabilize all the logical states $|\psi\rangle_L$ of the code. This means that each P_i has +1 eigenvalues for all possible values of $|\psi\rangle_L$.
(3) All the stabilizers of a code must commute with one another, so that $[P_i, P_j] = 0$ for all i and j. This property is necessary so that the stabilizers can be measured simultaneously.

Let's see how the general error correction procedure for a single cycle of a stabilizer code works. First, we need to prepare the logical states $|\psi\rangle_L$, that is encoded stage. Normally, the initial states will entangle with ancilla states. Then the encoded logical states are subject to an error process E. Next, we will measure the stabilizers by the syndrome extraction method, and we can read out the error from the ancilla systems. In the end, the decoding step, we need to apply an error correction procedure by choosing a unitary operation $\mathcal{R}$ to return the logical state to the codespace. We say the error correction is successful if the decode step has the results:

$$\mathcal{R}E\,|\psi\rangle_L = |\psi\rangle_L \tag{14}$$

It's trivially to satisfy this equation by letting $\mathcal{R} = E^\dagger$. But this is not the only solution in some cases. In fact, for any $\mathcal{R}$, if it equals $\mathcal{R}E = P$, this above equation will be satisfied. So, if the choice of $\mathcal{R}$ is unique, this is a degenerate code; if this is not unique, this is a non-degenerate code.

B. Example: the five-qubit code

The five-qubit code[13] is the smallest quantum stabilizer code that corrects for a single error. To prepare it, we can use the generation operators $P_i \in \mathcal{S}$, where $\mathcal{S}$ is the stabilizer group, as shown in TABLE. II, to encode the logical states:

$$\begin{aligned}|0\rangle_L &= \sum_{\mathcal{S}} P_i\,|00000\rangle \\ &= |00000\rangle + |10010\rangle + |01001\rangle + |10100\rangle \\ &+ |01010\rangle - |11011\rangle - |00110\rangle - |00101\rangle \\ &- |10001\rangle - |01100\rangle - |10111\rangle + |00101\rangle\end{aligned} \tag{15}$$

and

$$\begin{aligned}|1\rangle_L &= \bar{X}\,|0\rangle_L \\ &= |11111\rangle + |01101\rangle + |10110\rangle + |01011\rangle \\ &= +\,|10101\rangle - |00100\rangle - |11001\rangle - |00111\rangle \\ &= -\,|00010\rangle - |11100\rangle - |00001\rangle - |10000\rangle \\ &= -\,|01110\rangle - |10011\rangle - |01000\rangle + |11010\rangle\end{aligned} \tag{16}$$

The stabilizers encode five physical qubits into one logical qubit to correct a single X, Y or Z error. Unlike the nine-qubit codes which can correct 2 errors, this code can only correct a single error. And this is also a degenerate code since the recovery $\mathcal{R}$ is unique.

C. Example: the Shor code

The Shor's code, or the nine-qubit code has been discussed before, but this time, let's check the code in the view of stabilizer codes. In fact, the nine-qubit code is the combine of the two three-qubit codes for phase-flip and error-flip. The three-qubit error-flip code is:

$$|0\rangle_L = |000\rangle\,,|1\rangle_L = |111\rangle \tag{17}$$

with the stabiliezrs Z_1Z_2 and Z_2Z_3. And the three-qubit phase-flip code is:

$$|0\rangle_L = |+++\rangle\,,|1\rangle_L = |---\rangle \tag{18}$$

where $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$, with the stabiliezrs X_1X_2 and X_2X_3. Combine these two types code, we get the Shor's nine-qubit code with all the stabilizers:

$$\begin{aligned}\mathcal{S} = \{&Z_1Z_2, Z_2Z_3, Z_4Z_5, \\ &Z_5Z_6, Z_7Z_8, Z_8Z_9, \\ &X_1X_2X_3X_4X_5X_6, \\ &X_4X_5X_6X_7X_8X_9\}\end{aligned} \tag{19}$$

We can see each of the X-errors produce unique syndromes. In contrast, Z-errors that occur in the same block of the code have the same syndrome. For example, if we assume that the error is Z_1, by applying the recovery operation $\mathcal{R} = Z_1$, we can get the result $\mathcal{R}Z_1\,|\psi\rangle_L = Z_1Z_1\,|\psi\rangle_L = |\psi\rangle_L$. But this is not the unique choice, if $\mathcal{R} = Z_2$, the result is still the same, $\mathcal{R}Z_1\,|\psi\rangle_L = Z_2Z_1\,|\psi\rangle_L = |\psi\rangle_L$. Thus, the Shor's nine-qubit code is a non-degenerate code.

V. FAULT TOLERANCE

In the above discussion of quantum error correction codes, we have implicitly assumed that the errors only occur in certain locations in the circuit. Like in FIG. 3, the error only happens in the error region E, and it can run perfectly in other locations without any error. This is not possible in real experiment. In fact, for many quantum computing technologies, and measurement operations can be dominant sources of error. As such, it is unrealistic to assume that any part of the circuit is error free. What's more, the quantum gates themselves could happen some systematic errors within the logical

TABLE II. The stabilizer for the five-qubit code.

Name	Operators
P_1	$XZZXI$
P_2	$IXZZX$
P_3	$XIXZZ$
P_4	$ZXIXZ$
$\bar{Z}$	$ZZZZZ$
$\bar{X}$	$XXXXX$

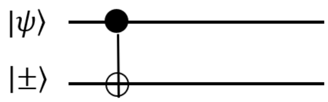

FIG. 5. An example of error propagation.

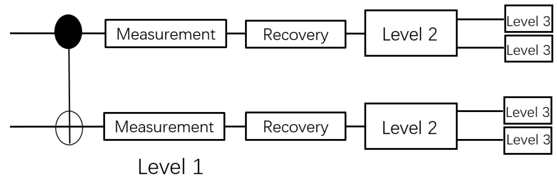

FIG. 6. An example of three level concatenated codes.

data block. For example, an error can propagate from the control qubit to the target qubit. See the example in FIG. 5, we have the states, $|\psi\rangle = a|0\rangle + b|1\rangle$, and $|\pm\rangle = |0\rangle \pm |1\rangle$. Then by performing a CNOT gate from the first qubit to the second, we get:

$$a|0\rangle(|0\rangle \pm |1\rangle) + b|1\rangle(\pm 1)(|0\rangle \pm |1\rangle) \tag{20}$$

Initially we flip the sign on the second qubit, then we will get a sign flip on the first qubit after the CNOT gate.

The basic principle of fault-tolerance is that the circuits used for gate operations and error correction procedures should not cause errors to cascade[14]. It's important to define a fault-tolerant measurement and a fault-tolerant state preparation. We say it's fault-tolerant if the failure of any single component in the procedure results in an error in at most one qubit in each encoded block of qubits at the output procedure. And if the preparing procedure is fault-tolerant, there is at most a single qubit in error in each block of qubits output from the procure. In fact, fault-tolerance is a complex conception, here we will use a simpler but easier model to understand fault-tolerance. Assume the error types on qubits are one of the four types: I, X, Y, Z. When performing gates, we allow errors to happen on two qubits in some probability with the form of tensor products of Pauli matrices. For example, an X error on the first qubit occurs just before any kind of gate. Assume this unitary operator of this gate is U, then the effective action of the circuit is $UX_1 = UX_1U^{\dagger}U = X_2U$, this means as though the error X happens before the gate, and the gate was applied correctly, but this X error occurred on both the first and the second qubits after the gate. So the question is, how to design a fault- tolerant circuit? The general structure of the circuit was first developed by Shor, and it should be noted that several more recent methods for fault-tolerant state preparation and correction now exist. Here we just want to show one simple way to understand fault-tolerant circuit.

One way is called the concatenated codes. The key point is to use iteration, by repeatedly applying the original quantum codes and constructing a hierarchy of quantum circuits. As illustrated in FIG. 6, the original circuit is C_0. The error enter on the first block is at most c_0p, where c_0 is a constant. After the second block, the probability for this circuit is at most $(c_0p)^2$. Now we will extend this quantum circuit by repeating this many times. Thus, if we concatenate k times, the failure probability is now $(c_0p)^{2^k}$. It's quite easy to find that $(c_0p)^{2^k} \leq (c_0p)^2$, only when $(c_0p)^2 = 1$ we get equivalence. And the more concatenations we have, the failure probability will be smaller when $(c_0p)^2 < 1$. Clearly, the disadvantage is that the size of the circuit increase as k time the size of the original circuit.

VI. OUTLOOK

A. Modern developments in quantum error correction codes

In the above discussion, we only focus on the most basic principals and codes examples. For the requirements of construction of large scale quantum computers, these basic codes are far not enough to use, we need more modern protocols. Since this will far outside our review paper, we only attempt to have some simple discussion of two of these more complicated codes.

The first one is called the surface code, or topological error correction, which is first introduced by Kitaev[15]. Surface code is defined on a lattice of qubits for detecting and correcting errors. The error suppression achieved by the surface code is usually estimated by simulating toy noise models describing random Pauli errors. The general design principle behind topological codes is that the code is built up by 'patching' together repeated elements. We will see that this modular approach ensures that the surface code can be straight-forwardly scaled in size whilst ensuring stabilizer commutativity. In terms of actual implementation, the specific advantage of surface code for current hardware platforms is that it requires only nearest-neighbour interactions. This is advantageous as many quantum computing platforms are unable to perform high-fidelity long-range interactions between qubits[6].

Gottesman, Kitaev and Preskill have formulated a way of encoding a qubit into an oscillator such that the qubit is protected against small shifts (translations) in phase space[16]. The idea underlying this encoding is that error processes of low rate can be expanded into small shift errors. The qubit space is defined as an eigenspace of two mutually commuting displacement operators S_q and S_p which act as large shifts in phase space. The GKP code is intrinsic fault-tolerance, and construed by only linear optical elements. These advantages make the GKP code an ideal encoding scheme in experiments.

B. Challenges need to overcome

As we have discussed lots of importance and advantages of quantum error correction codes, we cannot ignore the big challenges that need to overcome in the nowadays. (1) One is the code itself. Although quantum error correction offers a solution to control qubits in an error-free way to construct a quantum computer[6], we cannot not ignore that it requires additional large number of qubits to operate. This will significantly increase the overheads associated with quantum computing. We have developed lots of quantum error correction codes, from the earliest nine-qubit code to the surface codes and many other codes in the recent years; however, we have not found any "perfect" codes to avoid this disadvantage with retaining the advantages we need. (2) There is still an obvious disconnect between the abstract framework of quantum coding and the more physically realistic implementation of error correction for large-scale quantum information processing[14]. Even with the development of many of new quantum error correction codes, the physical construction and accuracy of current qubit fabrication is still hard to obtain benefits from them.

But the study of quantum error correction is and will still be active in the area of quantum information processing research. Although there are many difficulties, the development of quantum error correction codes proves that it is not impossible to perform reliable quantum computation. The latest results show that in quantum computers, as long as the error rate is below some certain threshold, quantum computation with arbitrary precision can be performed. We believe that with the efforts of scientists, realizing reliable quantum computing is no longer a dream.

[1] J. Birnbaum and R. S. Williams, Physics and the information revolution, Physics Today **53**, 38 (2000).

[2] D. Rotman, We're not prepared for the end of moore's law. (2020).

[3] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication, by CE Shannon (and Recent Contributions to the Mathematical Theory of Communication), W. Weaver* (University of illinois Press, 1949).

[4] W. K. Wootters and W. H. Zurek, A single quantum cannot be cloned, Nature **299**, 802 (1982).

[5] M. A. Nielsen and I. L. Chuang, Quantum computation and quantum information, Phys. Today **54**, 60 (2001).

[6] J. Roffe, Quantum error correction: an introductory guide, Contemporary Physics **60**, 226 (2019).

[7] D. Gottesman, Stabilizer codes and quantum error correction, arXiv preprint quant-ph/9705052 (1997).

[8] E. Knill and R. Laflamme, Theory of quantum error-correcting codes, Physical Review A **55**, 900 (1997).

[9] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, Mixed-state entanglement and quantum error correction, Physical Review A **54**, 3824 (1996).

[10] P. W. Shor, Scheme for reducing decoherence in quantum computer memory, Physical review A **52**, R2493 (1995).

[11] A. Steane, Multiple-particle interference and quantum error correction, Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences **452**, 2551 (1996).

[12] D. Gottesman, An introduction to quantum error correction, in *Proceedings of Symposia in Applied Mathematics*, Vol. 58 (2002) pp. 221–236.

[13] R. Laflamme, C. Miquel, J. P. Paz, and W. H. Zurek, Perfect quantum error correcting code, Physical Review Letters **77**, 198 (1996).

[14] S. J. Devitt, W. J. Munro, and K. Nemoto, Quantum error correction for beginners, Reports on Progress in Physics **76**, 076001 (2013).

[15] A. Y. Kitaev, Quantum computations: algorithms and error correction, Russian Mathematical Surveys **52**, 1191 (1997).

[16] D. Gottesman, A. Kitaev, and J. Preskill, Encoding a qubit in an oscillator, Physical Review A **64**, 012310 (2001).