Determining the threshold probabilities of error correction and error detection codes in a quantum circuit

Michele Valotti
March 16, 2018

Quantum computation is the future of modern technology, but, much like classical computing, it needs a reliable way to prevent our systems from becoming a chaotic ensemble of unpredictable information. The rules of quantum mechanics make this a challenging task, but the discovery of error detecting and correcting codes brings new hope to the possibility of scalable quantum machines, provided that the error rates are below a certain threshold. In this work we analyse the behaviour of two error correcting circuits (3-qubit code and 9-qubit code) and of an error detecting circuit (4-qubit code), attempting to find their threshold probability under the simplifying assumptions of perfectly applied quantum gates and uniform error rates. The values found were 0.5, 0.06 and 0.1 for the 3-qubit, 9-qubit and 4-qubit codes respectively.

1. Introduction

Classical, transistor based computers are ubiquitous in the modern world, and as the need for increasingly faster computational power grows, the size of the transistor on which classical chips rely must shrink. With this need for ever decreasing size come great engineering challenges, that have so far been solved by great improvements in manufacturing technologies. As we approach the physical limit of Moore's law [1], however, we realise that the quantum behaviours of microscopic transistors could be exploited, and might represent a great opportunity to exponentially increase our computational power, rather than hinder it. This calls for a shift in the way we think about computation, a change of paradigm, from classical bits to the realm of *quantum computing*.
In classical computation we work with deterministic bits, either 0 or 1, low or high voltage, but a quantum bit, or a *qubit*, as it is referred to in the literature, can be in a superposition of its *zero* and *one* states, oscillating in a sinusoidal fashion between the two according to the dynamics of Rabi oscillations. Practically, a qubit can be implemented in many different ways, ranging from the ground and excited state of an electron, to the different polarisation of an electromagnetic wave [2]. The benefit of working with quantum states comes from the way their state spaces scale: n qubits can be in a superposition of 2^n states, as opposed to n^2 for classical bits. This means that some calculations that would be unachievable with classical computing could be easily carried out using *qubits*, a well known example is the simulation of a quantum system, that would take longer than the age of the universe on our current microchips, but would be easily achievable with a quantum computer.
Building a quantum computer, however, is no easy feat, since quantum systems are incredibly delicate and susceptible to external conditions. This means that if we want to build a usable quantum computer we need to be able to cope with errors and unpredicted behaviour. This is why the field of *quantum error correction* was born: if we can tolerate a certain kind and number of errors, we might be able to build a machine can tolerate a misbehaving system.
We need to find a way to apply some form of error detection and correction to a quantum system that cannot be directly measured without being altered.

2. Theory

Classical Error Correction

A classical bit is either in a state 0 or in a state 1. During a computation, it can only be corrupted by one type of error: a bit flip, this will turn a 0 into a 1 and vice versa. A simple way to identify whether an error has occurred is to increase the number of qubits, encoding the information redundantly. For example, we could copy one bit in the 0 state to three bits in the same state. We then use these three physical bits as one logical bits, running them through the computation, and finally we observe the resulting state, taking a vote of majority to see which bit has been flipped.

Quantum Error Correction

Unfortunately, however, the same strategy cannot be employed with a qubit, due to some constraints caused by the strange world of quantum mechanics. A classical repetition code relies on the fact that information can be faithfully copied, but the *no-cloning theorem* [3] forbids this kind of operation. Furthermore, since a qubit is a superposition of two states, it can be subject to a continuum of errors, making it look like we would need a way to measure states with infinite precision. Lastly, direct measurement of a quantum state causes its wavefunction to collapse to a deterministic state, effectively destroying the superposition and rendering it no more useful than a classical bit.
This is where the field of quantum error correction comes to the rescue, introducing ways to indirectly measure and correct arbitrary errors, but to understand how this is possible, it is necessary to first introduce some useful mathematical definitions [3].
A single qubit is usually represented in a superposition of the two states of the *computational basis*:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle\,, \qquad (1)$$

and a system of multiple qubits is represented by the tensor product of single qubit states:

$$|q_1 q_2 \dots q_n\rangle = |q_1\rangle \otimes |q_2\rangle \otimes \dots \otimes |q_n\rangle\,, \qquad (2)$$

Where the notation on the left is shorthand for the notation on the right.

The operators that act on qubits are unitary, and they are represented by the Pauli operators X,Y,Z. Conventionally, the computational basis is defined as the eigenvectors of the Z operator, meaning that this will represent a phase flip ($Z|0\rangle = |0\rangle$ and $Z|1\rangle = -|1\rangle$) and the X operator will represent a bit flip ($X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$). The eigenstates of the X operator are defined as the *Hadamard basis*,

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad \text{and} \quad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \qquad (3)$$

In this basis the X operator acts as a phase flip and the Z operator acts as a bit flip.
To transform between the computational basis and the Hadamard basis we use a *Hadamard gate*, represented by the matrix

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \qquad (4)$$

Applying a Hadamard gate to a $|0\rangle$ or $|1\rangle$ state will turn it into a $|+\rangle$ or $|-\rangle$ state.
Another gate that will help us build quantum error correction circuits (QECC) is the CNOT gate, the quantum equivalent of the XOR gate in classical computations. The CNOT gate acts on two qubits, and, in the computational basis, flips the second one if the first one is in the state $|1\rangle$. It is formally represented by the matrix

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \qquad (5)$$

We mentioned earlier that because qubits are a superposition of states, they should be subject to a continuum of errors. The most remarkable feature of QEC, and what makes it possible in the first place, is that these errors can be digitised into a finite subset, effectively projecting the corrupted state in one error space. These error spaces are also orthogonal to each other and to the uncorrupted state, meaning that they can be measured without altering the original qubit. Hence we can identify what error has occurred into the system and correct it with the opportune operator.
This really is the most important result of quantum error correction and what could allow us to build quantum computers.
Since the Y operator is defined as Y = XZ, if we have a code that can successfully identify and correct X and Z errors, we have a code that can correct any arbitrary error on one qubit.

The 3-qubit code

To introduce the concept of quantum error correction it is useful to consider a simple example of a QECC that does not represent a realistic system, but illustrates nicely how to perform the operations necessary to correct a single error on a quantum state.
The 3-qubit code encodes one qubit into three [4], expanding the Hilbert space of the system to allow it to be projected to measurable error subspaces. One physical qubit is entangled to two other qubits though two CNOT gates as illustrated in Figure 1 [5], forming the equivalent of the initial state in a classical repetition code. It is important to notice, however, that

$$|\psi\rangle_L = \alpha|000\rangle + \beta|111\rangle \neq |\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle, \qquad (6)$$

hence we are not violating the *no-cloning theorem*.
Before the encoding, the qubit lived in a 2-dimensional Hilbert space, but after the encoding it is parametrised by an 8 dimensional Hilbert space. This 8 dimensional space contains 4 orthogonal subspaces corresponding to each bit flip error (or no error). When the encoded qubit is acted upon by the X operator, it is also projected to one of these subspaces. This projection is what allows us to identify the error on the encoded qubit without directly measuring its state: because the spaces are orthogonal, measuring what space the qubit is in will not alter its behaviour.
Once we know what error space the qubit is in (and hence what physical qubit has been flipped), we can apply the same X operator to that qubit and correct the error.
Because this is a very simple example. It is important to remember that the 3-qubit code is not a complete code, since it cannot detect phase flips, but for this case we will only consider this kind of error.
How do we measure what subspace the qubit is in? We need to introduce the concept of *syndrome measurement*. After the system has undergone a probabilistic error, we initialise two *auxilla qubits* and entangle the to the first and second, and second and third qubits respectively, using CNOT gates (Figure 1). The auxillas are initialised in the $|0\rangle$ state and are utilised to carry out the syndrome measurement, effectively acting as ZZI and IZZ operators on the encoded 3-qubit state. A bit flip on a single qubit is a +1 eigenstate of the Z operator, so if an error occurs in one of the first two qubits the measured auxilla will return a value of 1, otherwise it will return a value of 0. Similarly for the second auxilla, if an error occurs in the second or third qubit, we will measure a +1 eigenvalue. Therefore by measuring both auxillas we can determine on which single qubit the error has occurred (as seen in Table I): this is the equivalent of taking a vote of majority in the classical repetition code.

Auxilla Measurement	Projected State	Error
00	$\alpha\|000\rangle + \beta\|111\rangle$	No error
01	$\alpha\|001\rangle + \beta\|110\rangle$	X on qubit 3
10	$\alpha\|100\rangle + \beta\|011\rangle$	X on qubit 1
11	$\alpha\|010\rangle + \beta\|101\rangle$	X on qubit 2

Table I: the 3-qubit code can successfully identify one arbitrary bit flip on any physical qubit.

It must be noted, however, that the 3-qubit code, even when only bit errors are allowed, is not infallible and can be fooled if more than one error occurs: since it is doing the analogue of a vote of majority, if two qubits are flipped, the code will think the other qubit is in the wrong state and it will act on it, effectively corrupting the system more instead of correcting it.
A measure of how well a code is at correcting errors is given by the *fidelity*, defined as:

$$F = \sqrt{|\langle\psi|U|\psi\rangle|}, \qquad (7)$$

where U is a unitary operator, such as the bit flip error X or phase flip error Z.
This quantity will help us compare the behaviour of an unencoded qubit to that of a qubit undergoing error correction, and determine when error correction is advantageous.
The fidelity of a single qubit subject to bit flip errors only can be easily calculated: if the error probability is p, then the qubit will be uncorrupted with probability $(1-p)$ and its fidelity will be $\sqrt{1-p}$, since different error states are orthogonal.
Similarly, for the 3-qubit code, no qubit is flipped with probability $(1-p)^3$, one qubit is flipped with probability $3p(1-p)^2$, two qubits with probability $3p^2(1-p)$ and three qubits with probability p^3. Therefore the fidelity for the encoded qubit will be $\sqrt{1-3p^2+2p^3}$, accounting for the fact that the code can correct one bit flip.

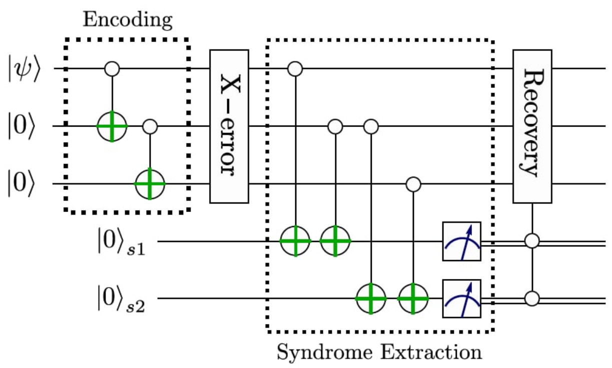

Fig. 1: QECC for 3-qubit code. The encoding part entangles two qubits to our initial state through two CNOT gates. The encoded state is then subject to a random X error. The syndrome extraction is done measuring the auxilla qubits *s1* and *s2*, after they have been entangled to the system. The appropriate correction is then applied in the recovery stage.

Another variant of the 3-qubit code is the *sign flip code*, that works in the same way but with a Hadamard transformation between the error and the syndrome extraction. This way we are able to identify phase flips (as they become bit flips in the Hadamard basis), but we are no longer able to detect bit flips.

The 9-qubit code

Now that we understand how error correction works for qubits, we can build a full QECC, able to correct for an arbitrary error on a qubit. We recall that an arbitrary error is a linear combination of X, Z and Y Pauli operators, and that Y=XZ, therefore, if we can correct for X and Z errors (bit and phase flip), we can correct any error by projecting the quantum state in either an X or Z eigenfunction.
The 9 qubit code [6] is a full QECC and it is able to detect and correct an arbitrary error on any one of the nine physical qubits the logical qubit is encoded in. To achieve this the code uses three 3-qubit codes preceded by a Hadamard gate, encoding the initial states into:

$$|0\rangle_L = \frac{1}{\sqrt{8}}(|000\rangle + |111\rangle)\otimes(|000\rangle + |111\rangle)\otimes(|000\rangle + |111\rangle)\,, \quad (8)$$

$$|1\rangle_L = \frac{1}{\sqrt{8}}(|000\rangle - |111\rangle)\otimes(|000\rangle - |111\rangle)\otimes(|000\rangle - |111\rangle) \quad (9)$$

through the circuit shown in Figure 2 [4].
Error correction is performed in each of the three blocks in the same way it was performed for the 3-qubit code to detect a maximum of 3 bit flip errors (provided they happen in each of the three different blocks). A syndrome extraction for the phase flip is then carried out, comparing the signs of block one and two, and two and three. The auxilla qubits used for this measurements are equivalent to the XXXXXXIII and IIIXXXXXX operators, returning an eigenvalue 1 if they detect an error and 0 if the state is uncorrupted. Since a phase flip has the same effect regardless of which qubit in a 3-qubit block it acts on, we only need to know in which block the phase flip has occurred, and not on which qubit. Error correction is then carried out in the usual way, applying the necessary operators to restore the original state. Much like the 3-qubit code, the 9-qubit code can be fooled, and if too many errors occur, the circuit will have a net negative effect on the system.

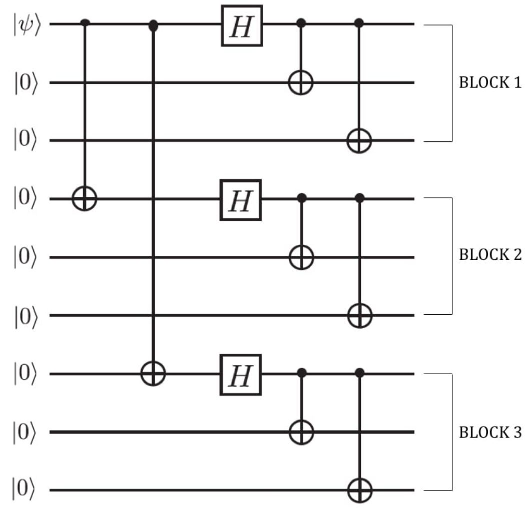

Fig. 2: encoding of a one logical qubit into 9 physical qubits in the 9-qubit code. Each block is in the same way as the 3-qubit code.

The 4-qubit code

Error correcting codes can be quite resource intensive, as it might have been noted in the 9-qubit code: we might want to sacrifice some accuracy for a more efficient circuit. An error detection code, such as the 4-qubit code, is able to identify whether an error has happened but not where it has happened, therefore it is not able to correct for it. The 4-qubit code [7] encodes two logical qubits into four physical ones (in a much more efficient way than the 9-qubit code), and is able to detect an arbitrary error on one of them. The states are encoded as follows:

$$|00\rangle_L = \frac{1}{\sqrt{2}}(|0000\rangle + |1111\rangle)\,, \quad (10)$$

$$|01\rangle_L = \frac{1}{\sqrt{2}}(|1100\rangle + |0011\rangle)\,, \quad (11)$$

$$|10\rangle_L = \frac{1}{\sqrt{2}}(|1010\rangle + |0101\rangle)\,, \quad (12)$$

$$|11\rangle_L = \frac{1}{\sqrt{2}}(|0110\rangle + |1001\rangle)\,. \quad (13)$$

The fact that it cannot correct these errors, however, means that the errored state must be discarded and the circuit reset and rerun. This is especially useful in state preparation. The syndrome measurement is then carried out on two auxillas represented by the operators ZZZZ and XXXX, yielding an eigenvalue of 1 if a bit or phase flip error is detected, respectively, and an eigenvalue of 0 if the state is uncorrupted.

The Threshold theorem

The design of a quantum circuit is of great importance in the context of error correction: its architecture must be so that if one error occurs it doesn't spread to the whole system, but rather remains confined to ideally one qubit, even after computational gate operations. This takes the name of *fault-tolerance* [4]. If a quantum circuit is designed with this in mind, we can attempt to build a concatenated circuit, one where the output of a certain code is used as the input of the following one. In this way we can hope to achieve quantum computation with an arbitrary degree of precision, given the adequate resources.
To ensure that concatenation will not inflate the error rates, however, we must ensure that the error rates of higher level codes are lower than those of lower level ones.
Consider, for instance, a fault-tolerant circuit where only one arbitrary error can be detected. Its failure rate will be

$$P_1 = cp^2 \,, \tag{14}$$

where c is the upper bound on the number of errors that can occur in the circuit. The second layer of the concatenated circuit will have a failure rate related to the first layer:

$$P_2 = c(P_1)^2 = c^3 p^4 \,. \tag{15}$$

For the code to be resilient to errors P_2 must be smaller than P_1 , since this would allow concatenated error correcting codes to lower the error rates rather than increasing them. This is possible if $cp < 1$ and this inequality defines the threshold. The threshold theorem states that a perfect quantum computer can be approximated by a noisy quantum computer provided that the error rates are low enough, precisely:

$$p < p_{th} = \frac{1}{c} \,, \tag{16}$$

where p_{th} is the threshold probability [4].
In this work we tried to numerically simulate the behaviour of the three quantum circuits previously described, and from their behaviour compared to that of an unencoded qubit, we inferred the threshold probability for each of them.

3. Method

We decided to simulate and calculate the threshold probability of three different QECC: the 3-qubit code, the 9-qubit code, and the 4-qubit code. The first was used as a proof of concept, since it is not a full quantum code, but we have theoretical predictions to compare it to. With the same methodology we used for this simple bit-flip correction code, we tested the 9-qubit code, as it is able to correct for an arbitrary error, and the 4-qubit code, as it represents a different approach, focusing on error detection rather than error correction.
All the simulations were run in *python* with the help of the *numpy* and *random* libraries.

The 3-qubit code

The initial state of the system was randomly chosen between one of the encoded states, either $|000\rangle$ or $|111\rangle$. These were calculated as tensor products of the vector representations of the $|0\rangle$ and $|1\rangle$ states. The error operators were defined as tensor products of X and I operators, since for this code we were only considering bit flips. An error acting on the first qubit, for example, would be represented by the matrix resulting from the tensor product of X, I and I, whereas an error on the second qubit would result from the action of the IXI operator. These errors were randomised according to a probability p, plotted on the horizontal axis.
To detect and correct possible errors we need to entangle the auxilla qubits: in our simulation this is done by the operators representing the auxillas. These are the matrices obtained from the ZZI and IZZ tensor products. To obtain the desired syndrome measurement (1 for a detected error, 0 for none), however, we must define a projector operator that, when dotted with the errored state, will return a binary result. This operator is defined as

$$\Pi = \frac{(I - D)}{2} \,, \tag{17}$$

where I is the identity matrix and D is the matrix representing the auxilla measurement.
Because we are initialising our system in one of the orthogonal encoded states the product

$$\langle \psi | \Pi | \psi \rangle \tag{18}$$

will result in either 1 or 0, depending on whether an error has been detected or not, respectively. The information gathered from this syndrome measurement is then fed into an error correcting function, that applies the apt operator to the encoded qubit, to attempt to recover its original state. As mentioned before, this is not always successful and the QECC will lower the fidelity of the system as the error rates increase.
The average fidelity for a given error rate is then calculated by taking an average of many runs, each returning a fidelity of 0 (if the final state has not been successfully corrected) or 1. This value is plotted over a range of probabilities and the curve we obtained in this way is compared to that of a single unencoded qubit. The unencoded qubit is initialised in either the $|0\rangle$ or the $|1\rangle$ state, it is subject to the same type of errors with the same probabilities, but it is not corrected by the QECC. The value for the *threshold probability* is found where the average fidelity of the encoded qubit intersects that of the unencoded one.

The 9-qubit code

The simulation for the 9-qubit code is run in a very similar way to that of the 3-qubit code: the state is initialised in one of its encoded states, but it is subject to both bit and phase flip error, probabilistically on all of the 9 qubits. The syndrome measurement is run in the same way as for the 3-qubit code, defining matrices to detect the bit flip errors and phase flip errors as the tensor products described in the theory. We were particularly careful to match the dimension of these matrices to that of the encoded states, multiplying by identity matrices where needed. The projector operators Π were also defined in the same way as the previous code, and the average fidelity was calculated in the same fashion.
The comparison between the efficiency of the 9-qubit code and that of an unencoded qubit yielded a value for the threshold probability of this QECC.

The 4-qubit code

Once again, the simulation for the 4-qubit code was carried out in a similar way to the previous ones, operating on the randomly chosen encoded state with error matrices and performing the syndrome measurement with the XXXX and ZZZZ operators. This code, however, does not correct errors, but rather only detects them. This was accounted for in our fidelity calculations: this quantity was averaged over the number of qubits that were not discarded by the code, that is those that were either in the correct state or those who were able to fool the syndrome measurement.
Finally, the performance of the 4-qubit code was compared to that of an unencoded qubit and the threshold probability was inferred.

4. Results and Discussion

The 3-qubit code

We considered the simple case of uniform error probability on all qubits, and plotted the average fidelity of the code against the error rate it was evaluated at. The average fidelity was evaluated over a range of 501 error probabilities, going from 0 to 0.9. The average is taken over 50000 qubits.
For the 3-qubit code we were only considering possible bit flip errors, therefore we compared this code to an unencoded qubit that could only be subject to X errors. This proved to be particularly useful as a proof of concept, since we have theoretical predictions for the behaviour of both of these systems.
Figure 3 illustrates our findings, with our simulated fidelity recreating the predicted results almost perfectly. The simulated fidelity of the single qubit decreases as $\sqrt{1-p}$ and that of the QECC decreases as $\sqrt{1-3p^2+2p^3}$, as shown by superimposed theoretical curves.
This confirms that our code is working in the expected way and we can use it to simulate other codes, employing the same strategy.

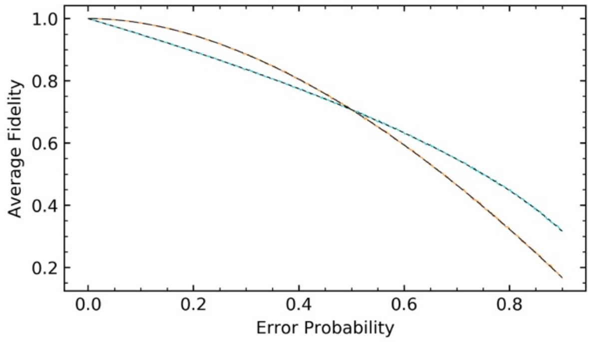

Fig. 3: average fidelity of the 3-qubit code (in orange), compared to an unencoded qubit (in cyan). The dashed and dotted lines are the theoretical predictions for the behaviour of the 3-qubit code and the uncorrected qubit, respectively.

The threshold probability value is found where the two curves intersect: if the average fidelity of the 3-qubit code is worse than that of a single uncorrected qubit concatenation becomes impossible and we cannot build arbitrarily large quantum circuits.
From Figure 3 it is clear that for this QECC the threshold probability $p_{th} = 0.5$, therefore if a bit flip has a higher chance of occurring rather than leaving the system unaltered our code fails to improve the circuit's performance. This result is quite intuitive: the 3-qubit code fails if more than one qubit is flipped, therefore, if the error probability is higher than half, it is more likely that two out of three bits will be corrupted, and the code will not perform as expected.
As mentioned in previous sections, this result very far from being realistic, since not only are we only considering one kind of error, but we are also assuming that this error happens between the encoding stage and the syndrome measurement. This means that we are also assuming that the state is initialised and encoded with no error, and the same is valid for the auxilla qubits and their syndrome measurement.
While fully simulating a quantum circuit, considering errors on state preparation and incorrectly applied gates, is beyond the scope of this work, we can simulate an arbitrary error and observe how a QECC can deal with it.

The 9-qubit code

We considered uniform error rates for this QECC as well, and we made the further assumption that bit flip and phase flips error rates are the same. Including both X and Z operations on the encoded and unencoded qubits means that we are simulating arbitrary errors. Even though these might not be completely realistic, since we are still assuming uniform probabilities, they are an improvement over the previously described case.
Figure 4 was obtained in a similar way to Figure 3, plotting the average fidelity over a range of error probabilities. Because the 9-qubit code involves much bigger matrices (512 dimensional vectors), however, we had to calculate the average fidelity over 10000 qubits and a range of 51 probabilities, still going from 0 to 0.9.
Figure 4 shows that our 9-qubit code is still able to outperform an unencoded qubit, but only for much lower

error rates: arbitrary errors have a higher chance of corrupting the state, and a 9-qubit code exposes the encoded state to a higher chance of being corrupted, since it encodes one qubit into nine.

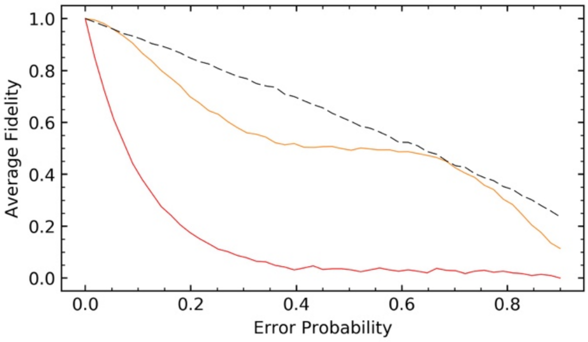

Fig.4: average fidelity of 9-qubit code (in orange) compared to the fidelity of an unencoded single qubit (black dashed line) and that of an encoded but not corrected 9-qubit (in red), all subject to bit and phase flip.

We also compared the fidelity of an encoded and corrected qubit to that of an encoded one that has not undergone syndrome measurement. In this way we were able to show that our 9-qubit code is in fact correcting errors on the qubits it measures.
The threshold is inferred from the point where the fidelity of the error correcting code intersect that of the single qubit. From Figure 4, we see that $p_{th} = 0.06$.
This value is much smaller than that of the 3-qubit code, but still higher than more realistic values found in the literature, which take into account more realistic error probabilities and the possibility of gate errors. For a 7 qubit code thresholds of $\sim 10^{-6}$ have been found [8], and detailed architectural designs like surface codes [9] show thresholds of $\sim 10^{-3} - \sim 10^{-2}$, making them promising for large scale quantum computing.

The 4-qubit code

Lastly we focused on the 4-qubit error detection circuit, making the same assumptions as we did for the 9-qubit code: accounting for any arbitrary errors and setting the error probability for bit and phase flips equal. Once again we did not consider gate errors and we assumed that the encoded state was perfectly initialised in one of its four state.
The fidelity of the QECC was compared to that of an unencoded qubit subject to the same arbitrary errors, over a range of 51 error probabilities, going from 0 to 0.9. Because the vectors and matrices for this code were smaller (16 dimensions) we were able to average the fidelity over a higher number of qubits, 50000 in this case. Figure 5 shows our findings, with the 4-qubit code beating the single qubit for error probabilities smaller than ~ 0.1. Hence the threshold was inferred to be $p_{th} = 0.1$.
This proves that an error detecting code can be effective and useful in the construction of a quantum computer, especially in specific cases such as state preparation. It must be also noted that, as mentioned before, an error correcting code is much less resource intensive than an error correcting one, being able to encode two logical qubits into four physical ones.

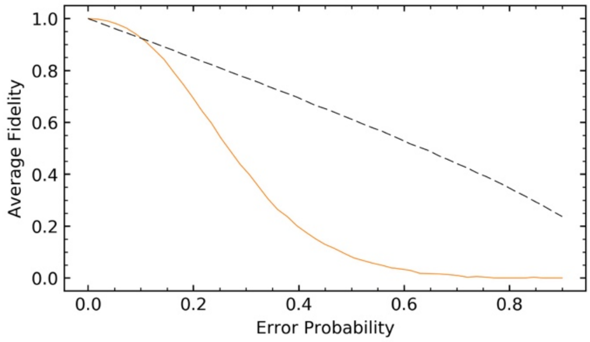

Fig. 5: average fidelity of 4-qubit error detection code (in orange) compared to that of an unencoded qubit (black dashed line), both subject to bit and phase flips.

5. Conclusions

In our work we showed that it is possible to carry out error correction on a quantum bit, and that arbitrary errors can be reduced to a set of known ones that can be acted upon with the appropriate operator to recover the initial state. We also showed that if the error rates of a system stay under a certain threshold, we can in theory build arbitrarily large quantum computers. These threshold probabilities were calculated for three different error correction and detection circuits, under some simplifying assumptions, by comparing the behaviours of these circuits to that of an unencoded qubit. Our simulations proved that applying error correction or error detection improves the fidelity of the quantum system below the threshold probability.

References

[1] Moore, Gordon E. (1965). "Cramming more components onto integrated circuits". Electronics Magazine.

[2] Jeremy L. O'Brien, Akira Furusawa, and Jelena Vuckovic. Photonic quantum technologies. Nature Photonics, 3(12):687{695, 2010.

[3] Michael a. Nielsen and Isaac L. Chuang. Quantum Computation and Quantum Information: 10th Anniversary Edition . 2011.

[4] Simon J Devitt, William J Munro, and Kae Nemoto. Quantum Error Correction for Beginners. Arxiv, 076001:1{41, 2013.

[5] Joschka Roffe, private communications.

[6] .P.W. Shor. Scheme for reducing decoherence in quantum computer memory. Phys. Rev. A., 52:R2493, 1995.

[7] M. Grassl, Th. Beth, and T. Pellizzari. Codes for the quantum erasure channel. Phys. Rev. A., 56:33, 1997.

[8] D. Gottesman. PhD Thesis (Caltech). quant-ph/9705052, 1997.

[9] D.S. Wang, A.G. Fowler, A.M. Stephens, and L.C.L. Hollenberg. Threshold Error rates for the toric and surface codes. Quant. Inf. Comp., 10:456, 2010.