The threshold probability p_{th} is fundamental due to the following theorem.

Theorem 7.1 (Threshold theorem). *If a threshold probability p_{th} exists, then it is always possible to correct errors at a faster than they are created. It is sufficient to increase the level k of encoding.*

Proof. The proof is trivial. As long as the the occurrence of an error on the physical qubit $p = \epsilon$ is smaller than the threshold probability p_{th}, then the ratio

$$\frac{p}{p_{th}} < 1, \tag{7.49}$$

and thus the quantity

$$\left(\frac{p}{p_{th}}\right)^{2^k}, \tag{7.50}$$

can be made suitably small by simply increasing k.

The beauty of the threshold theorem is its simplicity. However, it also highlights a non-trivial problem, which is the necessity of employing a very large number of physical qubits. It naturally follows the question: How many physical qubits are necessary to quantum error correcting a faulty circuit?

Suppose we have N components (this is the number of qubits times the number of gates). Suppose for each of these components one needs R physical qubit accounting for QEC at the 1 level of encoding. Then, after k levels of encoding there is a total of NR^k qubits. Suppose we want that the entire circuit works with a failing probability $P_{\text{fail, circuit}} < \epsilon$, where ϵ is a given probability. Then, per component, we have

$$P_{\text{fail}} = p_{\text{th}} \left(\frac{p}{p_{\text{th}}}\right)^{2^k} < \frac{\epsilon}{N}. \tag{7.51}$$

The question is then how many levels k of encoding are necessary? Or equivalently, how many physical qubits R^k per components are required? From the previous expression we obtain

$$2^k \sim \frac{\log_2\left(\frac{Np_{\text{th}}}{\epsilon}\right)}{\log_2\left(\frac{p_{\text{th}}}{p}\right)}, \tag{7.52}$$

which implies

$$R^k \sim \left(\frac{\log_2\left(\frac{Np_{\text{th}}}{\epsilon}\right)}{\log_2\left(\frac{p_{\text{th}}}{p}\right)}\right)^{\log_2 R}. \tag{7.53}$$

Thus, the size of the full circuit scales as

$$NR^k \sim \text{poly}\left(\log \frac{Np_{\text{th}}}{\epsilon}\right). \tag{7.54}$$

This is the quantitative result of the threshold theorem.

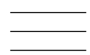
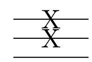
7.1.7 More layers of encoding or only more qubits

Until now, we worked under the assumption of having only noises that act independently on the physical qubits. Let us suppose now a different kind of noise map, that correlates the noise on different qubits. Specifically, we consider a Kraus map acting on N qubits, that reads

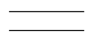
$$\mathcal{T}[\hat{\rho}] = \frac{1}{N(N-1)} \sum_{i \neq j}^{N} \left[(1-p)\hat{\rho} + p\hat{\sigma}_x^{(i)}\hat{\sigma}_x^{(j)}\hat{\rho}\hat{\sigma}_x^{(j)}\hat{\sigma}_x^{(i)} \right], \tag{7.55}$$

where there is a probability p that the noise acts on the qubits and the prefactor is required to properly normalise the state.

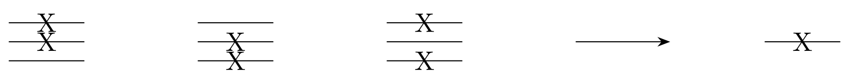
First of all, let us introduce a graphical notation. Suppose we have three physical qubits, and the noise has acted on the first and second. Then, we denote this with the following scheme

Before the noise  After the noise  (7.56)

Now, suppose we are performing the encoding with $N = 3$ qubits and $k = 1$ layer of encoding. If there are no errors — this happens with probability $(1 - p)$ — then physical qubits are

 and correspond to a logical qubit (7.57)

which is untouched by the noise. If there is one error, which corresponds to two qubits being affected with a probability p, then the physical qubits are in one of the following three states



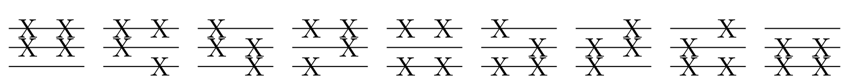
(7.58)

that correspond to a logical qubit that is affected to the noise. Here, we denote decodings with horizontal arrows. Thus, the protocol fails. These states contribute to the failing probability:

$$P_{\text{fail}} \sim 3p, \tag{7.59}$$

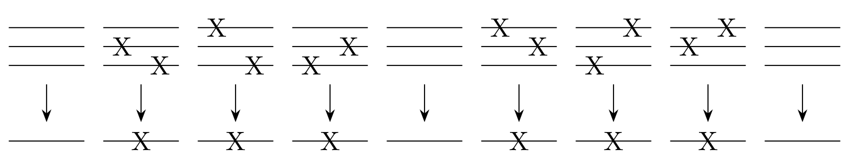
where the factor 3 is given by the number of equivalent states at the physical level, namely those represented graphically in the left side of Eq. (7.58).

If there are two errors, which is a process involving a probability p^2 and four qubits, one has the following 9 states:



(7.60)

which correspond to



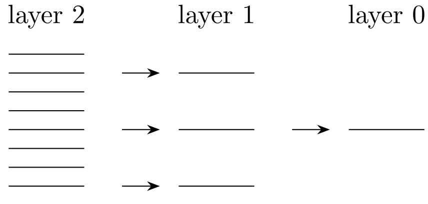
(7.61)

as the application of an bit-flip error twice on the same qubit correspond to not having an error, i.e. $\hat{\sigma}_x^2 = \hat{\mathbb{1}}$. In such a case, only some combination are faulty, while in others the errors have cancelled. Here, we denote equivalence with vertical arrows. These states will contribute to P_{fail} with $+6p^2$. Thus, one gets

$$P_{\text{fail}} = 3p + 6p^2 + \cdots \sim 3p, \tag{7.62}$$

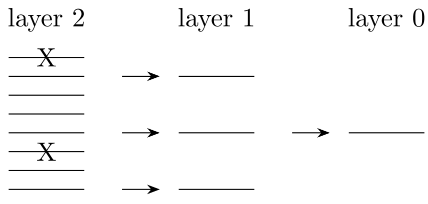
where the ... indicate higher order errors. Nevertheless, is the lowest order term in p that is the most significant, under the hypothesis of small error probabilities.

Consider now a double encoding $k = 2$ with a total of $N = 3^2 = 9$ qubits. In case of no errors, prob $= (1-p)$, we have



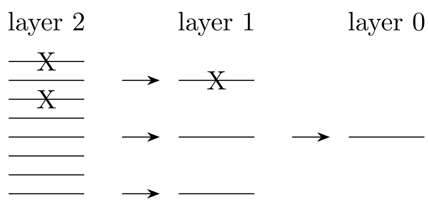
(7.63)

which do not contribute to P_{fail}. If there is one error, prob $= p$, we have $\binom{9}{2} = 36$ states. Some states display two affected physical qubits in different layer-1 logical qubit,



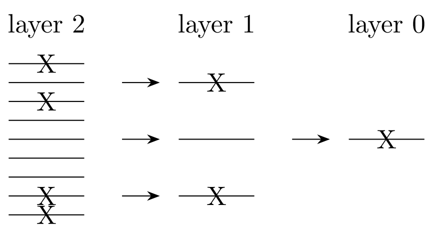
(7.64)

Some have two physical qubits in the same layer-1 logical qubit. Thus, the corresponding layer-1 logical qubit fails, but the layer-0 logical qubit is still protected



(7.65)

This is the worst-case scenario with one error. If there are two errors, prob $= p^2$, the states that are relevant are those that have 2 layer-1 logical qubits affected. Namely, they reproduce the same graph as that in Eq. (7.58). For example, of the form



(7.66)

At the layer-1, this correspond to a probability being $3p$. For each of the layer-1 affected logical qubits, one needs 2 layer-2 affected physical qubits, with associated a probability $3p$. Thus, one has

$$P_{\text{fail}} = (3p)^2 + \cdots \sim 9p^2. \tag{7.67}$$

With generic k layers of encoding, one has

$$P_{\text{fail}} = (3p)^k + \dots, \tag{7.68}$$

which is graphically represented in Fig. 7.5, where the lines are listed at increasing values of k and condense towards the value of $p_{\text{th}} = 1/3$.

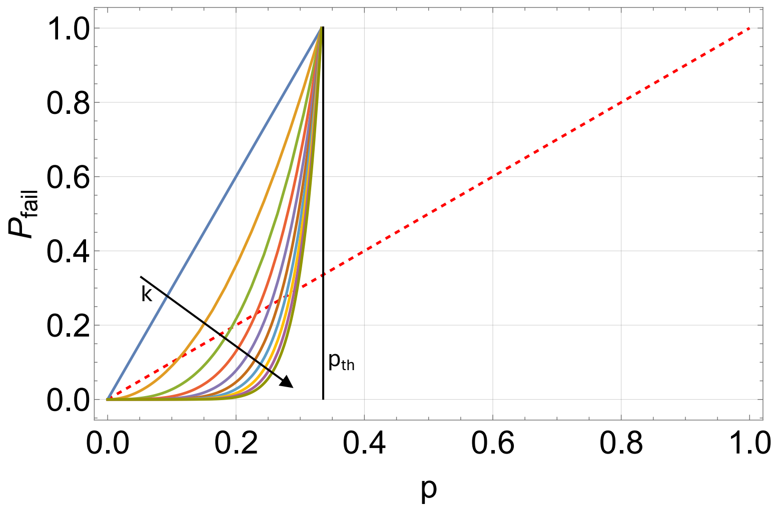


Fig. 7.5: Comparison of the failing probabilities P_{fail} for different k layers of encoding with 3 qubits each. The arrow indicates the direction of increasing values of k, while the vertical black line indicates the threshold probability p_{th}.

Let us now consider the alternative. Instead of taking a large number of layers of encoding with just a few qubits per layer, we consider a large number of qubits on a single encoding layer.

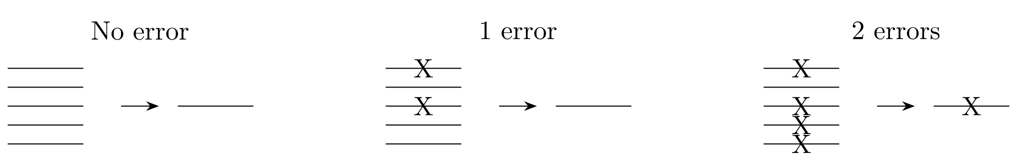
For $N = 3$, a single error, i.e. having two physical qubits affected, is sufficient to make the encoding fail:

(7.69)

where there are three different combinations (see Eq. (7.58)) that count. Thus, we have

$$P_{\text{fail}} \sim 3p \tag{7.70}$$

For $N = 5$, one requires two errors, with four qubits affected, to make the encoding fail. Indeed,



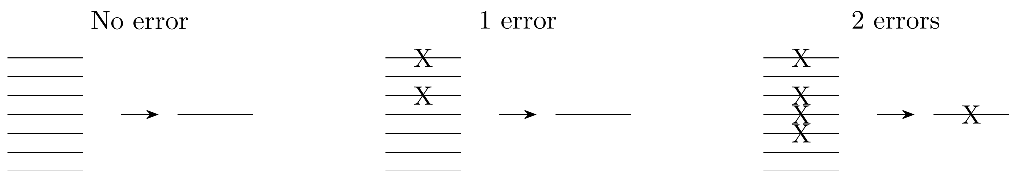
(7.71)

In such a case, the failing probability is given by

$$P_{\text{fail}} = \frac{1}{2}\binom{5}{2}\binom{3}{2}p^2 + \cdots \sim 15p^2, \tag{7.72}$$

where the first binomial chooses 2 qubits to affect among the available 5, the second binomial chooses 2 qubits among the remaining 3. The factor one-half acconts for the simmetry between the first error and the second one, i.e. between the first couple of affected qubits and the second one.

Also for $N = 7$, one requires 2 errors, i.e. four affected qubits. Indeed



(7.73)

In such a case, we get

$$P_{\text{fail}} = \frac{1}{2}\binom{7}{2}\binom{5}{2}p^2 + \cdots \sim 105p^2. \tag{7.74}$$

Figure 7.6 compares failing probabilities of the encodings with different values of N. As one can see the knee of the curves moves towards zero, meaning that the threshold does not exist.

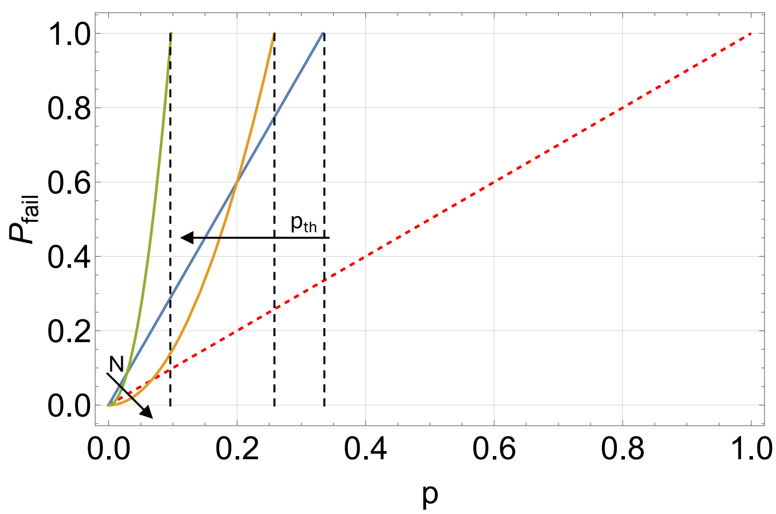


Fig. 7.6: Comparison of the failing probabilities P_{fail} for different values N of the qubits with a single layer of encoding. The arrow indicates the direction of increasing values of N, while the vertical black dashed lines indicate how the curves cannot define a threshold probability.

Thus, for the error defined in Eq. (7.55), employing several layers of encoding can allow for a fault-tolerant quantum computing, while employing only a single layer encoding with more qubits has strong limits.