

Coherent errors and readout errors in the surface code

Á. Márton¹ and J. K. Asbóth^{1,2}

¹Department of Theoretical Physics, Institute of Physics, Budapest University of Technology and Economics, Műegyetem rkp. 3., H-1111 Budapest, Hungary

²Wigner Research Centre for Physics, H-1525 Budapest, P.O. Box 49., Hungary

We consider the combined effect of readout errors and coherent errors, i.e., deterministic phase rotations, on the surface code. We use a recently developed numerical approach, via a mapping of the physical qubits to Majorana fermions. We show how to use this approach in the presence of readout errors, treated on the phenomenological level: perfect projective measurements with potentially incorrectly recorded outcomes, and multiple repeated measurement rounds. We find a threshold for this combination of errors, with an error rate close to the threshold of the corresponding incoherent error channel (random Pauli-Z and readout errors). The value of the threshold error rate, using the worst case fidelity as the measure of logical errors, is 2.6%. Below the threshold, scaling up the code leads to the rapid loss of coherence in the logical-level errors, but error rates that are greater than those of the corresponding incoherent error channel. We also vary the coherent and readout error rates independently, and find that the surface code is more sensitive to coherent errors than to readout errors. Our work extends the recent results on coherent errors with perfect readout to the experimentally more realistic situation where readout errors also occur.

1 Introduction

The surface code[1, 2] is one of the most promising candidates for quantum error correction. For a code patch of distance d , the collective quantum state of d^2 physical qubits is used to store a single logical qubit. Incoherent errors (from entanglement of the physical qubits with a memoryless environment) can be modeled as random Pauli

operators on the physical qubits. Repeated measurements of the parity check operators (a.k.a. stabilizer generators, to obtain the so-called syndrome) can be used to localize and correct such errors, with a success probability that increases as the code distance increases, $d \rightarrow \infty$, as long as the error rates are below the so-called threshold.

For incoherent errors, the value of the threshold depends on the details of the error model, of the error correction (decoding) procedure, and on the level of detail in which the measurements are modeled (circuit-level or not). For simple cases, the threshold is known from mappings between the correction of incoherent errors on the surface code and phase transitions in classical Ising models[1, 3]: it is around 10% with perfect and around 3% with imperfect measurements. This mapping can be extended to some other error models and codes as well[4, 5]. For more complicated error models and circuit-level modeling the threshold can be obtained numerically, using efficient simulation in the Heisenberg picture[6, 7], and is around 0.75%. These threshold values not very far from current experimental reality: below 1% for quantum gates, and a few % for readout[8, 9].

The effect of coherent errors on the surface code is less well understood. These are errors modeled by nonrandom unitary operators acting on each physical qubit separately at each timestep. In the simplest case – the one we will also consider – these are phase rotations of the qubits with a fixed angle θ , i.e., $e^{i\theta\hat{Z}}$. Such errors model the effect of components becoming miscalibrated, inevitable in long calculations. Since coherent errors are not Clifford operations, their effects cannot be simulated efficiently in the Heisenberg picture[6]. Brute-force simulations[10, 11], tensor network methods[12] or other efficient approximations[13], and the standard mappings to statistical physics mod-

els also break down (but note a recent extension of the mappings to a Majorana scattering network[14]).

One of the ways in which coherent errors are more complicated than incoherent errors is that they lead to coherent errors on the logical level as well. With coherent errors, the parity check measurements project the surface code into a random state, which even after error correction differs from the original state. For codes with an odd distance, this difference corresponds to a coherent rotation on the logical level. This logical-level coherence also makes the quantitative investigation of a quantum memory more complicated. Note, however, that scaling up the surface code leads to a "washing out" of coherence from the errors on the logical level. As shown analytically[15, 16], the random logical-level rotations correspond more and more to random Pauli noise as the code size is increasing. Note also, however, that the rate of this logical-level Pauli noise has not been computed analytically, and it can be considerably higher than what one would get by simply replacing coherent errors with their Pauli twirled counterparts[17, 11].

Coherent errors also seem to have a threshold, as shown numerically by Bravyi et al[18]. They have simulated relatively large code sizes using a mapping to Majorana fermions. They have seen the "washing out of the coherence", and found that the rate of the logical-level Z error is significantly higher than that obtained by Pauli twirling the physical-level errors. Nevertheless, their numerics revealed an error threshold for coherent errors too: for $\theta < \theta_{th} \approx 0.08\pi - 0.1\pi$, the logical-level error rates decrease as the code is scaled up. This value of the threshold is very close to that of the Pauli twirled physical error channel: $\sin(\theta_{th})^2 \approx 0.09$.

In this paper we bring the results on coherent errors closer to experimentally relevant setting by considering them together with measurement errors. We consider the simplest kind of measurement errors, the so-called phenomenological error model: perfect projective von Neumann measurements of the parity check operators, with possibly incorrectly recorded measurement results. We use the numerical simulation approach based on the mapping to Majorana fermions[18, 19], but combine this with readout errors, and a corresponding 3D decoding.

This paper is structured as follows. In Sec. 2 we briefly introduce the most important concepts of the surface code, including error correction by minimum weight perfect matching on a 3D syndrome graph. In Sec. 3 we introduce the key points of the simulation method using fermionic linear optics, as pioneered by Bravyi et al[18]. In Sec. 4 we present our results on the combined effects of coherent and readout errors on the surface code. In Sec. 5 we conclude the paper with a discussion of our results.

2 Surface code

We briefly introduce the surface code as a quantum memory, storing a single logical qubit, in the rotated basis[10], with the patch encoding[20].

2.1 Definition of the code space

A distance- d patch of the surface code (we always take d odd) consists of $n = d^2$ physical (data) qubits arranged in a square grid. An example with $d = 5$ is shown in Fig. 1. The faces of the grid are colored in a checkerboard pattern, light (brown) and dark (blue). Extra boundary faces are also included, on all the edges, to ensure that top and bottom edges consist of only blue faces, while left and right edges of only light faces (smooth/rough boundaries[2]).

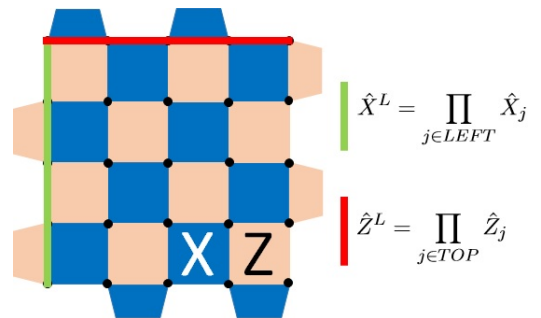


Figure 1: A patch of a surface code with code distance $d = 5$.

All faces of the grid correspond to parity check operators, which are the stabilizer generators of the code. These are products of Pauli operators on the qubits at the corners of the corresponding face. For each light (dark) face, \hat{Z} (\hat{X}) operators

are used, i.e.,

$$\hat{A}_f = \prod_{j \in \partial f} \hat{X}_j; \quad \hat{B}_f = \prod_{j \in \partial f} \hat{Z}_j; \quad (1)$$

$$\hat{S}_f = \hat{A}_f \text{ if } f \text{ dark}; \quad \hat{S}_f = \hat{B}_f \text{ if } f \text{ light.} \quad (2)$$

They have eigenvalues ± 1 , which correspond to even/odd parity of the corresponding group of qubits (in X or Z basis). Since any dark face shares either no corners or two corners with any light face, all of these stabilizer operators commute.

The code space ("quiescent state"[2]) of the surface code is defined as the $+1$ eigenspace of all the stabilizers. This is a two-dimensional space spanned by the logical basis states,

$$|0_L\rangle = N_d \prod_{f \in \text{blue}} \frac{1}{2} (1 + \hat{A}_f) |0\rangle^{\otimes n}; \quad (3)$$

$$|1_L\rangle = N_d \prod_{f \in \text{blue}} \frac{1}{2} (1 + \hat{A}_f) |1\rangle^{\otimes n}, \quad (4)$$

where $|0\rangle^{\otimes n}$ and $|1\rangle^{\otimes n}$ are the states where all physical qubits are $|0\rangle$ and $|1\rangle$ respectively, and $N_d = 2^{(d^2-1)/4}$ is a normalizing factor. Both logical basis states are highly entangled states of the physical qubits, and are locally indistinguishable from each other (all physical qubits have completely mixed reduced density matrices). The encoding of a logical qubit is a mapping from a Hilbert space of dimension 2 to one of dimension 2^n ,

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \rightarrow |\psi_L\rangle = \alpha |0_L\rangle + \beta |1_L\rangle. \quad (5)$$

The logical (or "encoded") operators \hat{X}^L and \hat{Z}^L must fulfil $\hat{X}^L |0_L\rangle = |1_L\rangle$, $\hat{X}^L |1_L\rangle = |0_L\rangle$, $\hat{Z}^L |0_L\rangle = |0_L\rangle$, $\hat{Z}^L |1_L\rangle = -|1_L\rangle$. One possible choice of such logical operators is products of \hat{X} , (\hat{Z}) on qubits along the left (top) edge,

$$\hat{X}^L = \prod_{j \in \text{LEFT}} \hat{X}_j; \quad \hat{Z}^L = \prod_{j \in \text{TOP}} \hat{Z}_j, \quad (6)$$

as shown in Fig. 1.

2.2 Coherent errors and their detection

Of the various error processes by which the environment can affect the states of the physical qubits, we focus on coherent errors[18]. Here each physical qubit undergoes a fixed $SU(2)$ unitary

operation in every timestep, resulting from e.g. calibration errors in a quantum computer. For simplicity, we take this unitary to be the same for every qubit, specifically, a rotation about the Z axis through an angle θ (noise parameter)[18]. Thus the unitary operator representing the effect of noise reads,

$$\hat{U} = \prod_{j=1}^n e^{i\theta \hat{Z}_j}. \quad (7)$$

The noise parameter θ can be converted to a *physical error rate* p , as

$$p = \sin^2(\theta). \quad (8)$$

This is the parameter of the dephasing channel obtained by Pauli twirling the coherent errors.

To detect and correct the coherent errors, we use the standard procedure of repeated measurements of the parity check operators. Each measurement results in a measurement outcome and a post-measurement quantum state. The string s of measurement outcomes is the *syndrome*,

$$\text{syndrome } s = (s_1, \dots, s_{n-1}), \quad (9)$$

with all elements $s_f \in \{+1, -1\}$. The (unnormalized) post-measurement state of the code can be obtained by a projection of the pre-measurement state,

$$\hat{\Pi}_s = \prod_{\forall f} \frac{1}{2} (1 + s_f \hat{S}_f); \quad |\Phi_s\rangle = \hat{\Pi}_s |\Phi\rangle. \quad (10)$$

Note that since we consider only coherent errors that are Z -rotations, only the X -parity check measurements can return with a -1 value.

2.3 Error correction with perfect readout

Before we discuss readout errors, we need to briefly summarize how the errors would be corrected if readout was perfect.

If some of the parity check measurements have resulted in an outcome of -1 , a *correction operation* \hat{C}_s is needed to bring the state back into the code space. Since the coherent errors only contain \hat{Z} , this correction involves flipping some qubits in the X basis by \hat{Z} , i.e.,

$$\hat{C}_s = \prod_{j \in l} \hat{Z}_j. \quad (11)$$

The set l of qubit indices is in a properly defined sense a 1-chain that connects the error locations with each other or with the left or right edge [1], i.e., whose boundary (in a homological sense) is the set of faces where the measured parity check operators are -1 .

Deciding on the correction operator for a given syndrome is the decoding problem. There are many possible correction operators for any given syndrome, which fall into two homological equivalence classes: when multiplying any two correction operators, if they are in the same class, we obtain a product of stabilizers, if they are in different classes, we obtain a logical \hat{Z}^L times a product of stabilizers. In principle, the likelihoods of the two classes (based on the error model) should be compared and any correction operator from the more likely class should be chosen. However, given the computational cost of the likelihood calculation, various approximate approaches, so-called decoders, have been developed [21, 22, 23]. We will discuss this in more detail after the introduction of readout errors.

The collective quantum state of n qubits after we measured syndrome s , and applied the corresponding correction operator, reads

$$|\Phi_s\rangle = \frac{1}{\sqrt{P(s)}} \hat{C}_s \hat{\Pi}_s \hat{U} |\psi_L\rangle. \quad (12)$$

This state is in the logical subspace. Moreover it can be written as a rotation around the logical Z axis [18] with the *logical rotation angle* $\theta_L(s)$,

$$|\Phi_s\rangle = e^{i\theta_L(s)\hat{Z}^L} |\psi_L\rangle. \quad (13)$$

Here both the probability $P(s)$ and the angle $\theta_L(s)$ are independent of the initial state $|\Psi_L\rangle$. These are unique properties of the rotated surface code, and are not necessarily true for surface codes on general lattices [19]. To obtain these properties all the X-stabilizers need to have even weight, while logical Z-operators should have odd weight. If this is not fulfilled, Eq. (12) can not be written as a logical rotation around the Z-axis, furthermore, the syndrome measurement is a weak measurement of the initial logical state $|\Psi_L\rangle$, resulting in an information "leak" [24].

2.4 Quantifying logical errors

To characterize the effectiveness of error correction, we use two quantitative measures, the

diamond-norm distance of a channel to the identity, and the *maximum infidelity* of the resulting state with the initial state [25]. These show how different the state of the encoded qubit is after the error correction process from the original state. In our case the average over syndromes of the diamond-norm distance can be expressed as [18]:

$$p_L^d = 2 \sum_s P(s) |\sin(\theta_L(s))|, \quad (14)$$

and the average over the syndromes of the maximum infidelity as

$$p_L^i = \sum_s P(s) \sin^2(\theta_L(s)). \quad (15)$$

Of these two measures, we prefer the maximum infidelity, since it is a more natural generalization of the logical error rate for coherent errors. However, we have also calculated the diamond-norm distance, and used it for the numerics - please find the corresponding analysis in Appendix A.

2.5 Readout errors, 3D syndrome

We take into account not only coherent errors on the physical qubits, but also readout errors distorting the result of the syndrome measurements. We consider the simplest, *phenomenological* noise model for the readout [26]: perfect syndrome measurements, whose outcome is unreliably recorded, with a readout error probability q , i.e.,

$$P(1 \rightarrow 0) = P(0 \rightarrow 1) = q, \quad (16)$$

and correspondingly, $P(1 \rightarrow 1) = P(0 \rightarrow 0) = 1 - q$. The obtained noisy syndrome is

$$\text{noisy syndrome: } s \rightarrow s'. \quad (17)$$

To solve the decoding problem in the presence of readout errors, we need to consider d consecutive rounds of syndrome measurements [2, 1]. Since errors occur between the rounds of syndrome measurements, the rounds of measurement outcomes differ from each other even if the measurements are perfect. The d rounds of syndromes constitute a *3D syndrome*, which without/with readout errors is

$$\underline{s} = \{s_1, s_2, \dots, s_d\} \rightarrow \underline{s}' = \{s'_1, s'_2, \dots, s'_d\}. \quad (18)$$

For error correction we have to solve the decoding of this 3D syndrome with some decoding technique, obtaining a correction operator $\hat{C}_{\underline{s}'}$. We will detail the decoding in the next Section.

We can express the final state of the code for a measured 3D syndrome \underline{s}' , where the corresponding 3D syndrome without readout error was \underline{s} , as

$$|\Phi_{\underline{s}, \underline{s}'}\rangle = \frac{1}{\sqrt{P(\underline{s})}} \hat{C}_{\underline{s}'} \hat{\Pi}_{s_d} \hat{U} \dots \hat{\Pi}_{s_1} \hat{U} |\psi_L\rangle. \quad (19)$$

Using $\hat{U} = \prod_{j=1}^n \exp(i\theta \hat{Z}_j)$, this can be rewritten (more details in the next section) as

$$|\Phi_{\underline{s}, \underline{s}'}\rangle = \hat{C}_{\underline{s}'} \hat{C}_{\underline{s}} e^{i\theta^*(\underline{s}) \hat{Z}^L} |\psi_L\rangle. \quad (20)$$

Here the rotation angle $\theta^*(\underline{s})$ only depends on the noiseless 3D syndrome \underline{s} , but not on the readout errors, nor on the initial state $|\psi_L\rangle$.

2.6 Error correction with readout errors

To correct errors based on the noisy 3D syndrome contaminated with readout errors we used the 3D version of the minimum weight perfect matching (MWPM) decoder[23], as implemented in PyMatching[27]. In this 3D case, like in the case with perfect measurements, errors are associated with marked vertices on a grid, we need to find the set of edges on the grid with the smallest weight that pair the vertices up or connect them to the right/left boundaries.

The grid here is 3-dimensional, with "space" coordinates ($d \times d$ grid) giving the position of the measured stabilizer operator and "time" coordinates (running from 2 to d) corresponding to the measurement round. Those vertices are marked where the measured stabilizer value differs from that measured in the previous round. "Spacelike" and "timelike" edges on the grid correspond to readout errors and physical errors. These carry different weights (w_s and w_t), since the rate p of coherent errors can differ from the rate q of readout errors,

$$w_s = \log\left(\frac{1-p}{p}\right); \quad w_t = \log\left(\frac{1-q}{q}\right). \quad (21)$$

The MWPM decoder finds the set of edges with the smallest overall weight, which perfectly connect the marked vertices (with each other, or with the left or right boundaries). The set of edges with the minimum weight is used to define the correction operator, which consists of a string

of \hat{Z} operators. Spacelike edges correspond to \hat{Z} operators, however timelike edges have no physical meaning, they are just virtual corrections of readout errors. A technical note: to ensure the correction operator brings the state back to the code space, the last measurement round is assumed to be free of readout errors. This is a common way of eliminating a source of error that is not compounded when the quantum code is used in a fault-tolerant quantum computation.

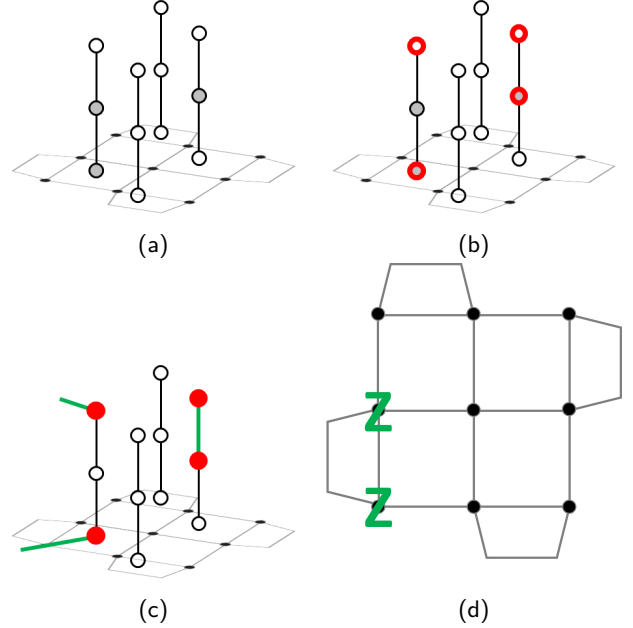


Figure 2: Specific example of MWPM decoding method on a code patch with $d = 3$. Stabilizer measurement outcomes are represented on a 3D grid, ± 1 outcomes as white/grey circles. Vertices, where the measured value differs from the previous round, are marked with red circles. The minimum weight set of edges, which perfectly connects the marked vertices denoted by green color. This set of edges force the correction including 2 \hat{Z} operators.

The final state, after the error correction operator has been applied, cf. Eq. (20), reads

$$|\Phi_{\underline{s}, \underline{s}'}\rangle = e^{i\theta_L(\underline{s}, \underline{s}') \hat{Z}^L} |\psi_L\rangle. \quad (22)$$

The logical rotation angle $\theta_L(\underline{s}, \underline{s}')$ depends on perfect and noisy 3D syndrome too, as

$$\begin{aligned} \theta_L = \theta^* & \leftarrow \hat{C}_{\underline{s}'} \hat{C}_{s_d} |\psi_L\rangle = |\psi_L\rangle; \\ \theta_L = \theta^* + \frac{\pi}{2} & \leftarrow \hat{C}_{\underline{s}'} \hat{C}_{s_d} |\psi_L\rangle = \hat{Z}^L |\psi_L\rangle. \end{aligned} \quad (23)$$

Here the property that the operator $\hat{C}_{\underline{s}'} \hat{C}_{s_d}$ acts like a logical Z-operator or an identity is guaranteed by the constraint of perfect measurements in the last round of error correction.

The average diamond-norm distance and maximum infidelity can be expressed as:

$$P_L^d = 2 \sum_{\underline{s}, \underline{s}'} P(\underline{s}) P(\underline{s} \rightarrow \underline{s}') |\sin(\theta_L(\underline{s}, \underline{s}'))|; \quad (24)$$

$$P_L^i = \sum_{\underline{s}, \underline{s}'} P(\underline{s}) P(\underline{s} \rightarrow \underline{s}') \sin^2(\theta_L(\underline{s}, \underline{s}')). \quad (25)$$

3 Fermionic Linear Optics Simulation

We simulate quantum error correction by sampling the random outcomes of the syndrome measurements, and the final state of the logical qubit after the error correction. This can be summarized in the following steps:

1. Generate a sample of the 3D syndrome \underline{s} - a sequence of d syndrome measurement rounds - from the probability distribution $P(\underline{s})$.
2. Calculate the corresponding rotation angle of the logical qubit, $\theta^*(\underline{s})$.
3. Generate readout errors, i.e., noisy syndrome \underline{s}' from probability distribution $P(\underline{s} \rightarrow \underline{s}')$.
4. Calculate whether the rotation angle of the logical qubit is changed as a result of the readout errors, i.e., whether $\theta_L = \theta^* + \pi/2$ or $\theta_L = \theta^*$, according to Eq. (23)

Steps 3. and 4. here are straightforward, they involve changing measured values of stabilizers with readout error probability q , and decoding using MWPM. Steps 1. and 2., however, can only be computed efficiently using Fermionic Linear Optics tools, and subtle tricks, as recently introduced by Bravyi et al [18]. We only introduce some of the main concepts of the method of Bravyi et al. here, and give a brief summary in an Appendix; see [18, 19] for more details. We describe how the method can be extended to the sampling of repeated syndrome measurements.

3.1 Defining Majoranas for the qubits

To make use of the tools of fermionic linear optics, we introduce a four-dimensional Hilbert space for each qubit, and four Majorana operators (Majoranas) acting in this Hilbert space. The Majoranas for the m -th qubit are denoted by $\hat{c}_j^{(m)}$, with $j = 1, 2, 3, 4$. They are similar to fermionic operators, in that different Majoranas

anticommute; however, all Majoranas square to the identity, $\hat{c}_j^{(m)} \hat{c}_l^{(m')} + \hat{c}_l^{(m')} \hat{c}_j^{(m)} = 2\delta_{jl}\delta_{m'm}$. Using the so-called C4 code [18, 28], the Pauli operators acting on the qubit are represented using Majoranas as

$$\begin{aligned} \hat{X}_m &= i\hat{c}_1^{(m)}\hat{c}_2^{(m)}; & \hat{Z}_m &= i\hat{c}_2^{(m)}\hat{c}_3^{(m)}; \\ \hat{Y}_m &= i\hat{c}_3^{(m)}\hat{c}_1^{(m)}. \end{aligned} \quad (26)$$

These operators fulfil the commutation relations expected of the Pauli operators.

The Majoranas require a Hilbert space that is larger than that of the qubit itself. In fact, since above the Pauli operators were represented by products of two Majoranas, all states of the qubit are represented in so-called fixed-parity subspaces. We work on the subspace defined as +1 eigenspace of the C4 stabilizer,

$$\hat{S}^{(m)} = -\hat{c}_1^{(m)}\hat{c}_2^{(m)}\hat{c}_3^{(m)}\hat{c}_4^{(m)}. \quad (27)$$

The advantage of introducing Majoranas is that initialization of the code, coherent errors, and sampling the measurement statistics of the stabilizers can all be mapped to the time evolution of a noninteracting fermionic system.

The main idea of the fermionic linear optics approach is that we can work with the covariance matrix of the Majorana operators. Therefore instead of simulating the state vector of d^2 qubits, with 2^{d^2} elements, it is enough to keep track of the covariance matrix with $(2d)^4$ elements. With proper transformations of the covariance matrix, corresponding to free fermionic time evolution and measurement of Majorana pairs, we are able to sample $\theta^*(\underline{s})$ from the distribution $P(\underline{s})$ in $\mathcal{O}(d^4)$ time.

We have extended the original simulation method for coherent errors [18], to the case of simultaneous coherent and readout errors. The key observation is that multiple rounds of coherent errors and stabilizer measurements can be decomposed into single rounds of inhomogeneous coherent errors, i.e., where the physical rotation angle θ can be different for each physical qubit.

Starting from Eq. (19), we are able to write it a slightly different way by inserting identities in the form $\hat{C}_{s_j} \hat{C}_{s_j}$,

$$\begin{aligned} |\Phi_{\underline{s}, \underline{s}'}\rangle &= \frac{1}{\sqrt{P(\underline{s})}} \hat{C}_{s'_1} \hat{C}_{s'_2} \hat{C}_{s'_3} \hat{C}_{s'_4} \hat{\Pi}_{s_d} \hat{U} \hat{C}_{s_{d-1}} \\ &\hat{C}_{s_{d-1}} \hat{\Pi}_{s_{d-1}} \hat{U} \dots \hat{C}_{s_1} \hat{C}_{s_1} \hat{\Pi}_{s_1} \hat{U} |\psi_L\rangle. \end{aligned} \quad (28)$$

Furthermore each round can be written in the form of Eq. (12), with \hat{U} replaced by inhomogeneous error operators $\hat{U}_j = \hat{U}\hat{C}_{s_{j-1}}$, and the normalization factor by $1/\sqrt{P(s_j)}$. Based on Eq. (13), we can express each round as a rotation about the Z axis,

$$\frac{1}{\sqrt{P(s_j)}} \hat{C}_{s_j} \hat{\Pi}_{s_j} \underbrace{\hat{U} \hat{C}_{s_{j-1}}}_{\hat{U}_j} |\psi_L\rangle = e^{i\theta_j^L \hat{Z}^L} |\psi_L\rangle, \quad (29)$$

where the logical rotation angle θ_j^L depends on all the previously measured syndromes (s_1, s_2, \dots, s_j) .

Finally one can write the final state for perfect syndrome \underline{s} , and noisy syndrome \underline{s}' as

$$|\Phi_{\underline{s}, \underline{s}'}\rangle = \hat{C}_{\underline{s}'} \hat{C}_{\underline{s}} e^{i\theta^*(\underline{s}) \hat{Z}^L} |\psi_L\rangle, \quad (30)$$

where the rotation angle $\theta^*(\underline{s})$ can be calculated from sampling single rounds of error correction with perfect syndromes and inhomogeneous coherent errors,

$$\theta^*(\underline{s}) = \sum_{j=1}^d \theta_j^L(s_1, s_2, \dots, s_j). \quad (31)$$

4 Numerical results

We used the fermionic linear optics method to simulate the surface code under coherent and readout errors, for code sizes up to $d = 19$. As detailed below, we sampled the logical rotation angle distribution, from which we computed – for the most susceptible initial states – the expectation value of the infidelity, which we call logical error rate. As code sizes were scaled up, we found threshold behaviour. In case the rates of coherent and readout errors were equal, $p = q$, we found that the threshold is close to the corresponding threshold of random Pauli Z + readout errors. Our results here are similar to those with perfect measurements by Bravyi et al[18].

We also investigated how, below the threshold, the logical error rates compare to those of the random Pauli Z + readout errors, and how the residual coherence in the logical error decreases as the code size is scaled up. As detailed below, we again find similar results to those with perfect measurements by Bravyi et al.[18]. Varying the rates of the two error processes independently, we mapped out the threshold on the (p, q) plane, and found that coherent errors are more critical

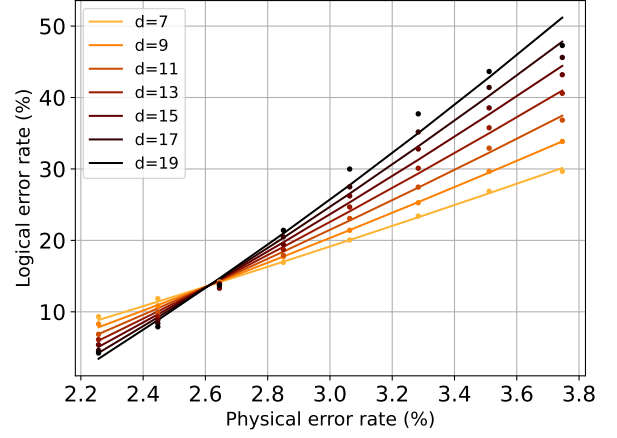


Figure 3: Error threshold as the code size is scaled up with coherent and readout error rates equal ($p = q$ is the "physical error rate"). Logical error rates for different code distances show threshold behaviour as functions of the physical error rate. For the fitted curves we used a finite size scaling ansatz[3], based on the statistical mechanical mapping of the surface code. Every point results from $5000 \times d \times 100$ rounds of simulation.

than readout errors: to achieve scalable error correction it is easier to compensate a high value of readout errors (at or above 10%) by reducing the rate of coherent errors than vice versa.

4.1 Threshold with equal coherent and readout errors

We ran extensive simulations to estimate the error threshold when the readout error rate q is set equal to the coherent error rate p . For every odd value of code distance d , up to $d = 19$, to obtain the numerical distribution of logical rotation angles, we sampled the noiseless 3D syndrome measurements 5000 times ($5000 d$ rounds), and then sampled 100 noisy syndromes from each of these. In Fig. 3, we show the resulting average logical error rate, calculated via Eqs. (24). We observe that for errors below a threshold, scaling up the code size decreases the logical errors, while above the threshold, scaling up only makes things worse by increasing the logical errors.

To obtain a precise value of the threshold, we fitted the numerical values using a finite size scaling ansatz[3], based on mapping of the surface code to statistical physics models[1]. Although the ansatz is strictly expected to work for random Pauli+readout errors, it also fits our numerics (coherent+readout errors). The threshold value

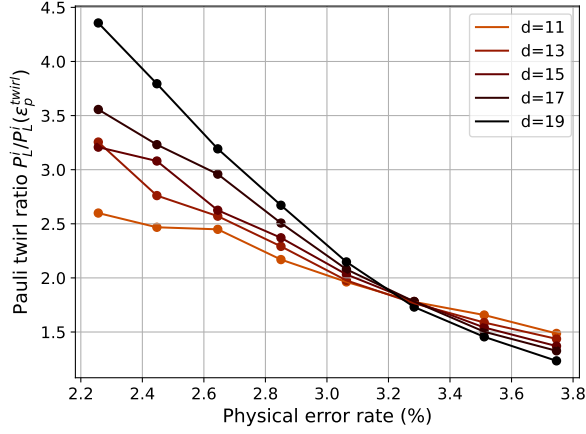


Figure 4: Pauli twirl ratio (for maximum infidelity) as functions of physical error rate (equal to readout error rate), for different code distances. Results obtained from the same data as in Fig. 3 and Monte Carlo simulations of incoherent noise.

is

$$p_{th}^i = 2.62\% \pm 0.02\%. \quad (32)$$

This is relatively close to the threshold of random Pauli + readout errors[3], which for the toric code is $p_{th} = 2.93\% \pm 0.02\%$. Using the diamond-norm distance, our numerics paints a somewhat different picture, as discussed in Appendix A.

4.2 Sub-threshold comparison with random Pauli+readout errors

We next compare the performance of the surface code under coherent+readout errors with its performance under random Pauli+readout errors, when error rates are below threshold. We use the same parameter $p = \sin^2(\theta)$ for the two channels, with the random Pauli channel being the Pauli twirled version of the coherent error channel,

$$\epsilon_{\theta}^{twirl}(\hat{\rho}) = \cos^2(\theta)\hat{\rho} + \sin^2(\theta)\hat{Z}\hat{\rho}\hat{Z}. \quad (33)$$

The *Pauli twirl ratio* is the ratio of logical error rate in case of coherent errors+readout errors, and the logical error rate when the coherent errors are replaced by their Pauli twirled version at the physical level. Therefore Pauli twirl ratio can be written as:

$$\frac{P^i(\epsilon_{\theta})}{P^i(\epsilon_{\theta}^{twirl})}; \quad \frac{P^d(\epsilon_{\theta})}{P^d(\epsilon_{\theta}^{twirl})}, \quad (34)$$

for the maximum infidelity and for the diamond-norm distance.

The numerically obtained values of the twirl ratio, shown in Fig. 4, indicate that below the threshold, coherent errors + readout errors lead to higher logical error rates than random Pauli + readout errors (high Pauli twirl ratio). Moreover, this difference grows as we scale up the size of the code. Note that in the limit of vanishing error rates, $p \rightarrow 0$, we expect the Pauli twirl ratio to reach a finite, code distance dependent value. The reason is that we expect the logical error rate for small p to scale as $P_L \approx N_d p^{(d+1)/2}$ for incoherent, and $P_L^i \approx N_d^2 (\sin^2(\theta))^{(d+1)/2}$ for coherent errors. Here N_d is the number of the shortest errors strings that cause a logical error, and it should be roughly the coherence ratio in the $p \rightarrow 0$ limit. Numerical investigation of the $p \rightarrow 0$ limit is, however, expensive due to the small error rates.

Interestingly, there is a threshold-like behavior of the Pauli twirl ratio, which is independent of code distance for $p \approx 3.3\%$, and decreases with code distance for p above. However, the value 3.3% is not the threshold of the code.

One of the key findings of Bravyi et al [18] is that quantum error correction "washes out" coherence in the logical level in the coherent errors (without readout errors). As the code distance increases, the distribution of logical rotation angles becomes more and more highly peaked around 0 and $\pi/2$, thus, the logical noise is better and better approximated by random Pauli processes. This property of quantum error correction has also been studied analytically[15, 16]. A practical quantity to study this effect is the *coherence ratio*[18], defined as

$$\frac{P_L^d}{P_L^{d,twirl}} = \frac{\sum_{\underline{s}, \underline{s}'} p(\underline{s}) p(\underline{s} \rightarrow \underline{s}') |\sin(\theta_L)|}{\sum_{\underline{s}, \underline{s}'} p(\underline{s}) p(\underline{s} \rightarrow \underline{s}') \sin^2(\theta_L)}, \quad (35)$$

where $P_L^{d,twirl}$ is just the diamond-norm distance for the twirled logical error channel (which is just the maximum infidelity up to a factor of 2). The coherence ratio is always greater than or equal to one, equality holds if θ_L only takes values $\{0, \pi/2\}$, i.e., if the logical noise is fully incoherent (probabilistic logical Z errors).

One would expect that this "washing out" of coherence is, if anything, made even stronger by the readout errors. Our numerics confirms this intuition. In Fig. 5 we show the coherence ratio of different code sizes with physical error rate and readout error rate set equal. We find that the

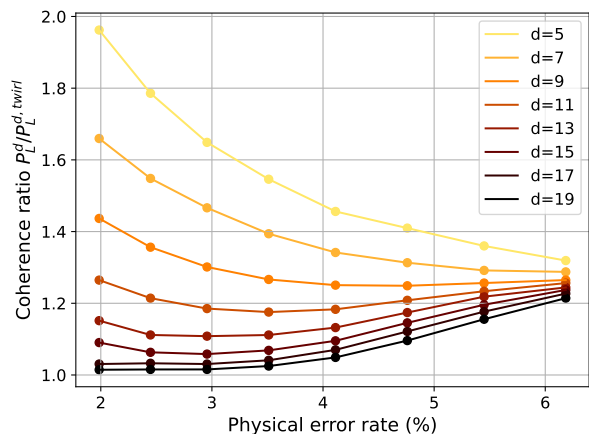


Figure 5: Coherence ratio as a function of error rate ($p = q$) with different code distances. Each point results from $5000 \times d \times 100$ rounds of simulation.

coherence ratio decreases as the code is scaled up, even above the threshold. Around the threshold, $p = q \approx 3\%$, practically all coherence is lost on the logical level, for code sizes $d = 17$ and above. Interestingly though, the coherence ratio appears to increase as the physical error rate is decreased from the threshold. Without readout errors[18], all of these qualitative trends are there, but the coherence ratio is 1.1 at the threshold even for a code size $d = 37$.

4.3 Independent coherent and readout errors

We have also investigated how varying the rate p of coherent errors and q of readout errors independently affects the threshold of the surface code. For many pairs of p and q we numerically ascertained whether scaling up the surface code decreases logical error rates (scalable QEC) or it increases them (unscalable QEC). The threshold should be in between these regions. We could not determine the threshold values more precisely, since the fitting ansatz we used for $p = q$ turned out to be a poor fit in many of the cases with asymmetric noise.

Our results, shown as a 2D map in Fig. 6, show that the surface code is more sensitive to coherent errors than to readout errors. If the coherent error rate is on the percent level, the surface code is quite robust against readout errors, scalable even with relatively high $q \approx 7\%$. However, if the readout error rate is on the percent level, the surface code still requires the coherent error rate to be below 3%.

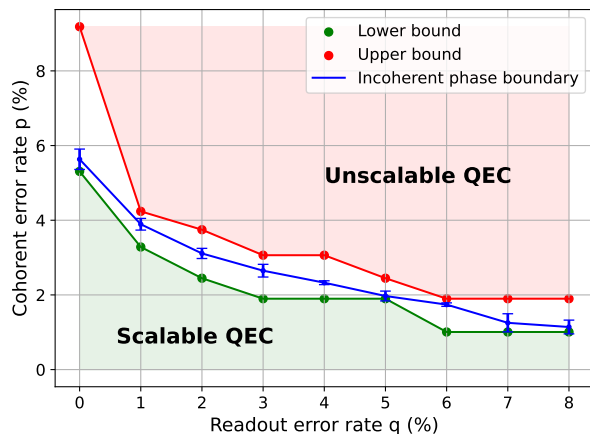


Figure 6: Threshold lines on the (p, q) plane for maximum infidelity (logical error rate). For a fix q value we run the simulation for 16 different θ values, and determined a lower and an upper bound for thresholds in each cases. The lower bound is the last point where the maximum infidelity is decreasing as the code distance is increasing. The upper bound however is the first point where the logical error rate is increasing as the distance is increasing. We numerically investigated the incoherent case via Monte Carlo simulations, and determined the threshold values for asymmetric p, q values with the fitting ansatz[3].

5 Discussion and outlook

We investigated numerically how well the surface code works as a quantum memory when there are coherent errors on the physical qubits as well as readout errors (phenomenological readout error model). We focused on a restricted class of coherent errors, namely, unitary phase rotations, $e^{i\theta\hat{Z}_j}$. This allowed us to use the theoretical tools of Fermionic Linear Optics, as applied to the surface code by Bravyi et al[18]. We extended that work by including readout errors as well, on a phenomenological level (perfect measurements, noisy recording of measurement results). The python source code for our numerical work is available on GitHub[29].

Our results show that the findings of Bravyi et al[18] on the effects of coherent errors on the surface code mostly carry over to when readout errors also occur. Namely, the surface code with coherent+readout errors has a threshold, which is close to that of the corresponding Pauli twirled error channel (random Pauli+readout errors). However, for error rates below the threshold, its logical error rates are significantly higher than that of the Pauli twirled error channel. Scal-

ing up the code size, coherence is washed out from the logical error. Moreover, we found that having a low value of the coherent errors is more important than a low value of readout errors (high readout error rates can be compensated by low coherent error rates, but less so vice versa).

A point that is worth further investigation is the differences in our results when using the diamond norm or the fidelity as quantitative measures of the reliability of quantum memory. As an example, these two measures gave quite different results regarding the threshold of coherent+readout errors: we observed a clear threshold using the fidelity, but less clear behavior using the diamond norm (Appendix A).

It would also be interesting to consider broader classes of coherent errors. A next step would be to consider coherent error parameters θ that are not constant, but vary from qubit to qubit or even fluctuate (this latter case modeling the combination of coherent and incoherent Z errors). A numerically more challenging question is how our results would be changed if even the axis of coherent rotation varied from qubit to qubit (not Z for all qubits as in our work) – unfortunately here the tools Bravyi et al.[18] do not apply. Even more challenging is to bring the error model closer to experimental reality, by modeling coherent errors on the circuit level. A step in this direction was taken recently, with a coherent error model that includes ancilla qubits, but uses a somewhat unrealistic multiqubit gate [30]. For this error model, a mapping to a three dimensional lattice gauge theory seems to suggest that when combined with incoherent errors, coherent errors ruin the threshold: even with arbitrarily small error rates, scaling the code size up beyond a certain size will increase noise the logical level. A related point is that effects of coherent errors can be mitigated in the surface code using extra ancilla qubits to realize code concatenation[31, 32], or in variants of Shor’s code by other means[33].

Acknowledgements

This research was supported by the Ministry of Culture and Innovation and the National Research, Development and Innovation Office (NKFIH) within the Quantum Information National Laboratory of Hungary (Grant No. 2022-2.1.1-NL-2022-00004), and by the NKIFH within

the OTKA Grant FK 132146. This research has been supported by the Horizon Europe research and innovation programme of the European Union through the IGNITE project and by the HORIZON-CL4-2022-QUANTUM01-SGA project 101113946 OpenSuperQPlus100 of the EU Flagship on Quantum Technologies. We also acknowledge support by the ÚNKP-22-2-II-BME-10 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund.

References

- [1] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. “Topological quantum memory”. *Journal of Mathematical Physics* **43**, 4452–4505 (2002).
- [2] Austin G Fowler, Matteo Mariantoni, John M Martinis, and Andrew N Cleland. “Surface codes: Towards practical large-scale quantum computation”. *Physical Review A* **86**, 032324 (2012).
- [3] Chenyang Wang, Jim Harrington, and John Preskill. “Confinement-Higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory”. *Annals of Physics* **303**, 31–58 (2003).
- [4] Héctor Bombin, Ruben S Andrist, Masayuki Ohzeki, Helmut G Katzgraber, and Miguel A Martin-Delgado. “Strong resilience of topological codes to depolarization”. *Physical Review X* **2**, 021004 (2012).
- [5] Christopher T Chubb and Steven T Flammia. “Statistical mechanical models for quantum codes with correlated noise”. *Annales de l’Institut Henri Poincaré D* **8**, 269–321 (2021).
- [6] Scott Aaronson and Daniel Gottesman. “Improved simulation of stabilizer circuits”. *Physical Review A* **70**, 052328 (2004).
- [7] Craig Gidney. “Stim: a fast stabilizer circuit simulator”. *Quantum* **5**, 497 (2021).
- [8] Sebastian Krinner, Nathan Lacroix, Ants Remm, Agustin Di Paolo, Elie Genois, Catherine Leroux, Christoph Hellings, Stefania Lazar, Francois Swiadek, Johannes

- Herrmann, et al. “Realizing repeated quantum error correction in a distance-three surface code”. *Nature* **605**, 669–674 (2022).
- [9] Rajeev Acharya et al. “Suppressing quantum errors by scaling a surface code logical qubit”. *Nature* **614**, 676 – 681 (2022).
- [10] Yu Tomita and Krysta M Svore. “Low-distance surface codes under realistic quantum noise”. *Physical Review A* **90**, 062320 (2014).
- [11] Daniel Greenbaum and Zachary Dutton. “Modeling coherent errors in quantum error correction”. *Quantum Science and Technology* **3**, 015007 (2017).
- [12] Andrew S Darmawan and David Poulin. “Tensor-network simulations of the surface code under realistic noise”. *Physical Review Letters* **119**, 040502 (2017).
- [13] Shigeo Hakkaku, Kosuke Mitarai, and Keisuke Fujii. “Sampling-based quasiprobability simulation for fault-tolerant quantum error correction on the surface codes under coherent noise”. *Physical Review Research* **3**, 043130 (2021).
- [14] Florian Venn, Jan Behrends, and Benjamin Béri. “Coherent-error threshold for surface codes from majorana delocalization”. *Physical Review Letters* **131**, 060603 (2023).
- [15] Stefanie J Beale, Joel J Wallman, Mauricio Gutiérrez, Kenneth R Brown, and Raymond Laflamme. “Quantum error correction decoheres noise”. *Physical Review Letters* **121**, 190501 (2018).
- [16] Joseph K Iverson and John Preskill. “Coherence in logical quantum channels”. *New Journal of Physics* **22**, 073066 (2020).
- [17] Mauricio Gutiérrez, Conor Smith, Livia Lulushi, Smitha Janardan, and Kenneth R Brown. “Errors and pseudothresholds for incoherent and coherent noise”. *Physical Review A* **94**, 042338 (2016).
- [18] Sergey Bravyi, Matthias Englbrecht, Robert König, and Nolan Peard. “Correcting coherent errors with surface codes”. *npj Quantum Information* **4** (2018).
- [19] F Venn and B Béri. “Error-correction and noise-decoherence thresholds for coherent errors in planar-graph surface codes”. *Physical Review Research* **2**, 043412 (2020).
- [20] Héctor Bombín and Miguel A Martin-Delgado. “Optimal resources for topological two-dimensional stabilizer codes: Comparative study”. *Physical Review A* **76**, 012305 (2007).
- [21] Nicolas Delfosse and Naomi H Nickerson. “Almost-linear time decoding algorithm for topological codes”. *Quantum* **5**, 595 (2021).
- [22] Sergey Bravyi, Martin Suchara, and Alexander Vargo. “Efficient algorithms for maximum likelihood decoding in the surface code”. *Physical Review A* **90**, 032326 (2014).
- [23] Austin G. Fowler. “Minimum weight perfect matching of fault-tolerant topological quantum error correction in average $o(1)$ parallel time”. *Quantum Info. Comput.* **15**, 145–158 (2015).
- [24] Eric Huang, Andrew C. Doherty, and Steven Flammia. “Performance of quantum error correction with coherent errors”. *Physical Review A* **99**, 022313 (2019).
- [25] Alexei Gilchrist, Nathan K. Langford, and Michael A. Nielsen. “Distance measures to compare real and ideal quantum processes”. *Physical Review A* **71**, 062310 (2005).
- [26] Christopher A Pattison, Michael E Beverland, Marcus P da Silva, and Nicolas Delfosse. “Improved quantum error correction using soft information”. preprint (2021).
- [27] Oscar Higgott. “Pymatching: A python package for decoding quantum codes with minimum-weight perfect matching”. *ACM Transactions on Quantum Computing* **3**, 1–16 (2022).
- [28] Alexei Kitaev. “Anyons in an exactly solved model and beyond”. *Annals of Physics* **321**, 2–111 (2006).
- [29] “FLO simulation of the surface code - python script”. https://github.com/martonaron88/Surface_code_FLO.git.
- [30] Yuanchen Zhao and Dong E Liu. “Lattice gauge theory and topological quantum error correction with quantum deviations in

the state preparation and error detection”. [preprint](#) (2023).

- [31] Jingzhen Hu, Qingzhong Liang, Narayanan Rengaswamy, and Robert Calderbank. “Mitigating coherent noise by balancing weight-2 z-stabilizers”. *IEEE Transactions on Information Theory* **68**, 1795–1808 (2021).
- [32] Yingkai Ouyang. “Avoiding coherent errors with rotated concatenated stabilizer codes”. *npj Quantum Information* **7**, 87 (2021).
- [33] Dripto M Debroy, Laird Egan, Crystal Noel, Andrew Ringer, Daiwei Zhu, Debopriyo Biswas, Marko Cetina, Chris Monroe, and Kenneth R Brown. “Optimizing stabilizer parities for improved logical qubit memories”. *Physical Review Letters* **127**, 240501 (2021).
- [34] S Bravyi and R König. “Classical simulation of dissipative fermionic linear optics”. *Quantum Information and Computation* **12**, 1–19 (2012).
- [35] Barbara M Terhal and David P DiVincenzo. “Classical simulation of noninteracting-fermion quantum circuits”. *Physical Review A* **65**, 032325 (2002).
- [36] Sergey Bravyi. “Lagrangian representation for fermionic linear optics”. *Quantum Information and Computation* **5**, 216–238 (2005).

A Results for diamond-norm distance

We have omitted some of the results for diamond-norm distance from the main part of our paper, because we think that maximum infidelity is a more convenient measure to use, but for the sake of completeness here we show the diamond-norm distance versions of Fig. 3, Fig. 4, and Fig. 6.

As far as the error threshold is concerned, we did not find the same convincing numerical evidence using the diamond-norm distance as we did with the infidelity. As shown in Fig. 7, the plots of the diamond-norm distance for different code sizes (quadratic curves fitted individually) do not intersect in the same point. Rather, the physical error rate where the curve of size d intersects that of the curve for size $d+2$ is smaller for larger code sizes. We try to estimate the threshold, i.e., a limit of the intersection points in the $d \rightarrow \infty$

limit, by plotting in the inset of Fig. 7 the physical error rates of these intersection points as the function of the inverse code size $1/d$. The data suggests a threshold around $p = 1.6\%$.

For the sake of completeness, we also include here plots of the Pauli twirl ratio, evaluated using the diamond-norm distance, Fig. 8, and a 2D map of how the finite-size threshold values depend on the coherent error rate p and readout error rate q , Fig. 9. These both show qualitatively and also quantitatively similar behavior to their infidelity-based counterparts, Figs. 4 and 6, respectively.

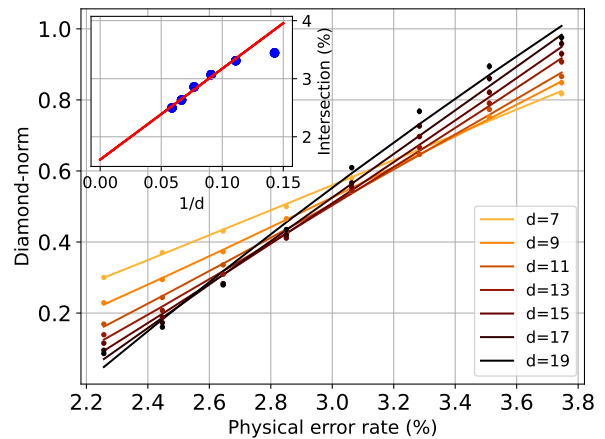


Figure 7: Diamond-norm distance as the function of physical error rate for several code distances. We fitted second order polynomials for each code distance, and plotted the intersections of these polynomials as the function of the inverse code distance. (At $1/d$ we placed the intersection of polynomials that correspond to code distances d and $d+2$.) We fitted a linear function to the first five data points, and read the $1/d = 0$ value as the threshold. Results obtained from the same data as in Fig. 3

B Sampling Coherent Errors in the Surface Code with Fermionic Linear Optics

Here we summarize the technical details of our simulations, which were based on the original proposal by Bravyi et al [18]. Previously we have shown Eq. (29) that the simulation of coherent+readout errors can be done by simulating error correction rounds with inhomogeneous coherent errors. Therefore here we restrict our description to single error correction rounds with perfect readout.

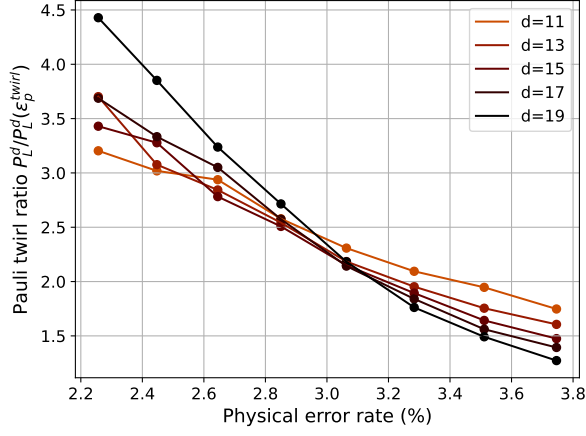


Figure 8: Pauli twirl ratio (for diamond-norm distances) as the function of physical error rate. Results obtained from the same data as in Fig. 4

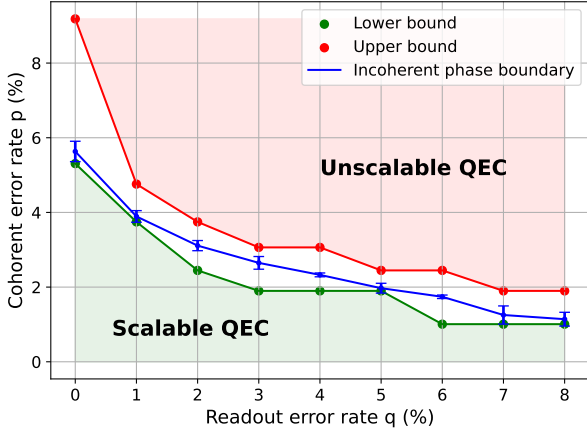


Figure 9: Threshold lines on the (p,q) plane for diamond-norm distance. Results obtained from the same data as in Fig. 6

First we introduce an exact mapping from the surface code to a fermionic system which can be described by Majorana operators. We show the fermionic linear optics algorithms for transforming the covariance matrix of the fermionic system. Then we describe the exact steps for sampling $\theta_L(s)$ from the probability distribution $P(s)$.

The aim of this Appendix is to help to understand the details of the fermionic linear optics algorithm[18]. For further details our code is also available on GitHub[29].

B.1 Encoding qubits into fermionic systems

As we mentioned in Sec.(3.1) physical qubits of the surface code can be represented as C4 codes.

However this mapping is not just a mathematical transformation, but a physically well-motivated thing.

The basic idea is to encode the physical qubits into double quantum dots with 2 fermionic modes described by creation and annihilation operators $\hat{a}_1^{(m)}, \hat{a}_1^{\dagger(m)}, \hat{a}_2^{(m)}, \hat{a}_2^{\dagger(m)}$ for the m -th qubit. With these fermionic operators in hand we can define the usual qubit basis states from the fermionic vacuum for one C4 code,

$$|0\rangle = \hat{a}_2^\dagger |\emptyset\rangle; \quad |1\rangle = \hat{a}_1^\dagger |\emptyset\rangle. \quad (36)$$

We can introduce Majorana operators in the following way:

$$\hat{c}_1 = i(\hat{a}_2 - \hat{a}_2^\dagger); \quad \hat{c}_2 = \hat{a}_1 + \hat{a}_1^\dagger; \quad (37)$$

$$\hat{c}_3 = i(\hat{a}_1 - \hat{a}_1^\dagger); \quad \hat{c}_4 = \hat{a}_2 + \hat{a}_2^\dagger. \quad (38)$$

As we define these Majorana operators for each qubit we are able to express encoded Pauli operators $\hat{X}_m, \hat{Z}_m, \hat{Y}_m$ and C4 stabilizers $\hat{S}^{(m)}$, as we did in Eqs. (26)(27).

It is important to note that $\hat{S}^{(m)}\hat{X}_m, \hat{S}^{(m)}\hat{Z}_m$ and $\hat{S}^{(m)}\hat{Y}_m$ are also good Pauli operators, we will take advantage of this later.

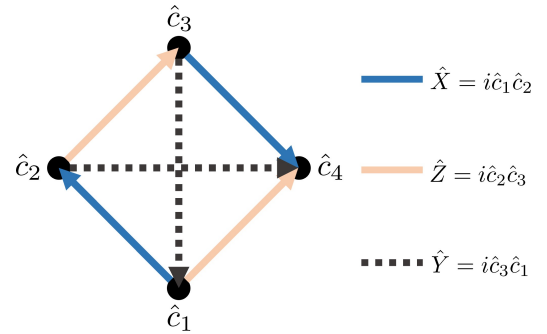


Figure 10: Visualization of one C4 code. Black dots represent Majorana fermions and arrows represent encoded Pauli operators (first Majorana operator is the tail second is the head). We have drawn the products of C4 stabilizer and Pauli operators too.

B.2 Majorana representation of surface code

With the help of encoded C4 codes now we can build up the whole surface code, and identify the Majorana representation of stabilizers and logical operators. First define some Majorana pairs, which connect Majorana fermions belong two different C4 codes, we call them *link operators*,

$$\hat{L}_e = i\hat{c}_p\hat{c}_q. \quad (39)$$

Here we get rid of the superscript index of the Majorana operators, and we use p, q indices instead. These are at the ends of edge e . Link operators assigned to every edge of the code patch, but the order of the Majorana operators is not well-defined (we have some freedom in the orientation of arrows, which represents link operators). The only restriction is that the product of link operators on the boundaries of faces have to be equal to the stabilizers on these specific faces,

$$\prod_{e \in \partial f} \hat{L}_e = \hat{S}_f \quad (40)$$

We can fulfill this restriction by building up our code patch from basic building blocks, see Fig 11. Note that every other C4 code is rotated by 90° , so only "proper" sides of C4 codes join to proper faces (X/Z operators to X/Z stabilizers).

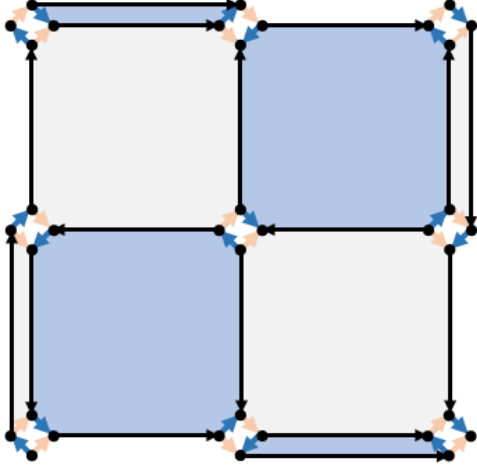


Figure 11: Basic building block of Majorana representation of the surface code. The orientation of link operators ensures Eq. (40). Bigger patches can be achieved by obvious duplication and translation of this small block.

Next step is the expression of logical operators with the Majorana fermions of C4 codes. First we can observe that every Majorana fermion is part of a link operator, except 4 corner Majoranas (see Fig. 12), from these Majoranas we can build up the so-called *logical C4 code* with logical C4 Pauli operators analogous to Eq. (26):

$$\hat{X}_{C4}^L = i\hat{c}_1^L \hat{c}_2^L; \quad \hat{Z}_{C4}^L = i\hat{c}_2^L \hat{c}_3^L. \quad (41)$$

Original logical operators of the surface code can be expressed with the help of these logical C4 operators and link operators based on the definition

Eq. (6):

$$\hat{X}^L = \hat{X}_{C4}^L \prod_{e \in \text{LEFT}} \hat{L}_e \quad \hat{Z}^L = \hat{Z}_{C4}^L \prod_{e \in \text{TOP}} \hat{L}_e. \quad (42)$$

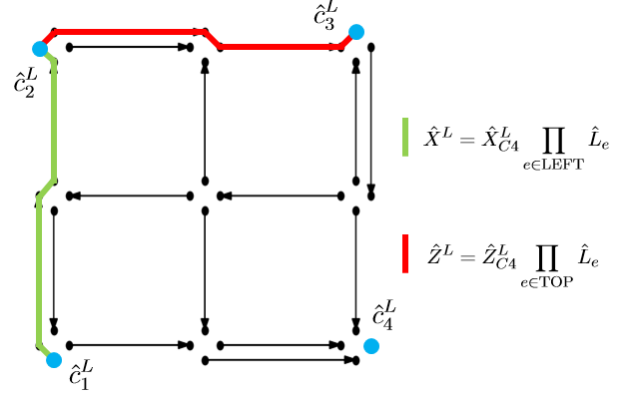


Figure 12: Logical operators and logical C4 code of the Majorana representation of the surface code.

One of the most important thing for the simulation is that we can express any logical state of the surface code with the help of Majorana operators. First define the state stabilized by all link operators,

$$\hat{L}_e |\Phi_{link}\rangle = |\Phi_{link}\rangle \quad \forall e. \quad (43)$$

Note this $|\Phi_{link}\rangle$ state is not an eigenstate of C4 codes, and it does not determine the state of logical Majoranas. The logically encoded version of state $|\psi\rangle$ can be expressed as

$$|\psi_L\rangle = 2^{(n-1)/2} \prod_{j=1}^n \frac{1}{2} (1 + \hat{S}^{(j)}) |\Phi_{link}\rangle \otimes |\psi_{C4}^L\rangle, \quad (44)$$

where $|\psi_{C4}^L\rangle$ is the encoded version of $|\psi\rangle$ into the logical C4 code.

One can easily check that this state is the +1 eigenstate of all the surface code stabilizers and the logical operators of the surface code are acting just as encoded logical C4 operators on $|\psi_{C4}^L\rangle$.

B.3 Fermionic Linear Optics

Fermionic Linear Optics (FLO) [34, 35, 36] simulations based on the fact that some specific fermionic states, consist of N fermions, can be fully characterized by just a simple $2N \times 2N$ matrix, these states are called *pure Gaussian states*, and the matrix is the *covariance matrix*,

$$M_{jk}(|\psi\rangle) = \langle \psi | i\hat{c}_j \hat{c}_k | \psi \rangle - i\delta_{jk}. \quad (45)$$

Here Majorana operators $(\hat{c}_1, \hat{c}_2, \dots, \hat{c}_{2n})$ describe the fermionic system of n fermions. Covariance matrix can be written in every state, but in pure Gaussian states it will be orthogonal. (It can be an alternative definition, and it is also easy to check). The most important fact is that pure Gaussian states are fully characterized by their covariance matrix [35], so simulation of pure Gaussian states can be realized by just tracking a matrix with $\mathcal{O}(n^2)$ elements.

The limitation of fermionic linear optics simulations is that we can only apply operations that bring the state from a pure Gaussian state to another pure Gaussian state. During the simulation of coherent errors in the surface code 3 kind of enabled operations appear[18]:

- Initialization of states stabilized by Majorana pairs $i\hat{c}_p\hat{c}_q$.
- Application of operator $e^{\theta\hat{c}_p\hat{c}_q}$.
- Measurement of Majorana pair $i\hat{c}_p\hat{c}_q$.

For clarifying these operations we express the covariance matrices induced by them.

The covariance matrix of a pure Gaussian state $|GS\rangle$ stabilized by $2n$ (p, q) pairs,

$$i\hat{c}_p\hat{c}_q|GS\rangle = |GS\rangle \quad \forall(p, q), \quad (46)$$

can be expressed as:

$$M_{jk} = \sum_{(p,q)} \delta_{jp}\delta_{kq} - \delta_{jq}\delta_{kp}. \quad (47)$$

The covariance matrix of the pure Gaussian state $e^{\theta\hat{c}_p\hat{c}_q}|GS\rangle$ can be written as:

$$M_{jk} = \langle GS| e^{-\theta\hat{c}_p\hat{c}_q} i\hat{c}_j\hat{c}_k e^{\theta\hat{c}_p\hat{c}_q} |GS\rangle - i\delta_{jk}. \quad (48)$$

Measurement of a Majorana pair can be realized by a projective measurement (we only deal with outcome +1 here, because -1 outcome can be obtained by measuring $i\hat{c}_q\hat{c}_p$ instead of $i\hat{c}_p\hat{c}_q$). So the covariance matrix of the post-measurement state can be expressed as:

$$M_{jk} = \frac{\langle GS| (1 + i\hat{c}_p\hat{c}_q) i\hat{c}_j\hat{c}_k (1 + i\hat{c}_p\hat{c}_q) |GS\rangle}{2 \langle GS| (1 + i\hat{c}_p\hat{c}_q) |GS\rangle}, \quad (49)$$

if $j \neq k$, and 0 otherwise. For further progress we need to express the transformed covariance matrices from the initial ones. As far as we are

working with pure Gaussian states we can use Wick's theorem for this purpose,

$$i^p \langle GS| \hat{c}_{j_1} \hat{c}_{j_2} \dots \hat{c}_{j_{2p}} |GS\rangle = \text{Pf}(M(|GS\rangle))_{j_1, j_2, \dots, j_{2p}}. \quad (50)$$

Therefore we are able to express the expectation value of any Majorana chain as the Pfaffian of a sub-matrix of the covariance matrix.

Through lengthy, but quite straightforward calculations one can derive the efficient algorithms, which realize the transformations of the covariance matrix expressed in Eqs. (48), (49).

Here we show the algorithm for the coherent rotation around the Z-axis Alg. 1, and the algorithm for the measurement of Majorana pairs Alg. 2

Algorithm 1: Rotation (M, θ, p, q)

```

M'[p, :] ← M[p, :] cos(2θ) − M[q, :] sin(2θ)
M'[q, :] ← M[q, :] cos(2θ) + M[p, :] sin(2θ)
M'[:, p] ← M[:, p] cos(2θ) − M[:, q] sin(2θ)
M'[:, q] ← M[:, q] cos(2θ) + M[:, p] sin(2θ)
M'[p, p] ← 0
M'[q, q] ← 0
M'[p, q] ←
    M[p, q] cos2(2θ) − M[q, p] sin2(2θ)
M'[q, p] ←
    −M[p, q] cos2(2θ) + M[q, p] sin2(2θ)
return M'

```

Algorithm 2: Measurement (M, p, q)

```

probability ← (1/2)(1 + M[p, q])
if p ≠ 0 then
    K ← M[p, :]
    L ← M[q, :]
    M' ← M + (1/2p)(LKT − KLT)
    M'[p, :] ← 0
    M'[q, :] ← 0
    M'[:, p] ← 0
    M'[:, q] ← 0
    M'[p, q] ← 1
    M'[q, p] ← −1
end
return (M', probability)

```

B.4 Simulating error correction

Now we will write a more detailed description on how to perform the first two steps of our al-

gorithm described in Sec. (3). In the spirit of Eq. (29) it is enough to simulate one error correction round for inhomogeneous coherent errors. However sampling from the probability distribution $P(s)$, and calculation of the assigned logical rotation angle $\theta_L(s)$ involve the previously introduced fermionic linear optics operations Algs. 1,2

One of the most painful restriction is that we can not simulate stabilizer measurements efficiently because they involve the joint measurement of 8 Majoranas. Therefore instead of measuring X stabilizers we measure X operators of individual C4 codes, and instead of sampling from probability distribution of syndromes, we are sampling from the distribution of X measurement outcomes. We are allowed to do this, because X stabilizer measurements can be led back to X measurements. For this purpose we introduce the concept of m syndrome, which stores the X measurement outcomes.

$$m \in \{+1, -1\}^n \quad (51)$$

We assign an s syndrome to every m syndrome through the following function:

$$s = \delta(m) : \quad s_f = \prod_{j \in \partial f} m_j. \quad (52)$$

We can sample from probability distribution of m syndromes $P(m)$, because

$$P(s) = \sum_{m: \delta(m)=s} P(m). \quad (53)$$

It is important to note that this is allowed only, because X measurements are commuting just like stabilizer measurements. If we want to measure both kind of (X and Z) stabilizers this simplification to the measurement of individual Pauli operators is prohibited, because of the non-trivial commutation relation. This is the reason behind our limitation of coherent errors to rotations around Z-axis, because in this case Z stabilizer measurements will be trivial, and we don't have to take them into account.

In the simulation we are sampling probability distribution $P(m)$ with a Monte Carlo algorithm. We order the measurement outcomes m_1, m_2, \dots, m_n , and draw m_j based on its conditional probability:

$$P(m_j | m_1, m_2, \dots, m_{j-1}) = \frac{P(m_1, m_2, \dots, m_j)}{P(m_1, m_2, \dots, m_{j-1})}. \quad (54)$$

We define the probability of a syndrome-part m_A :

$$P(m_A) = P(m_1, m_2, \dots, m_j), \quad (55)$$

where A is the patch including the first j qubit of the code.

We can express this probability $P(m_A)$ by considering the Majorana representation of the logical + state Eq. (44) (note $P(s)$ is independent of the initial state, so we can choose $|+_L\rangle$ safely).

$$P(m_A) = \nu \langle \Phi_{link} | \langle +_{C4}^L | \hat{O}_1^\dagger, \dots, \hat{O}_j^\dagger \hat{O}_j, \dots, \hat{O}_1 | \Phi_{link} \rangle | +_{C4}^L \rangle, \quad (56)$$

Where ν is just an A dependent normalization factor, $|+_L\rangle$ is the encoded $|+\rangle$ state in the logical C4 code and \hat{O}_j operation contains the projection to the C4 subspace of the j -th C4 code, the coherent error on the j -th qubit and the measurement of \hat{X}_j . This operator has the following form:

$$\hat{O}_k = \frac{1}{2} (1 + m_k \hat{X}_k) e^{i\theta_k \hat{Z}_k} \frac{1}{2} (1 + \hat{S}^{(k)}). \quad (57)$$

$|\Phi_{link}\rangle |+_L\rangle$ state is stabilized by all the link operators, \hat{X}_{C4}^L and $\hat{S}^L \hat{X}_{C4}^L$ operators of the logical C4 code, so it is clearly a Gaussian state. Furthermore we are able to write the operator \hat{O}_k as it only contains enabled fermionic linear optics operations (measurement of Majorana pairs, and coherent Z rotations),

$$\hat{O}_k = \frac{1}{2} (1 + m_k \hat{S}^{(k)} \hat{X}_k) \frac{1}{2} (1 + m_k \hat{X}_k) e^{i\theta_k \hat{Z}_k}. \quad (58)$$

Now it is clear that we can sample from the probability distribution $P(m)$ through fermionic linear optics simulations with a Monte Carlo algorithm, but we need to calculate the final state for a given syndrome, or at least the logical rotation angle $\theta_L(s)$. For this purpose we write $\tan^2(\theta_L(s))$ in the following form:

$$\tan^2(\theta_L(s)) = \left| \frac{\langle +_L | \hat{Z}^L | \Phi_s \rangle}{\langle +_L | \Phi_s \rangle} \right|^2. \quad (59)$$

Starting from the product state $|+\rangle^{\otimes n}$ and measure all Z stabilizers we can get $|+_L\rangle$ state. However since measuring Z stabilizers commute with the measurement of X stabilizers, correction operator and coherent Z errors, we can write Eq. (59) as,

$$\tan^2(\theta_L) = \left| \frac{\langle +^{\otimes n} | \hat{Z}^L | \Phi_s \rangle}{\langle +^{\otimes n} | \Phi_s \rangle} \right|^2. \quad (60)$$

Finally one can observe that these expressions are just special cases for $P(m)$, because $|+\rangle^{\otimes n} \langle +|^{\otimes n} = \hat{\Pi}_{m=0}$ is just the projection when all $m_j = 1$. So finally we can write

$$\tan^2(\theta_L) = \frac{\langle +_L | \hat{U}_-^\dagger \hat{\Pi}_{m=0} \hat{U}_- | +_L \rangle}{\langle +_L | \hat{U}_+^\dagger \hat{\Pi}_{m=0} \hat{U}_+ | +_L \rangle}, \quad (61)$$

where $\hat{U}_- = \hat{Z}^L \hat{C}_s \hat{U}$ and $\hat{U}_+ = \hat{C}_s \hat{U}$, so these are just special cases of inhomogeneous coherent errors: $\prod_j \exp(i\theta_j \hat{Z}_j)$.

One can also derive the following relation in a similar way:

$$\tan^2(\theta_L - \pi/4) = \frac{\langle Y_L | \hat{U}_-^\dagger \hat{\Pi}_{m=0} \hat{U}_- | Y_L \rangle}{\langle Y_L | \hat{U}_+^\dagger \hat{\Pi}_{m=0} \hat{U}_+ | Y_L \rangle}, \quad (62)$$

where $|Y_L\rangle$ is the logical $+1$ eigenstate of \hat{Y}^L operator. With Eqs. (61),(62) in hand we are able to determine the logical rotation angle modulo π .

An additional thing which is not crucial, but it can help you to reduce the cost of the simulation is the simulation only the active part of the covariance matrix. This means that you don't need to store the whole covariance matrix, only the non-trivial ($M_{jk} \neq 1, M_{jk} \neq 0$) sub-matrix. This can be done due to the fact that the measurements of Majorana pairs unbuckle rows and columns from the active part of the covariance matrix. Rotations are bringing in some new rows and columns, but the size of the active sub-matrix will be still notably smaller than the full covariance matrix.