

OPTIMAL BACON-SHOR CODES

JOHN NAPP and JOHN PRESKILL

*Institute for Quantum Information and Matter, California Institute of Technology
Pasadena, CA 91125, USA*

We study the performance of Bacon-Shor codes, quantum subsystem codes which are well suited for applications to fault-tolerant quantum memory because the error syndrome can be extracted by performing two-qubit measurements. Assuming independent noise, we find the optimal block size in terms of the bit-flip error probability p_X and the phase error probability p_Z , and determine how the probability of a logical error depends on p_X and p_Z . We show that a single Bacon-Shor code block, used by itself without concatenation, can provide very effective protection against logical errors if the noise is highly biased ($p_Z/p_X \gg 1$) and the physical error rate p_Z is a few percent or below. We also derive an upper bound on the logical error rate for the case where the syndrome data is noisy.

Keywords: Quantum error correction

1 Introduction

Bacon-Shor codes [1, 2] are quantum subsystem codes [3, 4] which are well suited for applications to fault-tolerant quantum memory [5, 6], because error syndrome information can be extracted by measuring only two-qubit operators that are spatially local if the qubits are arranged in a two-dimensional lattice. In this paper we assess the performance of these codes.

We consider noise models such that qubits in the code block are subject to both bit flip (X) errors and dephasing (Z) errors, where the bit flips occur with probability p_X and phase errors occur with probability p_Z . We assume that the noise acts independently on each qubit, and that the X and Z errors are uncorrelated. Under these assumptions we find the optimal block size of the code, and the failure probability achieved by this optimal code. We obtain analytic formulas for the optimal failure probability for the case of unbiased noise ($p_X = p_Z$) and the case of highly biased noise ($b \equiv p_Z/p_X \gg 1$). In both cases the formula applies in the asymptotic limit of small p_Z . Our results show that the failure probability of the optimal code falls exponentially in $1/p_Z$. We also consider the case where the code's syndrome bits are prone to error, and derive upper bounds on the failure probability in that case.

We find that for the case of unbiased noise ($p_Z = p_X \equiv p$), the optimal Bacon-Shor code

achieves the failure probability

$$BSOptimalFailProb(p) = \left(\frac{2}{\pi \ln 2}\right)^{1/2} \exp\left(\frac{\ln^2 2}{8}\right) p^{1/2} \exp\left(-\frac{\ln^2 2}{8p} + O(p)\right) \quad (1)$$

or

$$\ln[BSOptimalFailProb(p)] = -A/p - (0.5) \ln(1/p) + C + O(p), \quad (2)$$

where

$$A = .0600566, \quad C = .0175217. \quad (3)$$

For the case of highly biased noise, we find

$$\ln[BSOptimalFailProb(p_Z, b)] = -A(b)/p_Z - (0.5) \ln(1/p_Z) + C(b) + O(p_Z \text{ polylog}(b)), \quad (4)$$

where

$$A(b) = \frac{1}{8} \left(W(\sqrt{b})\right)^2 + O(b^{-1/2} \ln b); \quad (5)$$

here W denotes the Lambert W function, with asymptotic expansion

$$W(\sqrt{b}) = \ln \sqrt{b} - \ln \ln \sqrt{b} + \frac{\ln \ln \sqrt{b}}{\ln \sqrt{b}} + O\left(\frac{\ln \ln \sqrt{b}}{\ln^2 \sqrt{b}}\right). \quad (6)$$

We also compute the asymptotic form of $C(b)$ for $b \gg 1$.

In the case where the error syndrome is noisy, the reliability of the syndrome can be improved by measuring it repeatedly. We consider an idealized noise model such that qubit errors and syndrome measurement errors are equally likely and independent. For that model we derive a lower bound on the coefficient $\tilde{A}(b)$ of $1/p_Z$ in the natural logarithm of the logical failure rate, finding

$$\tilde{A}(b) \geq \frac{Z(b)}{4\mu^2} W\left(\sqrt{\frac{b}{4Z(b)}}\right), \quad (7)$$

where $\mu = 2.63816$ is the connective constant of a self-avoiding walk on a two-dimensional square lattice, and $Z(b)$ is a slowly varying monotonic function which ranges from $1/e$ to 1 as b increases from 1 to infinity. This bound applies for any value of $b \geq 1$.

Previous work [7, 8] on the applications of Bacon-Shor codes to fault-tolerant quantum computing has focused on relatively small codes used at the bottom layer of a concatenated coding scheme. Here we emphasize that a sufficiently large Bacon-Shor code, used by itself without concatenation, can also be quite effective if the physical error rate is low enough. For example, if the syndrome is perfect, then the probability of a logical failure is below 2×10^{-19} for $p_Z = .01$, $b = 100$, and below 10^{-12} for $p_Z = .03$, $b = 1000$. Fault-tolerant circuits based on Bacon-Shor codes will be more fully discussed and analyzed in a separate paper [9], where we consider in particular the case of highly biased noise.

2 2D Bacon-Shor code

The two-dimensional Bacon-Shor code [1, 2] is a quantum subsystem code (really a family of codes), which pieces together two dual quantum repetition codes, one protecting against bit flip errors and one protecting against phase errors. Like any subsystem code [3, 4], it can be defined by its “gauge algebra” — a set of Pauli operators that commute with the logical operators. The center of the gauge algebra is the code’s stabilizer group, and the code space is determined by fixing the eigenvalues of all the stabilizer operators to be (say) $+1$. Other operators in the gauge algebra act nontrivially on “gauge qubits” but trivially on the code’s protected qubits.

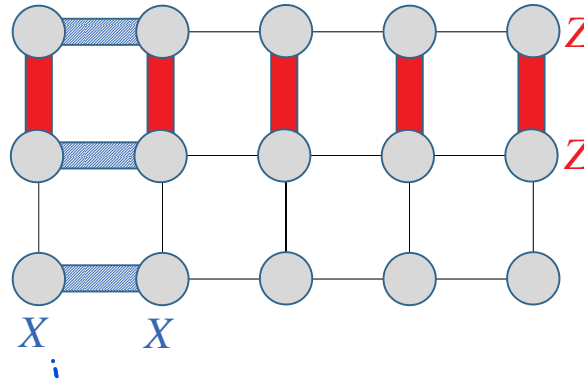
We can't seem to do XX in stim? but if for gauge algebra maybe not necessary

logical ops?

Specifically, we consider combining a length- m repetition code to protect against Z errors with a length- n repetition code to protect against X errors, where Z and X denote the single-qubit Pauli operators σ_Z and σ_X . The resulting Bacon-Shor code block contains mn qubits arranged at the sites of an $m \times n$ square lattice. The gauge algebra is generated by two-qubit operators XX acting on pairs of neighboring qubits in the same horizontal row of the lattice and two-qubit operators ZZ acting on neighboring qubits in the same vertical column. There is one protected qubit, and we may choose the logical Pauli operators to be $\bar{X} = X^{\otimes n}$ acting on all the qubits in a column and $\bar{Z} = Z^{\otimes m}$ acting on all the qubits in a row. The stabilizer group has $m + n - 2$ generators — there are $m - 1$ X -type generators, each a product of $2n$ X s acting on all the qubits in a pair of neighboring columns, and $n - 1$ Z -type generators, each a product of $2m$ Z s acting on all the qubits in a pair of neighboring rows. (We say that a Pauli operator is X -type if it is a tensor product of X s and identity operators, and we say it is Z -type if it is a tensor product of Z s and identity operators.) Aside from the protected logical system, there are $(m - 1)(n - 1)$ gauge qubits.

Generator vs Gauge Qubit?

4 X -type generators, each a product of 6 X s acting on all of the qubits in a pair of neighbouring columns



2 Z -type generators, each a product of 10 Z s acting on all qubits in a pair of neighbouring rows

Fig. 1. Bacon-Shor code block for $m = 5$ and $n = 3$. A weight- $(2m)$ Z -type stabilizer generator is the product of m weight-two Z -type gauge operators (solid red); a weight- $(2n)$ X -type stabilizer generator is the product of n weight-two X -type gauge operators (hashed blue).

stabilizer generator == generator

weight = number of pauli operators in product of generator

Though each Z -type stabilizer generator (or “check operators”) has weight $2m$, its value can be ascertained by measuring m weight-two gauge operators. For example, the product of $2m$ Z s in the first two rows is equivalent to the product of m Z -type gauge operators, each acting on a pair of qubits in the first two positions of the same column. To determine the check operator, we may measure these m gauge operators (obtaining either $+1$ or -1 for each outcome) and multiply the results. Similarly, the value of a weight- $2n$ X -type check

?

This means that can
reuse the qubits

operator can be found by measuring n weight-two X -type gauge operators and multiplying the results. Furthermore, since all check operators commute with all gauge-qubit operators, measuring X -type gauge qubit operators to determine the value of an X -type check operator does not disturb the values of any Z -type check operators (though it may flip the value of Z -type gauge operators), and vice-versa.

If m and n are both odd, then the code can correct $(m-1)/2$ Z errors and $(n-1)/2$ X errors. To describe the error recovery procedure, it is useful to exploit the freedom to “fix the gauge”, *i.e.* apply gauge operators to the block that commute with the logical qubit operators. An even number of X errors in a row can be “gauged away” by applying gauge operators. If there are an odd number of X errors in a row, we may “gauge away” all errors except one, which can be moved into the first position in the row. Thus, in an appropriate gauge, X errors occur only in the first column. Likewise, in an appropriate gauge, Z errors occur only in the first row. In this gauge, then, the check operators test whether two neighboring qubits in the first column agree in the Z basis, and whether two neighboring qubits in the first row agree in the X basis. Error recovery proceeds by applying X to at most $(n-1)/2$ qubits in the first column, to restore all Z -type check operators to the value $+1$, and by applying Z to at most $(m-1)/2$ qubits in the first row, to restore all X -type check operators to the value $+1$. Thus, error recovery is successful if the number of rows with an odd number of X errors is at most $(n-1)/2$ and if the number of columns with an odd number of Z errors is at most $(m-1)/2$.

If we choose the gauge so that $ZZ = +1$ for any pair of qubits in the same column, then the eigenstates of \bar{X} with eigenvalues ± 1 become

$$|\pm\rangle_C \propto (|00\cdots 0\rangle \pm |11\cdots 1\rangle)^{\otimes m}; \quad (8)$$

these are tensor products of m length- n “cat states” in the standard basis, one for each column. If we choose the gauge so that $XX = +1$ for any pair of qubits in the same row, then the eigenstates of \bar{Z} with eigenvalues ± 1 becomes

$$\begin{aligned} |0\rangle_C &\propto (|++\cdots+\rangle + |--\cdots-\rangle)^{\otimes n}, \\ |1\rangle_C &\propto (|++\cdots+\rangle - |--\cdots-\rangle)^{\otimes n}; \end{aligned} \quad (9)$$

these are tensor products of n length- m “cat states” in the dual basis, one for each row.

To perform a destructive measurement of the logical operator \bar{Z} , we can measure all mn qubits in the Z basis, compute the parity of the outcomes for each row, and then decode the measurement by performing a majority vote on the row parities. Likewise, to perform a destructive measurement of the logical operator \bar{X} , we can measure all mn qubits in the X basis, compute the parity of the outcomes for each column, and then decode the measurement by performing a majority vote on the column parities.

3 Failure probability for the 2D Bacon-Shor code

Consider an independent noise model in which the probability of an X error for each qubit is p_X , and the probability of a Z error is p_Z (that is, for each qubit, there is no error with probability $(1-p_X)(1-p_Z)$, an X error with probability $p_X(1-p_Z)$, a Z error with probability $(1-p_X)p_Z$ and an $iY = ZX$ error with probability p_Xp_Z). For any fixed values of p_X and p_Z ,

there is an optimal choice for the code dimensions m and n which minimizes the probability of a logical error. We will estimate the optimal failure probability in the limit of small p . Since X and Z error recovery are identical (except for the interchange of the rows and columns of the lattice), we will estimate the probability of a logical Z error; the same derivation also applies to the probability of a logical X error.

First we note that for a column of length n , if Z errors occur independently with probability p at each position, then the probability of an odd number of errors in the column is

$$\text{OddProb}(p, n) = \frac{1}{2} (1 - (1 - 2p)^n). \quad (10)$$

We derive this formula by observing that the terms of even order in p in the binomial expansions of $((1 - p) + p)$ and $((1 - p) - p)$ are identical, while the terms of odd order in p are the same except for a sign flip; hence the difference $\frac{1}{2} (1 - (1 - 2p)^n)$ sums up all the odd-order terms. To minimize the probability of a logical error for given p , we will choose n and m so that $p^2 mn$ is $O(1)$. Therefore in the limit of large n we may use the approximation

$$\text{OddProb}(p, n) = \frac{1}{2} (1 - \exp[-2p(1 + p)n + O(p^3 n)]). \quad (11)$$

We keep track of the $O(p^2 n)$ in the exponential because $\text{OddProb}(p, n)$ will be raised to a power $O(m)$ in the expression for the Bacon-Shor failure probability.

Now consider a length- m repetition code, where bit errors occur independently with probability x . If m is odd, an encoded error occurs if the number of bit errors is $(m + 1)/2$ or more. Therefore the failure probability of the repetition code is

$$\begin{aligned} \text{RepFailProb}(x, m) &= \sum_{k=(m+1)/2}^m \binom{m}{k} x^k (1 - x)^{m-k} \\ &= \left(\frac{x}{1 - x} \right)^{1/2} [x(1 - x)]^{m/2} \sum_{r=0}^{(m-1)/2} \binom{m}{\frac{m+1}{2} + r} \left(\frac{x}{1 - x} \right)^r. \end{aligned} \quad (12)$$

From the Stirling approximation,

$$\binom{m}{\frac{m+1}{2} + r} \approx 2^m \left(\frac{2}{\pi m} \right)^{1/2} \exp \left(-\frac{2}{m} \left(r + \frac{1}{2} \right)^2 \right), \quad (13)$$

neglecting a multiplicative $O(1/m)$ correction. Making another $O(1/m)$ multiplicative error, we may replace the exponential inside the sum over r by 1, obtaining

$$\text{RepFailProb}(x, m) \approx 2^m \left(\frac{2}{\pi m} \right)^{1/2} \left(\frac{x}{1 - x} \right)^{1/2} [x(1 - x)]^{m/2} \sum_{r=0}^{(m-1)/2} \left(\frac{x}{1 - x} \right)^r, \quad (14)$$

and we also make a negligible error by extending the upper limit on the sum to infinity, finding

$$\sum_{r=0}^{\infty} \left(\frac{x}{1 - x} \right)^r = \frac{1 - x}{1 - 2x}, \quad (15)$$

and thus

$$\text{RepFailProb}(x, m) \approx \left(\frac{2}{\pi m}\right)^{1/2} \left(\frac{x(1-x)}{(1-2x)^2}\right)^{1/2} [4x(1-x)]^{m/2}. \quad (16)$$

To compute the probability of a Z -type logical error for the Bacon-Shor code, we substitute $\text{OddProb}(p, n)$ for x in the expression for $\text{RepFailProb}(x, m)$, finding

$$\begin{aligned} [4x(1-x)]^{m/2} &\approx (1 - \exp[-4p(1+p)n + O(p^3n)])^{m/2} \\ &= \exp\left(\frac{2p^2mn[1 + O(p) + O(p^2n)]}{e^{4pn} - 1}\right) (1 - e^{-4pn})^{m/2}, \end{aligned} \quad (17)$$

and

$$\frac{x(1-x)}{(1-2x)^2} \approx \frac{1}{4} (e^{4pn} - 1); \quad (18)$$

hence

$$\begin{aligned} \text{BSZFailProb}(p_Z, m, n) &= \text{RepFailProb}(\text{OddProb}(p_Z, n), m) \\ &\approx \frac{1}{\sqrt{2\pi m}} \exp\left[\frac{2p_Z^2mn}{e^{4p_Zn} - 1}\right] (e^{4p_Zn} - 1)^{1/2} (1 - e^{-4p_Zn})^{m/2}, \end{aligned} \quad (19)$$

up to multiplicative corrections higher order in p_Z (assuming p_Zn and p_Zm are $O(1)$). By the same argument, the probability of an X -type logical error is given by a similar expression, but with m and n interchanged and p_Z replaced by p_X :

$$\begin{aligned} \text{BSXFailProb}(p_X, m, n) &= \text{RepFailProb}(\text{OddProb}(p_X, m), n) \\ &\approx \frac{1}{\sqrt{2\pi n}} \exp\left[\frac{2p_X^2mn}{e^{4p_Xm} - 1}\right] (e^{4p_Xm} - 1)^{1/2} (1 - e^{-4p_Xm})^{n/2}, \end{aligned} \quad (20)$$

up to multiplicative corrections higher order in p_X (assuming p_Xm and p_Xn are $O(1)$).

3.1 *Unbiased noise*

If the noise is unbiased ($p_X = p_Z \equiv p$), then the optimal Bacon-Shor code is symmetric ($m = n$), and the X -type and Z -type logical errors occur with equal probability. To find the optimal value of n for small p , we note that

$$\ln[\text{BSFailProb}(p, n)] \approx \frac{n}{2} \ln(1 - e^{-4pn}) + \dots, \quad (21)$$

where the ellipsis indicates corrections suppressed by powers of p or $1/n$. This expression, regarded as a function of n , attains its minimum when $y = pn$ satisfies

$$\ln(1 - e^{-4y}) + \frac{4y}{e^{4y} - 1} = 0, \quad (22)$$

or $4y = \ln 2$. (Though n is actually required to be an odd integer, we may ignore this requirement if p is small and n is correspondingly large.) Substituting $4pn \approx \ln 2$ into the

expression for the Bacon-Shor failure probability Eq.(19) we find

$$\begin{aligned} BSOptimalFailProb(p) &\approx \left(\frac{2}{\pi \ln 2}\right)^{1/2} \exp\left(\frac{\ln^2 2}{8}\right) p^{1/2} \exp\left(-\frac{\ln^2 2}{8p}\right) \\ &= (1.01768)\sqrt{p} \exp\left(-\frac{.0600566}{p}\right), \end{aligned} \quad (23)$$

or

$$\ln[BSOptimalFailProb(p)] \approx -A/p + B \ln p + C + O(p) \quad (24)$$

where

$$A = .0600566, \quad B = .5, \quad C = .0175217. \quad (25)$$

This optimal failure probability is achieved by choosing the linear size n of the code such that

$$pn = \frac{1}{4} \ln 2 = .173287. \quad (26)$$

In terms of the asymptotically optimal linear size $n = \frac{\ln 2}{4p}$, the optimal failure probability can be expressed as

$$\begin{aligned} BSOptimalFailProb(n) &\approx \left(\frac{2}{\pi \ln 2}\right)^{1/2} \exp\left(\frac{\ln^2 2}{8}\right) \left(\frac{\ln 2}{4n}\right)^{1/2} \exp\left(-\frac{n \ln 2}{2}\right) \\ &= \frac{2^{\ln 2/8}}{\sqrt{2\pi n}} 2^{-n/2}. \end{aligned} \quad (27)$$

Although there is strictly speaking no accuracy threshold for this family of codes, the performance of the optimal code is very good when the error rate is sufficiently small. For example, for $p = .001$, the optimal value of n is 173, and the corresponding failure probability is 2.638×10^{-28} , while the asymptotic formula Eq.(23) predicts 2.663×10^{-28} .

3.2 Highly biased noise

If the noise is highly biased ($b \equiv p_Z/p_X \gg 1$), then the optimal Bacon-Shor code is highly asymmetric — it combines a length- m repetition code to protect against Z errors with a length- n repetition code to protect against X errors, where $m \gg n$. In the limit of large noise bias, we can find analytic formulas for the optimal values of m and n and for the corresponding optimal probability of a logical error.

From Eq.(19) and Eq.(20), we have

$$\begin{aligned} \ln[BSZFailProb(p_Z, m, n)] &\approx \frac{m}{2} \ln(1 - e^{-4p_Z n}) + \dots, \\ \ln[BSXFailProb(p_X, m, n)] &\approx \frac{n}{2} \ln(1 - e^{-4p_X m}) + \dots. \end{aligned} \quad (28)$$

where the ellipsis indicates corrections suppressed by powers of p_Z , p_X , m^{-1} , n^{-1} . For $b \equiv p_Z/p_X \gg 1$, the optimal values of m and n are such that $4p_X m$ and $e^{-4p_Z n}$ are both

small; hence we can justify keeping the leading terms in a power series expansion in these small quantities, obtaining

$$\begin{aligned}\ln [BSZFailProb(p_Z, m, n)] &\approx -\frac{m}{2} \exp(-4p_Z n) - \frac{m}{4} \exp(-8p_Z n) + \dots, \\ \ln [BSXFailProb(p_X, m, n)] &\approx \frac{n}{2} \ln(4p_X m) - p_X m n + \dots.\end{aligned}\quad (29)$$

For now we neglect the non-leading terms in the expansion of both logarithms, which we will verify *a posteriori* are additive corrections of order $b^{-1/2} \ln b$.

We will minimize the failure probability subject to the constraint that (the leading contributions to) the logs of the Z and X failure probabilities are equal. Because we are neglecting the higher order terms in both Eq.(28) and Eq.(29), the solution we find may not be the true optimum, but we will see that it provides an upper bound on the optimal failure probability which is reasonably tight provided $p_Z \ln^4 b$ and $b^{-1/2} \ln b$ are small. Defining the variables

$$X = 4p_X m, \quad Z = 4p_Z n, \quad (30)$$

we want to find the values of X and Z that minimize the function

$$F(X, Z) = Z \ln X \quad (31)$$

(the log of the failure probability multiplied by $8p_Z = 8bp_X$) subject to the constraint

$$Z \ln X = -bX e^{-Z}. \quad (32)$$

Introducing a Lagrange multiplier λ to do the constrained minimization, we obtain the equations

$$-(\lambda - 1)Z = \lambda bX e^{-Z} = (\lambda - 1) \ln X \quad (33)$$

together with the constraint equation Eq.(32); these equations imply

$$X = e^{-Z} \quad \text{and} \quad Ze^Z = \sqrt{b} \quad \Rightarrow \quad X = Z/\sqrt{b}. \quad (34)$$

Evaluating F at its minimum yields

$$F_{\min} = -Z^2. \quad (35)$$

The solution to the equation $Ze^Z = \sqrt{b}$ is the Lambert W function $Z = W(\sqrt{b})$, which has the asymptotic expansion

$$Z = W(\sqrt{b}) = \ln \sqrt{b} - \ln \ln \sqrt{b} + \frac{\ln \ln \sqrt{b}}{\ln \sqrt{b}} + O\left(\frac{\ln \ln \sqrt{b}}{\ln^2 \sqrt{b}}\right) \quad (36)$$

for $b \gg 1$, and the log of the optimal failure probability is

$$\ln [BSZFailProb(p_Z, m, n)] \approx \ln [BSXFailProb(p_Z, m, n)] \approx -\frac{W^2(\sqrt{b})}{8p_Z} + \dots \quad (37)$$

The optimal code has dimensions

$$\begin{aligned} n &= \frac{Z}{4p_Z} \approx \frac{1}{4p_Z} W(\sqrt{b}) \approx \frac{1}{4p_Z} \ln \sqrt{b}, \\ m &= \frac{X}{4p_X} \approx \frac{1}{4p_Z} \sqrt{b} W(\sqrt{b}) \approx \frac{1}{4p_Z} \sqrt{b} \ln \sqrt{b}, \end{aligned} \quad (38)$$

with aspect ratio $m/n = \sqrt{b}$. As the bias increases with p_Z fixed, the code size creeps up slowly in the X -protection direction, and more rapidly in the Z -protection direction.

For the optimal code the probability of an odd number of X errors in a row decreases as b increases according to

$$\text{OddProb}(p_X, m) \approx \frac{1}{2} (1 - e^{-2p_X m}) \approx p_X m \approx \frac{1}{4} X \approx \frac{1}{4} \frac{W(\sqrt{b})}{\sqrt{b}} \approx \frac{1}{4} \frac{\ln \sqrt{b}}{\sqrt{b}}, \quad (39)$$

while the probability of an odd number of Z errors in a column asymptotically approaches $1/2$ according to

$$\text{OddProb}(p_Z, n) \approx \frac{1}{2} (1 - e^{-2p_Z n}) = \frac{1}{2} (1 - e^{-Z/2}) \approx \frac{1}{2} (1 - \sqrt{X}), \quad (40)$$

or

$$\frac{1}{2} - \text{OddProb}(p_Z, n) \approx \frac{1}{2} \left(\frac{W(\sqrt{b})}{\sqrt{b}} \right)^{1/2}. \quad (41)$$

The optimal failure probability is much smaller for $b \gg 1$ than in the case $b = 1$ for the same value of p_Z , but the price paid is that the block size is correspondingly significantly larger:

$$mn = \frac{\sqrt{b} W^2(\sqrt{b})}{16p_Z^2}, \quad (42)$$

as compared to

$$n^2 = \frac{\ln^2 2}{16p^2} \quad (43)$$

in the unbiased case.

Using Eq.(38), we can now evaluate the subleading terms in Eq.(29), finding

$$\begin{aligned} -\frac{m}{4} \exp(-8p_Z n) &\approx -\frac{m}{4} X^2 \approx -\frac{1}{4} \left(\frac{\sqrt{b} W(\sqrt{b})}{4p_Z} \right) \left(\frac{W(\sqrt{b})}{\sqrt{b}} \right)^2 = -\frac{W^3(\sqrt{b})}{16p_Z \sqrt{b}}, \\ -p_X mn &\approx -\frac{p_Z}{b} \left(\frac{\sqrt{b} W(\sqrt{b})}{4p_Z} \right) \left(\frac{W(\sqrt{b})}{4p_Z} \right) = -\frac{W^2(\sqrt{b})}{16p_Z \sqrt{b}}. \end{aligned} \quad (44)$$

Aside from logarithmic factors, these terms are suppressed by $O(b^{-1/2} \ln b)$ compared to the leading terms in Eq.(29). We therefore expect our estimate for the optimal code dimensions to be accurate for $b \gg 1$. However, although these corrections are small compared to the leading terms, they are not necessarily negligible, and in particular they can become important as

$p_Z \rightarrow 0$ with b fixed. To obtain accurate results for small p_Z , we should resum the power series expansion in Eq.(29), or in other words use the full expressions in Eq.(28), with the appropriate values of m , n , Z , and X plugged in. Thus,

$$\begin{aligned} BSZOptimalFailProb(p_Z, b) &\approx (1 - e^{-Z})^{m/2} \approx \left(1 - \frac{W(\sqrt{b})}{\sqrt{b}}\right)^{\frac{\sqrt{b}W(\sqrt{b})}{8p_Z}}, \\ BSXOptimalFailProb(p_Z, b) &\approx (1 - e^{-X})^{n/2} \approx \left(1 - \exp\left(-e^{-W(\sqrt{b})}\right)\right)^{\frac{W(\sqrt{b})}{8p_Z}}, \end{aligned} \quad (45)$$

provide more accurate approximations than $\exp(-W^2(\sqrt{b})/8p_Z)$.

Plugging our solutions for Z and X back into the expressions for the prefactors in Eq.(19) and Eq.(20), we again find somewhat different scaling with the bias for the two failure probabilities. For the Z failure probability the prefactor is

$$\frac{1}{\sqrt{2\pi m}} \exp\left[\frac{2p_Z^2 mn}{e^{4p_Z n} - 1}\right] (e^{4p_Z n} - 1)^{1/2} = \left(\frac{2p_Z n}{\pi Z m}\right)^{1/2} \exp\left[\frac{\frac{1}{8}Z^2 \frac{m}{n}}{e^Z - 1}\right] (e^Z - 1)^{1/2}; \quad (46)$$

making the approximation $e^Z - 1 \approx e^Z$ as before, and using $m/n = \sqrt{b} = Ze^Z$, this becomes

$$\approx \left(\frac{2p_Z}{\pi}\right)^{1/2} b^{-1/4} Z^{-1/2} e^{Z^3/8} e^{Z/2} = \left(\frac{2p_Z}{\pi}\right)^{1/2} Z^{-1} e^{Z^3/8}. \quad (47)$$

For the X failure probability the prefactor is

$$\frac{1}{\sqrt{2\pi n}} \exp\left[\frac{2p_X^2 mn}{e^{4p_X m} - 1}\right] (e^{4p_X m} - 1)^{1/2} = \left(\frac{2p_Z}{\pi Z}\right)^{1/2} \exp\left[\frac{\frac{1}{8}X^2 \frac{n}{m}}{e^X - 1}\right] (e^X - 1)^{1/2}; \quad (48)$$

making the approximation $e^X - 1 \approx X$ as before, this becomes

$$\approx \left(\frac{2p_Z}{\pi}\right)^{1/2} Z^{-1/2} \exp\left(\frac{X}{8\sqrt{b}}\right) X^{1/2} = \left(\frac{2p_Z}{\pi}\right)^{1/2} b^{-1/4} \exp\left(\frac{X}{8\sqrt{b}}\right) \approx \left(\frac{2p_Z}{\pi}\right)^{1/2} b^{-1/4}. \quad (49)$$

We also note that, as in Eq.(17), there are corrections to the factor $e^{Z^3/8}$ of the form

$$\exp\left(\frac{1}{8}Z^3 [1 + O(p_Z) + O(p_Z Z)]\right). \quad (50)$$

In fact we can sum up these corrections to all orders in $p_Z Z$ (still neglecting corrections suppressed by further powers of p_Z) to obtain the improved approximation

$$\exp\left(\frac{1}{8p} Z^2 (1 - e^{-pZ}) [1 + O(p_Z)]\right). \quad (51)$$

This correction can be significant for $pZ = O(1)$, but in that case our bound on the failure probability is rather loose anyway (see below), so including the correction is not so important.

Combining the estimated prefactors with Eq.(45) we obtain the formulas for the failure probabilities:

$$\begin{aligned} BSZOptimalFailProb(p_Z, b) &\approx \left(\frac{2p_Z}{\pi}\right)^{1/2} Z^{-1} e^{Z^3/8} \left(1 - \frac{Z}{\sqrt{b}}\right)^{\frac{Z\sqrt{b}}{8p_Z}}, \\ BSXOptimalFailProb(p_Z, b) &\approx \left(\frac{2p_Z}{\pi}\right)^{1/2} b^{-1/4} (1 - \exp(-e^{-Z}))^{\frac{Z}{8p_Z}}, \end{aligned} \quad (52)$$

where $Z = W(\sqrt{b})$. These asymptotic formulas apply if we fix the bias b at a large value and then allow p_Z to become sufficiently small. We see that the nonleading contribution to the log of the failure probability is approximately $\frac{1}{8}Z^3$ for Z -errors and approximately $-\frac{1}{4}\ln b$ for X errors. These nonleading terms are small compared to the leading term $\frac{1}{8p_Z}Z^2$, which we minimized to obtain our estimates, provided $p_Z Z \approx p_Z \ln b \ll 1$. A numerical fit indicates that the leading correction to our asymptotic formula is approximately $(.01)p_Z(\ln b)^\delta$ where $\delta \approx 4$. When this correction is small, we expect our estimate of the optimal failure probability to be reasonably tight.

4 Comparison with numerics

To check the accuracy of our formulas, we have also numerically determined the optimal values of m and n , and the corresponding optimal failure probability, for a variety of values of p_Z and b .

Fig. 2 shows the numerically optimized failure probability and the asymptotic estimate Eq.(23) as a function of $p \equiv p_Z = p_X$ for the unbiased case ($b = 1$). Our formula overestimates the exact result by less than 1% for $p < .001$ and by less than 10% for $p < .01$.

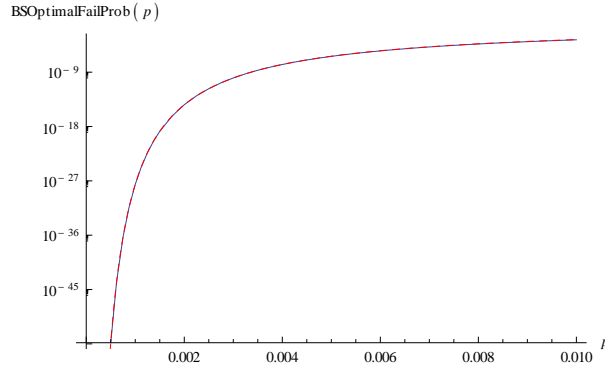


Fig. 2. The optimal failure probability for the Bacon-Shor code (solid blue curve) and the estimate Eq.(23) (dashed red curve) for unbiased noise, as a function of $p \equiv p_Z = p_X$. The curves nearly coincide.

Fig. 3 shows the numerically optimized failure probability and the estimate Eq.(52) as a function of the bias b for $p_Z = .01$. Here we have actually plotted the total error probability assuming X and Z errors are independent; that is,

$$\begin{aligned} BSOptimalFailProb &= BSOptimalZFailProb + BSOptimalXFailProb \\ &\quad - BSOptimalZFailProb \times BSOptimalXFailProb. \end{aligned} \quad (53)$$

The agreement is good for $b > 10$, though the discrepancy grows with increasing b , reaching about 10% for $b = 500$.

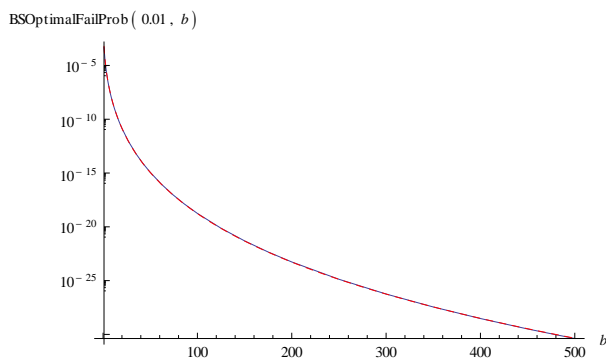


Fig. 3. The optimal failure probability for the Bacon-Shor code (solid blue curve) and the estimate Eq.(52) (dashed red curve) for $p_Z = .01$, as a function of the bias $b = p_Z/p_X$. The curves nearly coincide.

Fig. 4 shows the numerically optimized failure probability and the estimate Eq.(52) as a function of the bias b for $p_Z = .03$. Now our formula badly overestimates the failure probability for large b , with the discrepancy reaching a factor of 13 for $b = 5000$. In this regime, the condition $(.01)p_Z \ln^4 b \ll 1$ is not well satisfied ($(.01)p_Z \ln^4 b \approx 1.6$ for $b = 5000$).

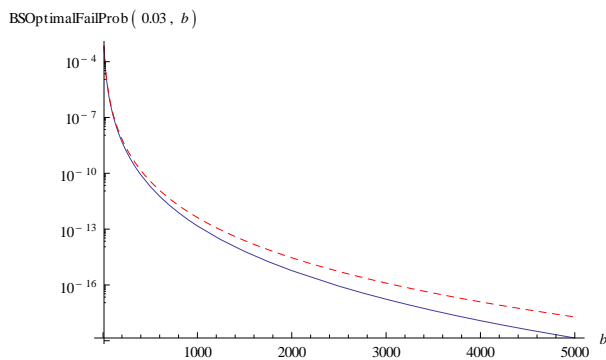


Fig. 4. The optimal failure probability for the Bacon-Shor code (solid blue curve) and the estimate Eq.(52) (dashed red curve) for $p_Z = .03$, as a function of the bias $b = p_Z/p_X$.

Fig. 5 shows the numerical and analytic results as a function of p_Z for $b = 1000$, and Fig. 6 shows the ratio of the two, illustrating that the agreement improves rapidly as p decreases.

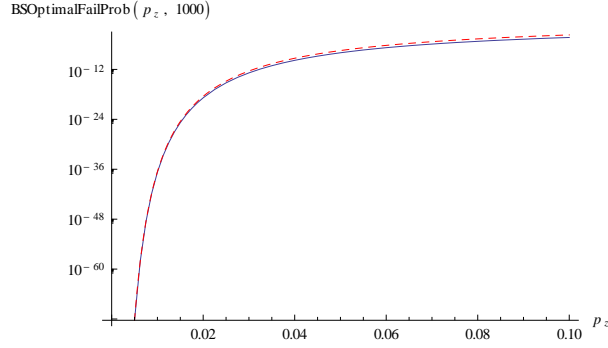


Fig. 5. The optimal failure probability for the Bacon-Shor code (solid blue curve) and the estimate Eq.(52) (dashed red curve) for $b = 1000$, as a function of the error probability p_Z .

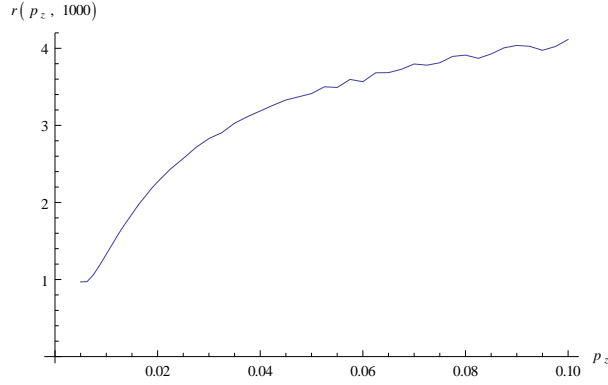


Fig. 6. The ratio of the optimal failure probability to the estimate Eq.(52) for $b = 1000$, as a function of the error probability p_Z . The kinks in the plot occur because the optimal dimensions $m \times n$ of the Bacon-Shor code are integers which change discontinuously.

5 2D Bacon-Shor recovery with syndrome measurement errors

Measurement of the weight-two gauge operator XX can be executed by a simple circuit with four locations — e.g. preparation of an ancilla qubit in the $X = 1$ eigenstate $|+\rangle$, two successive controlled- X gates with ancilla qubit as control and data qubit as target, followed by measurement of the ancilla qubit in the X basis. A similar circuit measures ZZ , but with the controlled- X gates replaced by controlled- Z gates.

A fault in the circuit that flips the ancilla qubit could cause the measurement outcome to be incorrectly recorded, and a faulty two-qubit gate could damage a qubit in the code block. However, a single fault in the circuit will not cause two errors in the code block that cannot be gauged away. A Z -type syndrome bit is obtained by computing the parity of n outcomes of XX gauge qubit measurements, and an error in any one of those n measurements could flip the bit, so that the probability of an error in a syndrome bit is roughly n times larger than the probability of error in each gauge qubit measurement. Similarly, an X -type syndrome bit is obtained by computing the parity of m outcomes of ZZ gauge qubit measurements, so the probability of an error in a syndrome bit is roughly m times the probability of error in each

gauge qubit measurement.

Though measurement and data errors may actually be correlated, these considerations motivate an idealized model for Bacon-Shor error recovery with noisy measurements. For the Z -type syndrome we consider a repetition code of length m , where both the probability of error per data bit and the probability of error per syndrome bit is $\tilde{p}_Z = \text{OddProb}(p_Z, n)$, and p_Z is the fixed physical error rate for a Z -type error. Likewise, for the X -type syndrome we consider a repetition code of length n , where both the probability of error per data bit and the probability of error per syndrome bit is $\tilde{p}_X = \text{OddProb}(p_X, m)$, and p_X is the fixed physical error rate for an X -type error.

We will study this model for both unbiased noise and highly biased noise with $b = p_Z/p_X \gg 1$. Justifying highly biased noise models raises subtle issues — why should the nontrivial quantum gates used in the syndrome measurement circuit strongly favor Z errors over X errors [10, 9]? We will not address these issues here; rather we shall be satisfied to say that a model in which both the data qubit errors and the syndrome bit errors are highly biased is mathematically natural and worthy of investigation. Ref. [9] contains a much more complete discussion of fault-tolerant error correction for Bacon-Shor codes.

A nearly optimal recovery scheme for a repetition code with syndrome errors was described in [11, 12]. To obtain more reliable syndrome information we follow the history of the syndrome through many measurement cycles, and assume that the probabilities for both data errors and syndrome errors are given by \tilde{p}_Z, \tilde{p}_X in each round of syndrome measurement. The syndrome history can be represented on a two-dimensional square lattice in spacetime, where each horizontal row of squares records the results from one cycle of (possibly noisy) syndrome measurement and each square represents a data bit — a marked vertically oriented link in the row indicates that the two data bits sharing that link were found to have opposite parity. We identify all the boundary points of this chain of nontrivial syndrome bits, and apply the Edmonds matching algorithm to find the minimum-weight chain in spacetime with these boundary points (relative to the boundary of the sample). This minimum-weight chain identifies the most likely error history compatible with the observed syndrome history, where vertical edges in the chain are hypothetical syndrome measurement errors and horizontal edges are hypothetical data errors. The actual error chain combined with the hypothetical chain comprises a closed chain relative to the boundary, and a segment of this chain stretching across the sample, connecting its left and right boundary, signifies a logical error. If we consider the repetition code on a circle rather than an open one-dimensional lattice, so there is no sample boundary, then a closed chain wrapping around the cylinder indicates a logical error.

This scheme has been studied previously [11, 12] in the case where the probability p of a data error or syndrome error is a constant independent of the length m of the repetition code; in that case recovery is successful with a probability approaching 1 as $m \rightarrow \infty$ for $p < 10.3\%$ using the matching algorithm, and for $p < 11.0\%$ using the optimal algorithm. Now we wish to reconsider the efficacy of the matching algorithm in the case where the error rate scales nontrivially (roughly linearly) with m .

Based on our earlier results for the case of ideal syndrome measurement, we anticipate that if m is chosen optimally then the probability per unit time of an encoded error in the

Bacon-Shor code has the form

$$\ln[BSOptimalFailRate(p_Z, b)] \approx -\tilde{A}(b)/p_Z + \tilde{B}(b) \ln p_Z + \tilde{C}(b) + \dots, \quad (54)$$

but with different functions of the noise bias $\tilde{A}(b), \tilde{B}(b), \tilde{C}(b)$ than in the case of ideal measurements. We could try to estimate these functions using a Monte Carlo method, in which we generate sample error histories, infer the syndrome, find the minimum weight matching of the syndrome's boundary points, and then determine whether a logical error has occurred. This method is difficult to carry out, however, because for small p_Z the optimal failure probability is quite small, and we need to generate many samples to estimate it with reasonable statistical accuracy. Instead, we will use an analytic argument to obtain a rather loose upper bound on \tilde{A} (which dominates the scaling of the failure probability when p_Z is small) for both the case of unbiased noise ($b \equiv p_Z/p_Z = 1$), and the case of highly biased noise ($b \equiv p_Z/p_X \gg 1$).

Since we expect the value of \tilde{A} to be the same for a planar Bacon-Shor code as for a code defined on a torus, we will consider the case of a torus for convenience. Thus we consider the syndrome history (for either the X or Z errors) on a cylinder, closed in the spatial direction but open in the temporal direction. If there is a logical error, then the combination of the actual error chain and the hypothetical error chain must contain a self-avoiding cycle that wraps once around the cylinder, where at least half of the edges in this cycle have actual errors (otherwise there would a lower-weight choice for the hypothetical chain). If we fix a cycle of length r , then the probability that a particular set of s edges in the cycle have actual errors, while the remaining $r - s$ edge do not, is

$$\tilde{p}^s (1 - \tilde{p})^{r-s} = [\tilde{p}(1 - \tilde{p})]^{r/2} [\tilde{p}/(1 - \tilde{p})]^{s-r/2} \leq [\tilde{p}(1 - \tilde{p})]^{r/2} \quad (55)$$

where the inequality holds provided $\tilde{p} \leq 1 - \tilde{p}$ and $s \geq r/2$. The number $\binom{r}{s}$ of ways to place s errors on the cycle is bounded above by 2^r , therefore, the probability of failure arising from this particular length r cycle is bounded above by $[4\tilde{p}(1 - \tilde{p})]^{r/2}$.

Considering the Z error syndrome history for definiteness, we now let $SAC(r, m)$ denote the number of distinct self-avoiding cycles of length r that wrap around the cylinder of circumference m . Here we are not counting the freedom to translate the cycle in either space or time. Taking this freedom into account, we obtain an upper bound on the probability of error per unit time

$$BSZFailRate(p_Z, m, n) \leq m \sum_{r=m}^{\infty} SAC(r, m) [4\tilde{p}_Z(1 - \tilde{p}_Z)]^{r/2}. \quad (56)$$

The factor of m in front arises from the m possible spatial translations of the cycle, and a factor of time T arising from time translations has been divided out to obtain a failure rate per unit time. Note that the cycle length is in principle unbounded, as it could extend indefinitely in the time direction.

The number $SAC(r, m)$ of self-avoiding cycles of length r on the cylinder is bounded above by the number $SAW(r)$ of self-avoiding open walks with a specified starting point, for which an upper bound is known of the form [13, 14]

$$SAW(r) \leq \gamma r^\beta \mu^r, \quad (57)$$

where $\mu \approx 2.6381585$, $\beta = 11/32$, and $\gamma \approx 1.17704$. Plugging the upper bound on $SAW(r)$ into our expression for $BSZFailRate$, we find

$$\begin{aligned}
BSZFailRate(p_Z, m, n) &\leq \gamma m \sum_{r=m}^{\infty} r^{\beta} \mu^r [4\tilde{p}_Z(1 - \tilde{p}_Z)]^{r/2} \\
&\leq \gamma m m^{\beta} [4\mu^2 \tilde{p}_Z(1 - \tilde{p}_Z)]^{m/2} \sum_{s=m}^{\infty} \left(\frac{m+s}{m}\right)^{\beta} [4\mu^2 \tilde{p}_Z(1 - \tilde{p}_Z)]^{s/2} \\
&\leq \gamma m^{\beta+1} [4\mu^2 \tilde{p}_Z(1 - \tilde{p}_Z)]^{m/2} \sum_{s=0}^{\infty} e^{\beta s/m} [4\mu^2 \tilde{p}_Z(1 - \tilde{p}_Z)]^{s/2} \\
&= \gamma m^{\beta+1} [4\mu^2 \tilde{p}_Z(1 - \tilde{p}_Z)]^{m/2} \frac{1}{1 - \sqrt{4\mu^2 e^{2\beta/m} \tilde{p}_Z(1 - \tilde{p}_Z)}}. \quad (58)
\end{aligned}$$

We will obtain our upper bound on the failure rate by choosing m and n such that $4\mu^2 e^{2\beta/m} \tilde{p}_Z(1 - \tilde{p}_Z)$ is strictly less than one; therefore we can bound the last factor by a constant, obtaining

$$\ln[BSZFailRate(p_Z, m, n)] \leq \frac{m}{2} \ln[4\mu^2 \tilde{p}_Z(1 - \tilde{p}_Z)] + O(\ln m), \quad (59)$$

and by similar reasoning

$$\ln[BSXFailRate(p_X, m, n)] \leq \frac{n}{2} \ln[4\mu^2 \tilde{p}_X(1 - \tilde{p}_X)] + O(\ln n). \quad (60)$$

5.1 Unbiased noise

In the case of unbiased noise, where $p_Z = p_X \equiv p$, we choose $n = m$ and recall that $\tilde{p} = \frac{1}{2}(1 - (1 - 2p)^m) \approx \frac{1}{2}(1 - e^{-2pm})$, so that

$$4\tilde{p}(1 - \tilde{p}) \approx 1 - e^{-4pm}. \quad (61)$$

Thus we conclude that the failure rate for either Z -type or X -type errors is

$$\ln[BSFailRate(p, m)] \leq \frac{m}{2} \ln[\mu^2(1 - e^{-4pm})] + O(\ln m). \quad (62)$$

To find the value of m that minimizes this upper bound (for asymptotically large m), we solve

$$0 = \frac{1}{2} \left(\ln[\mu^2(1 - e^{-4pm})] + \frac{4pm}{e^{4pm} - 1} \right) + O(1/m), \quad (63)$$

for $\mu = 2.6381585$, finding $pm \approx 0.0139682$, so that

$$\ln[BSOptimalFailRate(p)] \leq -\frac{.00679079}{p} + O(\ln p). \quad (64)$$

Thus we derive a lower bound

$$\tilde{A} \geq .00679079, \quad (65)$$

which is about an order of magnitude smaller than the value of A found for the case of ideal syndrome measurement.

Even more simply, we may observe that $\tilde{p} \leq pm$ and $\tilde{p}(1 - \tilde{p}) \leq \tilde{p}$ to obtain

$$\ln[BSFailRate(p, m)] \leq \frac{m}{2} \ln(4\mu^2 pm) + O(\ln m), \quad (66)$$

which is optimized by choosing $4\mu^2 pm = 1/e$, so that

$$\ln[BSoptimalFailRate(p)] \leq -\frac{1}{8\mu^2 ep} + O(\ln p) = -\frac{.00660714}{p} + O(\ln p), \quad (67)$$

which provides nearly as good a lower bound on \tilde{A} .

5.2 Highly biased noise

In the case of biased noise, again using the approximations $\tilde{p}_Z(1 - \tilde{p}_Z) \leq p_Z n$ and $\tilde{p}_X(1 - \tilde{p}_X) \leq p_X m$, the leading contributions to the Z and X failure rates are

$$\begin{aligned} \ln[BSZFailRate(p_Z, m, n)] &\approx \frac{m}{2} \ln(4\mu^2 p_Z n) = \frac{1}{8\mu^2 p_X} X \ln Z = \frac{1}{8\mu^2 p_Z} bX \ln Z, \\ \ln[BSXFailRate(p_X, m, n)] &\approx \frac{n}{2} \ln(4\mu^2 p_X m) = \frac{1}{8\mu^2 p_Z} Z \ln X, \end{aligned} \quad (68)$$

where

$$X = 4\mu^2 p_X m, \quad Z = 4\mu^2 p_Z n, \quad b = \frac{p_Z}{p_X}. \quad (69)$$

(Actually, the approximation $p_Z(1 - \tilde{p}_Z) \approx p_Z n$ is not necessarily so accurate when m and n are chosen optimally, but it suffices for deriving an upper bound on the failure rate; if we do not make this approximation the analysis becomes more complicated and the upper bound improves by only about 8%.) Equating the upper bounds on the X and Z failure rates, we are to minimize

$$F(X, Z) = Z \ln X \quad (70)$$

subject to the constraint

$$Z \ln X = bX \ln Z. \quad (71)$$

Introducing a Lagrange multiplier λ to do the constrained minimization, we obtain the equations

$$\begin{aligned} (1 + \lambda) \ln X &= \lambda b \frac{X}{Z}, \\ (1 + \lambda) \frac{Z}{X} &= \lambda b \ln Z, \end{aligned} \quad (72)$$

which imply

$$\ln X = \frac{1}{\ln Z}. \quad (73)$$

Using the constraint Eq.(71), we see that the aspect ratio of the optimal code is

$$\frac{m}{n} = b \frac{X}{Z} = \frac{\ln X}{\ln Z} = \ln^2 X. \quad (74)$$

When the bias b is very large, the optimal value of Z (that is, when m and n are chosen to optimize the approximate expression Eq.(68), an upper bound on the actual failure rate) approaches one from below, while $4\mu^2 e^{2\beta/m} \tilde{p}_Z(1 - \tilde{p}_Z)$ remains strictly less than one for m large, so we can still bound the last factor in Eq.(58) by a constant. To estimate the failure probability, we first determine Z by solving Eq.(71) in the form

$$Z = b \ln^2 Z \exp(1/\ln Z). \quad (75)$$

Then we find X by substituting this value of Z back into Eq.(71), finding

$$\frac{b}{Z} = \frac{\ln^2 X}{X} \Rightarrow \frac{1}{\sqrt{X}} \ln \left(\frac{1}{\sqrt{X}} \right) = \sqrt{\frac{b}{4Z}}. \quad (76)$$

Recalling that the Lambert W function $W(z)$ is defined by $We^W = z$, we conclude that

$$\ln \left(\frac{1}{\sqrt{X}} \right) = W \left(\sqrt{\frac{b}{4Z}} \right). \quad (77)$$

Thus we find that the value of F at its minimum is

$$F_{\min} = Z \ln X \approx -2Z W \left(\sqrt{\frac{b}{4Z}} \right), \quad (78)$$

and we obtain upper bounds on the leading contributions to the failure rates:

$$\ln[BSZF\text{ailRate}(p_Z, m, n)] \approx \ln[BSXF\text{ailRate}(p_X, m, n)] \approx -\frac{Z(b)}{4\mu^2 p_Z} W \left(\sqrt{\frac{b}{4Z(b)}} \right), \quad (79)$$

where $Z(b)$ is a slowly varying monotonic function, ranging from $1/e$ to 1 as b increases from 1 to infinity. Comparing to the upper bound for the case of ideal syndrome measurement we find

$$A/\tilde{A} = \frac{\mu^2}{2Z(b)} \frac{W^2(\sqrt{b})}{W(\sqrt{b/4Z})}, \quad (80)$$

a slowly increasing function of b . The optimal aspect ratio is

$$\frac{m}{n} = \ln^2 X \approx 4W^2 \left(\sqrt{\frac{b}{4Z(b)}} \right); \quad (81)$$

in contrast to the case where the syndrome is perfect, this aspect ratio grows only polylogarithmically with the bias b .

For example, by solving Eq.(75), we find $Z = .3679, .7046, .8428, .8994$ for $b = 1, 10^2, 10^4, 10^6$ and corresponding values of $-F_{\min} = .3679(= 1/e), 2.01255, 4.92966, 8.48426$. For $b = 10^2, 10^4, 10^6$, our values of A/\tilde{A} are 5.231, 11.50, and 18.29 respectively.

6 Higher-dimensional Bacon-Shor codes

The two-dimensional Bacon-Shor code can be extended to higher dimensions. Bacon [1] discussed one such extension to three dimensions. In Bacon’s code, defined on an $L \times L \times L$ cubic lattice (for L odd), the logical operators \bar{X} and \bar{Z} are both weight L^2 operators defined on orthogonal planes, and a length L repetition code protects against both X and Z errors. Here we will briefly discuss a different way to extend the code to three (or more) dimensions, in which the logical operator \bar{Z} resides on a horizontal plane and the logical operator \bar{X} resides on a vertical line. This alternative formulation is less symmetric but in some ways more natural; it may be more effective than Bacon’s code if the noise is highly biased, and it is more robust against errors in the measurement of the Z -type error syndrome

We consider an $m \times n \times k$ cubic lattice, and choose the gauge algebra to be generated by two-qubit operators XX acting on pairs of neighboring qubits in the same horizontal plane, and by two-qubit operators ZZ acting on pairs of qubits in the same vertical column. There is one protected qubit; the weight- mn operator \bar{Z} acts on all qubits in a horizontal plane and the weight- k operator \bar{X} acts on all the qubits in a vertical column. The stabilizer group has $mn - 1$ independent X -type generators, each a product of $2k$ X s acting on all the qubits in a pair of neighboring vertical columns, and $k - 1$ independent Z -type generators, each a product of $2mn$ Z s acting on all the qubits in a pair of neighboring horizontal planes. As in the 2D code, the value of a check operator can be found by measuring weight-two gauge operators and combining the results. For example, the weight- $2mn$ product of Z s in two neighboring planes is obtained by measuring mn gauge operators ZZ , each acting on two qubits in the same column, and multiplying the results together.

As in the 2D case, the error recovery procedure is clarified if we adopt a suitable gauge. An even number of Z errors in a column can be gauged away, and if there are an odd number of Z errors in a column, there is a gauge such that all errors but one are removed, and the single error lies in the uppermost horizontal plane. Similarly, an even number of X errors in a plane can be gauged away, and if there are an odd number of X errors in a plane, all but one can be removed, with the single error lying in the northwest corner of the plane. The check operators, whose value can be inferred from measurements of many weight-two gauge operators, check whether qubits in the northwest corner of neighboring planes agree in the Z basis, and whether qubits in the top plane of neighboring columns agree in the X basis.

The planar repetition code corrects $(mn - 1)/2$ Z errors in the plane, and the columnar repetition code corrects $(k - 1)/2$ X errors in the column. Aside from its greater length (assuming $mn > k$), the planar code has another advantage over the columnar code — measuring all local X -type gauge operators determines the error syndrome redundantly, which improves the robustness against syndrome measurement errors. From the gauge qubit measurements, we determine the value of $X^{\otimes 2k}$ for any two neighboring columns; there are $m(n - 1)$ such pairs of “east-west” neighbors as well as $n(m - 1)$ pairs of “north-south” neighbors, for a total of $2mn - m - n$ check bits, or which only $mn - 1$ are independent.

Recovery from Z errors succeeds if the number of vertical columns with an odd number of Z errors is no more than $(mn - 1)/2$. and recovery from X errors succeeds if the number of horizontal planes with an odd number of X errors is no more than $(k - 1)/2$. Setting $m = n$, therefore, the probability of a logical Z error in an independent error model (assuming perfect

syndrome measurements) is

$$BS3DZFailProb(p_Z, m, k) = RepFailProb(OddProb(p_Z, k), m^2) \quad (82)$$

where p_Z is the Z error rate, and the probability of a logical X error is

$$BS3DXFailProb(p_X, m, k) = RepFailProb(OddProb(p_X, m^2), k) \quad (83)$$

where p_X is the X error rate. These are the same functions we encountered in our analysis of the 2D code, except with m replaced by m^2 and n by k .

In four dimensions, denote the four directions by x, y, z, w . To define the Bacon-Shor code, choose the gauge algebra to be generated by XX acting on neighboring qubits in each xy -plane (with z and w fixed) and by ZZ acting on neighboring qubits in each zw -plane (with x and y fixed). Note that each xy -plane intersects exactly once with each zw -plane. The logical \bar{Z} is the product of all Z s in an xy -plane, and the logical \bar{X} is the product of all X s in a zw -plane. For a suitable gauge choice, all X errors can be moved to the zw -plane with $x = y = 0$ and all Z errors can be moved to the xy -plane with $z = w = 0$. Hence for this four-dimensional Bacon-Shor code a two-dimensional repetition code protects against logical Z errors and another two-dimensional repetition code protects against logical X errors. If we choose the dimensions of the hypercubic lattice to be $m \times m \times n \times n$, then the probability of a logical Z error (assuming perfect syndrome measurement) in an independent noise model is

$$BS4DZFailProb(p_Z, m, n) = RepFailProb(OddProb(p_Z, n^2), m^2) \quad (84)$$

where p_Z is the physical Z error probability per qubit, and the probability of a logical X error is

$$BS4DXFailProb(p_X, m, n) = RepFailProb(OddProb(p_X, m^2), n^2), \quad (85)$$

where p_X is the physical X error probability per qubit. These are the same functions as for the 2D code, but with m replaced by m^2 and n by n^2 . The advantage of the four-dimensional code over the two-dimensional or three-dimensional code is that measuring the weight-two local gauge operators provides redundant syndrome information (and hence improved robustness against syndrome measurement errors) for both the Z and X errors.

7 Conclusions

We have studied the performance of Bacon-Shor codes against both unbiased and biased noise, concluding that the codes provide excellent protection against logical errors if the error probability per qubit is less than a few tenths of a percent in the unbiased case, and if the dephasing error probability is less than a few percent in the case of highly biased noise, assuming the syndrome is measured perfectly.

Using quantum codes to protect quantum computers from noise raises many thorny problems [5, 6]; in particular, we need to build a set of gadgets that reliably execute logical gates acting on the code space, and in the case of biased noise we need to employ a physical gate set compatible with the bias [10, 9]. We have not discussed these issues here. Instead, our goal has been to gain a better understanding of the properties of the Bacon-Shor code family, without getting bogged down in detailed constructions of fault-tolerant protocols. We expect,

though, that our analytic formulas for the logical failure probabilities of optimal codes will provide helpful guidance for the construction of fault-tolerant schemes.

Acknowledgments

We thank Peter Brooks and Franz Sauer for valuable discussions. This work was supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20165. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC or the U.S. Government. We also acknowledge support from NSF grant PHY-0803371, DOE grant DE-FG03-92-ER40701, NSA/ARO grant W911NF-09-1-0442, Caltech's Summer Undergraduate Research Fellowship (SURF) program, and the Victor Neher SURF Endowment. The Institute for Quantum Information and Matter (IQIM) is an NSF Physics Frontiers Center with support from the Gordon and Betty Moore Foundation.

References

1. D. Bacon, Operator quantum error-correcting subsystems for self-correcting quantum memories, *Phys. Rev. A* 73, 012340 (2006), arXiv:quant-ph/0506023.
2. P. W. Shor, Scheme for reducing decoherence in quantum computer memory, *Phys. Rev. A* 52, 2493 (1995).
3. D. W. Kribs, R. Laflamme, D. Poulin, and M. Lesosky, Operator quantum error correction, *QIC* 6, 382-399 (2006), arXiv:quant-ph/0504189.
4. D. Poulin, Stabilizer formalism for operator quantum error correction, *Phys. Rev. Lett.* 95, 230504 (2005). arXiv: quant-ph/0508131.
5. P. W. Shor, Fault-tolerant quantum computation, in *Proceedings, 37th Annual Symposium on Foundations of Computer Science*, pp. 56-65 (Los Alamitos, CA, IEEE Press, 1996), arXiv:quant-ph/9605011.
6. D. Gottesman, An introduction to quantum error correction and fault-tolerant quantum computation, arXiv:0904.2557 (2009).
7. P. Aliferis and A. W. Cross, Subsystem fault tolerance with the Bacon-Shor code, *Phys. Rev. Lett.* 98, 220502 (2007), arXiv:quant-ph/0610063.
8. A. W. Cross, D. P. DiVincenzo, and B. M. Terhal, A comparative code study for quantum fault-tolerance, arXiv:0711.1556.
9. P. Brooks and J. Preskill, Fault-tolerant quantum computation with asymmetric Bacon-Shor codes, unpublished (2012).
10. P. Aliferis and J. Preskill, Fault-tolerant quantum computation against biased noise, *Phys. Rev. A* 78, 052331 (2008), arXiv:0710.1301.
11. E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Topological quantum memory, *J. Math. Phys.* 43, 4452-4505 (2002), arXiv:quant-ph/0110143.
12. C. Wang, J. Harrington, and J. Preskill, Confinement-Higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory, *Annals of Physics* 303, 31-58 (2003), arXiv:quant-ph/0207088.
13. N. Madras and G. Slade, *The Self-Avoiding Walk* (Birkhauser, 1996).
14. I. Jensen, Enumeration of self-avoiding walks on the square lattice, *J. Phys. A* 37, 5503-5524 (2004), arXiv:cond-mat/0404728.