Quantum Information and Computation, Vol. 13, No. 3&4 (2013) 0195–0209
© Rinton Press

MAGIC-STATE DISTILLATION WITH THE FOUR-QUBIT CODE

ADAM M. MEIER[a]
Physics Department, University of Colorado at Boulder, 390 UCB
Boulder, CO 80309-0390

BRYAN EASTIN
Northrop Grumman Coproration
Baltimore, MD

EMANUEL KNILL
Information Technology Laboratory, National Institute of Standards and Technology, 325 Broadway
Boulder, CO 80305

Received May 11, 2012
Revised October 1, 2012

The distillation of magic states is an often-cited technique for enabling universal quantum computing once the error probability for a special subset of gates has been made negligible by other means. We present a routine for magic-state distillation that reduces the required overhead for a range of parameters of practical interest. Each iteration of the routine uses a four-qubit error-detecting code to distill the +1 eigenstate of the Hadamard gate at a cost of ten input states per two improved output states. Use of this routine in combination with the 15-to-1 distillation routine described by Bravyi and Kitaev allows for further improvements in overhead.

Keywords: magic-state, distillation, Clifford, four-qubit code

Communicated by: S Braunstein & R Laflamme

1 Introduction

Many techniques for robustly implementing quantum gates most naturally generate only a finite subset of the unitary operators. Frequently, the naturally convenient quantum operations generate the full set of Clifford operations, which consists of the Clifford group of unitaries augmented by measurement and state preparation in the standard basis. Clifford operations are sufficient for stabilizer-state preparations and measurements and thus underlie stabilizer-based error correction and much of the associated theory of fault tolerance. Though inadequate for universal quantum computing, the Clifford operations can be supplemented by any unitary outside of the Clifford group to obtain a universal set [1]. Consequently, the problem of achieving universality is often reduced to that of finding a way of robustly implementing a single non-Clifford unitary gate.

Given the ability to perform Clifford operations, non-Clifford gates can be indirectly implemented using certain non-stabilizer states as a consumable resource. The advantage of

[a]meieram@colorado.edu

this approach lies in the possibility of distilling such resource states prior to use. Distillation is a technique whereby a collection of independently prepared faulty resource states can be converted into a smaller number of resource states whose fidelity with respect to the ideal state is higher. Some states have the property that one can distill them using only Clifford operations. States that are both sufficient for universality and distillable in this way are known as *magic states*. Magic-state distillation allows faulty magic states to be used as a resource for robust universal quantum computing.

The notion of magic states was introduced by Bravyi and Kitaev [2], who showed that the (magic) eigenstates of the one-qubit Clifford gates T and H can be distilled from copies of these states with error probabilities of up to 0.173 and 0.141 per state, respectively. Their distillation routines work by projecting several such faulty copies of a specified magic state (henceforth, resource state) into a stabilizer code and then decoding the result, checking for and discarding on any indication of error. Distillation of the T-eigenstate $|T\rangle$ employs a projection onto the 5-qubit distance-3 code, while distillation of the H-eigenstate $|H\rangle$ relies on the 15-qubit Reed-Muller code; both distillation routines result in one improved resource state. We refer to the $|H\rangle$-distillation routine as the 15-to-1 routine.

An apparently distinct routine for distilling $|H\rangle$ using the 7-qubit Steane code was proposed previously by Knill [3], but Reichardt found the two routines to be equivalent [4]. Reichardt additionally showed that the error threshold for distilling $|H\rangle$ could be improved from 0.141 to 0.146 via a 7-to-1 distillation routine, thereby proving that every faulty $|H\rangle$ outside of the set of stabilizer states is distillable with a finite routine. Campbell and Browne proved the impossibility of a similar result for $|T\rangle$ by showing that no finite distillation routine is capable of distilling faulty $|T\rangle$ arbitrarily near the boundary of the stabilizer states [5, 6].

The focus of each of the aforementioned papers is on the threshold for magic-state distillation, but the efficiency of a distillation routine is crucial to its practical utility. Of particular concern is the number of faulty resource states required as input to distill each resource state of some desired quality. This ratio can contribute strongly to the overhead required to perform a quantum computation. For example, in the large-scale, quantum-dot architecture proposed in Ref. [7], the authors estimate that 90 % of the resources required to implement Shor's algorithm are dedicated to magic-state distillation. Less dramatically, Ref. [8] describes an ion trap architecture wherein magic state distillation consumes 25.5 % of the area in a 32-bit quantum Fourier transform layout. With this overhead in mind, we describe a routine for distilling $|H\rangle$ that reduces the number of input resource states required per output state, distilling 2 improved resource states from 10. The routine can be used either solely or in combination with previously developed routines to obtain resource reductions for a variety of parameter ranges of interest.

After explaining the needed background in Sec. 2, we introduce and analyze the proposed distillation routine in Secs. 3 and 4 and compare it to the 15-to-1 routine in Sec. 6. Sec. 5 explains how sequential distillation rounds can be combined. Concluding remarks appear in Sec. 7.

2 Basic Concepts and Notation

We denote the one-qubit Pauli operators by I, X, Y, and Z, where I is the identity and the others are i times the conventional π-rotations associated with the eponymous axes of the

Bloch sphere. The n-qubit Pauli group consists of all n-fold tensor products of the one-qubit Pauli operators multiplied by $\{\pm 1, \pm i\}$.

The Clifford group consists of the unitary operators that normalize the Pauli group. Any Clifford unitary can be constructed by composition of tensor products of H (Hadamard), T, and ${}^{C}X$ (controlled-NOT or controlled-X) gates up to an unimportant global phase. In the standard basis, these gates are given by

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad T = \frac{e^{i\pi/4}}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix},$$

and

$${}^{C}X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

For convenience, we additionally employ the following one- and two-qubit Clifford gates: $P(\frac{\pm\pi}{2}) = e^{\mp iP\pi/4}$ where $P \in \{X, Y, Z\}$ ($\pi/2$ rotations about the axes of the Bloch sphere), $S = e^{i\pi/4} Z(\frac{\pi}{2})$, ${}^{C}Z$ (controlled-Z), ${}^{C}Y$ (controlled-Y), and $\mathfrak{X}$ (SWAP). We also make frequent use of three unitaries not contained in the Clifford group: $Y(\frac{\pm\pi}{4})$ (the $\pm\pi/4$ rotations about the Y axis) and ${}^{C}H$ (controlled-H).

We use the term "Clifford operation" to refer to any quantum operation that can be implemented using Clifford unitaries together with preparation and measurement in the standard basis. States that can be prepared with Clifford operations are known as stabilizer states. Pure stabilizer states are $+1$ eigenstates of a complete set of commuting (generally multi-qubit) Pauli operators; this set of Pauli operators is known as the stabilizer generator. Similarly, one can define a quantum (stabilizer) code as the $+1$ eigenspace of a non-maximal set of stabilizers.

Whenever necessary, subscripts on operators are used to identify the qubits that they act upon. A bar over an operator indicates that it is a logical (or encoded) operator. That is, it acts as the specified operator on qubits encoded in a quantum code.

Quantum circuit diagrams in this paper conform to the notation used in Ref. [9] with the following exceptions: Wires representing multiple qubits are not specially decorated, and the symbols

are used to represent the ${}^{C}Z$ gate and projective measurements in the eigenbases of X and Z, respectively.

We denote the $+1$ (-1) eigenstate of the Hadamard gate by $|H\rangle$ ($|-H\rangle$), where $|H\rangle = \cos(\frac{\pi}{8})|0\rangle + \sin(\frac{\pi}{8})|1\rangle = Y(\frac{\pi}{4})|0\rangle$. $|H\rangle$ is not a stabilizer state, so Clifford operations are not sufficient for its preparation; however, they are sufficient for its distillation [2]. As shown in Fig. 1(c), $|H\rangle$ can be used together with Clifford operations to implement the non-Clifford gate $Y(\frac{\pi}{4})$, so $|H\rangle$ is a magic state.

Distillation is the process of converting multiple faulty copies of a desired (resource) state into, typically fewer, improved copies of the state. In magic-state distillation, Clifford operations are used to project faulty magic states into a subspace. Given input states of suitable

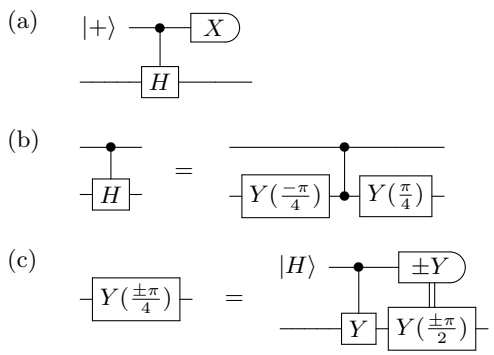

Fig. 1. Circuits showing that the Hadamard operator can be measured non-destructively using only Clifford operations and two copies of $|H\rangle$. The circuit in (a) implements a non-destructive measurement of the Hadamard operator using the non-Clifford ${}^{C}H$ gate, while (b) and (c) give circuit identities that can be used to break the measurement up into Clifford operations and $|H\rangle$ resource states. The classical control in (c) is meant to indicate (for the + case) that a positive Y measurement triggers a $Y(\frac{\pi}{2})$ gate while a negative measurement triggers the identity.

quality and a well-chosen subspace, successful projection allows one to extract higher-fidelity copies of the desired state. When failure to project is detected, the output states are discarded. It is assumed that Clifford operations can be implemented perfectly, which is justified in the commonly considered situation where fault-tolerant techniques provide highly accurate Clifford operations as a matter of course but rely on techniques such as magic-state distillation for universality.

State distillation is facilitated by randomization, which can be used to simplify errors on resource states. For example, the twirling superoperator

$$\mathcal{H}(\rho) = \frac{1}{2}\rho + \frac{1}{2}H\rho H^{\dagger} \,, \tag{1}$$

which can be implemented by applying either I or H with equal probability, decoheres an input state in the eigenbasis of the H operator. For any input state, the resulting state is a probabilistic mixture of $|H\rangle$ and $|-H\rangle$ states. That is, for some $0 \leq p \leq 1$,

$$\mathcal{H}(\rho) = (1-p)|H\rangle\langle H| + p|-H\rangle\langle -H| \,.$$

Because $Y|H\rangle = |-H\rangle$, we can characterize any faulty $|H\rangle$ state that has been twirled with $\mathcal{H}$ as suffering from stochastic Y errors with some probability p that depends on the input state ρ. We assume throughout this paper that resource states are twirled prior to use.

We label distillation routines by their input/output ratios, so an m-to-n distillation routine takes m resource states as input and produces n resource states as output.

3 10-to-2 Distillation Routine

The basic form of our routine for magic-state distillation is as follows: Resource states are encoded into a quantum code; these encoded resource states are verified through an encoded measurement; and finally the code is decoded, leaving, when no errors are indicated, resource states of better quality. The intuition behind this approach is that one would like simply to measure whether a resource state is good, but doing so requires additional resource states

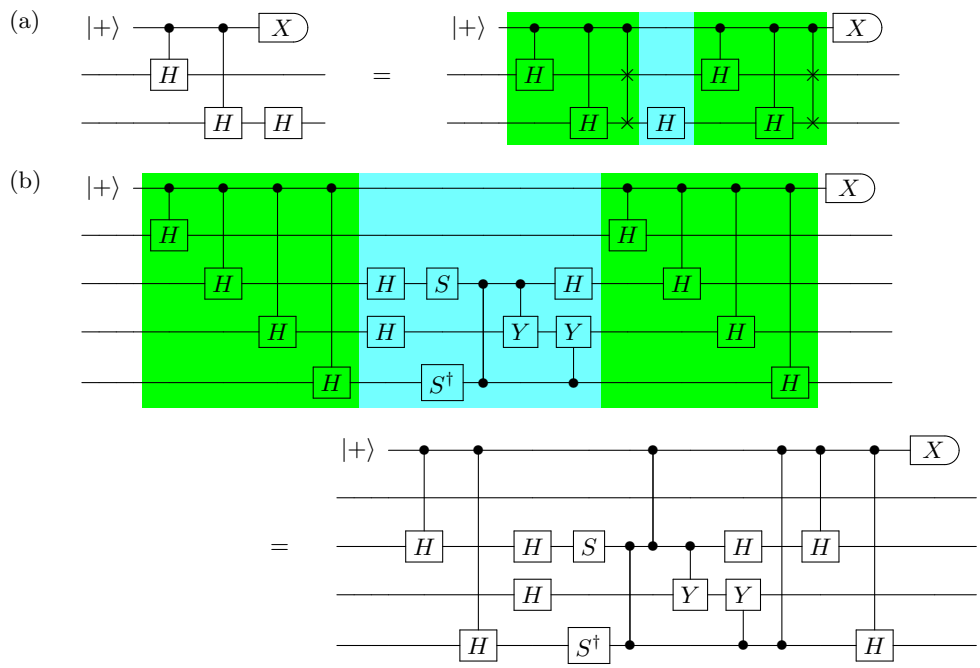

Fig. 2. Circuits that detect whether two input states are in the subspace spanned by $|H\rangle|{-H}\rangle$ and $|{-H}\rangle|H\rangle$, when the input states are (a) unencoded and (b) encoded in the four-qubit code. Corresponding blocks of the circuits in parts (a) and (b) are indicated by shading. The translation from (a) to (b) relies on the fact that, for the four-qubit code, transversal Hadamard effects $\bar{H}_1\bar{H}_2\overline{\mathsf{X}}_{12}$. The equivalence in (b) eliminates four ${}^{C}H$ gates, reducing the number of resource states required by eight. In total, this figure shows that only four ${}^{C}H$ gates are required to project the two qubits encoded in the four-qubit code into either the subspace spanned by $\{|H\rangle|{-H}\rangle, |{-H}\rangle|H\rangle\}$ or that spanned by $\{|H\rangle|H\rangle, |{-H}\rangle|{-H}\rangle\}$. The circuits shown also apply an incidental Hadamard gate to the second logical qubit. Further details are given in Fig. A.1 in the appendix.

whose own errors might go undetected in such a measurement. Errors on these states are rendered detectable by performing an encoded version of the measurement in a fault-tolerant fashion. This is the approach employed in reference [3] for distilling $|H\rangle$ using the 7-qubit Steane code. The routine described here is instead based on the 4-qubit error-detecting code.

As the +1 eigenstate of the Hadamard operator, the state $|H\rangle$ can be verified by measuring H. Measurement of the Hadamard operator is impossible using only Clifford operations, but it can be accomplished, as shown in Fig. 1, with the help of two additional $|H\rangle$ states.

To render errors during the Hadamard measurement detectable, the routine first encodes a pair of faulty resource states into the $\mathcal{C}_4$ code [3] and then performs an encoded measurement $\bar{H}_1\bar{H}_2$ on the pair. This measurement determines whether the pair is in the logical subspace spanned by $|H\rangle|{-H}\rangle$ and $|{-H}\rangle|H\rangle$ and can therefore detect whether one of the states had an error (see Sec. 4).

The $\mathcal{C}_4$ code is a $[[4, 2, 2]]$ quantum code defined by the stabilizer generator matrix:

$$\begin{bmatrix} X \otimes X \otimes X \otimes X \\ Z \otimes Z \otimes Z \otimes Z \end{bmatrix}. \tag{2}$$

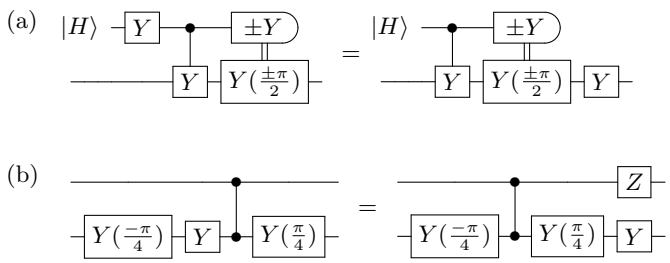

Fig. 3. Circuit identities for propagating Y errors on gate states. (a) The rule for propagating a Y error on a resource state used to apply a $Y(\frac{\pm\pi}{4})$ gate as in Fig. 1(c). The effect of the error is the same as applying the Y error after the gate. (b) A Y error on the first resource state required to implement a ${}^{C}H$ gate using the circuits in Fig. 1 propagates to a Z error on the control and a Y error on the target. An error on the second resource state propagates trivially to a Y error on the target.

Our choices for logical X and Z operators are:

$$\begin{aligned}
\bar{X}_1 &= X \otimes X \otimes I \otimes I, \\
\bar{Z}_1 &= Z \otimes I \otimes I \otimes Z, \\
\bar{X}_2 &= X \otimes I \otimes I \otimes X, \quad \text{and} \\
\bar{Z}_2 &= Z \otimes Z \otimes I \otimes I.
\end{aligned}$$

Because any one-qubit Pauli operator anticommutes with some stabilizer generator of $\mathcal{C}_4$, it is possible to detect any error on a single qubit of the code.

The set of stabilizer generators of $\mathcal{C}_4$ is symmetric with respect to exchange of X and Z, so $H \otimes H \otimes H \otimes H$ is a valid encoded gate, and for the choice of logical Pauli operators given above it effects a logical Hadamard on both encoded qubits followed (or, equivalently, preceded) by a logical `SWAP`. Consequently, the controlled-$(\bar{H}_1\bar{H}_2\bar{\mathtt{X}}_{12})$ gate (the control is unencoded and the target is encoded in $\mathcal{C}_4$) can be accomplished using a sequence of four ${}^{C}H$ gates. Using this gate one can derive a circuit that implements the encoded measurement, $\bar{H}_1\bar{H}_2$, as shown in Fig. 2, by means of four ${}^{C}H$ gates implemented with a total of eight resource states.

The final step of the distillation routine is to use Clifford operations to decode the logical qubits and measure the syndrome of the $\mathcal{C}_4$ code, leaving two output resource states. The routine succeeds and accepts the output if neither the encoded measurement nor the syndrome indicates an error. Otherwise the output is discarded. We analyze the error patterns for the full distillation circuit, shown in Fig. 4(a), in the next section.

4 Analysis

Given perfect Clifford operations and twirled resource states, the only possible errors in our distillation circuit are Y errors on the input resource states. For simplicity we assume that the input states to be distilled are independent and all have the same error probability p.

As described in the previous section, the ten input resource states can be partitioned into two resource states that are encoded into the code $\mathcal{C}_4$ (data states) and four pairs of resource states used to implement ${}^{C}H$ gates (gate states). The effect of one error on either type of resource state can be understood as follows.

A Y error on one of the data states becomes an encoded Y error, which flips the outcome of the encoded measurement (the measurement of $\bar{H}_1\bar{H}_2$) and is thus detected by the routine. The decoding exactly reverses the encoding, and the logical gates in between preserve logical Y errors, so errors on data states persist on the output and are not detected by the syndrome measurement.

As shown in Fig. 3, a Y error on one of the gate states causes the intended ${}^{C}H$ gate to act as ${}^{C}H$ followed by either $Z \otimes Y$ or $I \otimes Y$, depending on which resource state was in error. Using circuit identities, these errors can be propagated to a common location just before the second set of ${}^{C}H$ gates, as depicted in Fig. 4(b). At this location, such an error appears as a combination of some logical operator and a Y error on a single qubit, which is not an encoded Pauli operator for the $\mathcal{C}_4$ code. This Y error is followed only by logical operators, which cannot take an error subspace to a non-error subspace, and decoding, which returns a syndrome indicating whether the state is in an error subspace. Consequently, a single error on a gate state is detected by the syndrome measurement.

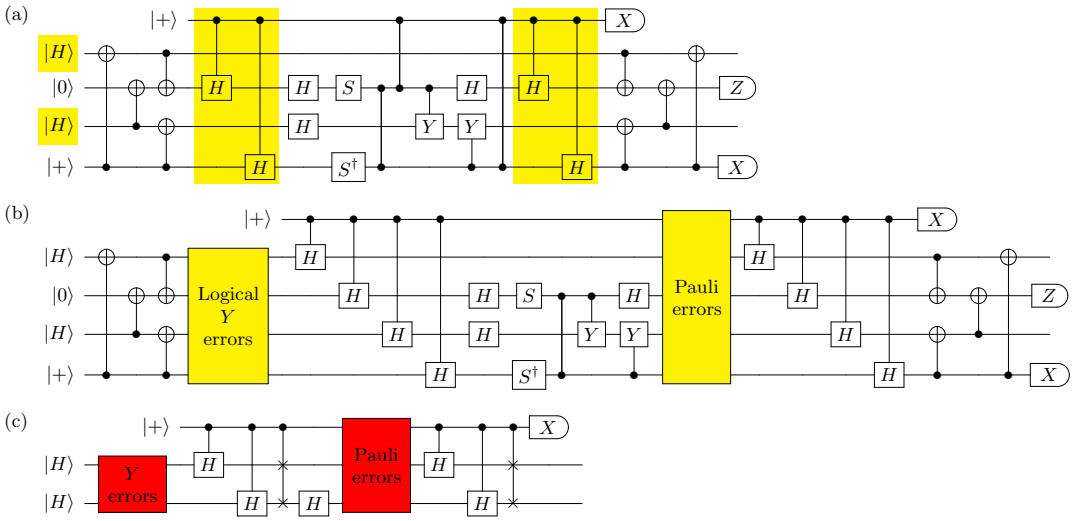

Fig. 4. Illustration of a method of classifying errors in our distillation routine. The first circuit shows the full distillation routine with possible error locations shaded. Using circuit identities, errors at any of these locations can be concentrated into one of two regions (and types) yielding an equivalent circuit of the form shown in (b). Any Pauli errors on the lower four (code) qubits in the second error region of (b) will be detected by the decoding circuit unless they act as a logical (Pauli) operator on the code. The remaining possible errors, those undetectable by the syndrome measurement, may then be enumerated and classified efficiently using the logical circuit shown in (c).

The effect of multiple errors is best understood by propagating the errors from both gate and data states to two locations, as described in Fig. 4. The Y Pauli operators from any pair of errors on gate states (described above) combine to form a logical operator for the code, so any even number of errors on such states will fail to be detected by the decoder, while any odd number of errors will be detected. For each error pattern that is not detected by the syndrome, one can consider the effect of the logical errors on the encoded information and encoded measurement. For example, even numbers of Z errors on the encoded-measurement ancilla

will cancel and cause the distillation to be accepted. Each pattern of errors on the resource states can then be classified first by whether it is detected and then by whether it causes a non-trivial logical error. Because errors on each state are considered to be equiprobable and independent, this enumeration determines the probability $a(p)$ of the distillation routine accepting and the marginal error probability $e(p)$ of an output state conditional on acceptance. It happens that $e(p)$ does not depend on which of the two output states is considered.

Based on the observation that any single error results in rejection, a simple estimate of the acceptance probablity is $a(p) = 1 - 10p + O(p^2)$. The exact accounting yields

$$\begin{aligned} a(p) = 1 - 10p + 58p^2 - 192p^3 + 400p^4 \\ - 544p^5 + 480p^6 - 256p^7 + 64p^8 \ . \end{aligned}$$

The probability of an undetected error on the output states is the probability that the routine accepts and that the output nevertheless has an error. Because any single error is detected, this probability has order p^2. The marginal undetected-error probability of the first (or identically the second) output state is

$$\begin{aligned} u(p) = 9p^2 - 56p^3 + 160p^4 - 256p^5 \\ + 240p^6 - 128p^7 + 32p^8 \ . \end{aligned}$$

For our purposes, this is the quantity of interest, but one can also compute the probability of at least one undetected error on the two outputs. This is given by

$$\begin{aligned} u_2(p) = 13p^2 - 80p^3 + 228p^4 - 368p^5 \\ + 352p^6 - 192p^7 + 48p^8 \ . \end{aligned}$$

The quality of the distillation routine's output is quantified by the marginal probability of error of an output state conditional on acceptance:

$$e(p) = u(p)/a(p) \ . \tag{3}$$

The corresponding probability of at least one error on the two outputs conditioned on acceptance is $e_2(p) = u_2(p)/a(p)$. It can be shown numerically that $e_2(p) \leq 2e(p) - e(p)^2$, so errors on the two output states are positively correlated. In fact, the probability of an error on both output states is of order p^2.

5 Distillation Sequences

The ultimate goal of magic-state distillation is to produce resource states of sufficiently high quality that they can be used to implement all non-Clifford gates in a computation without significantly increasing the probability that the computation will fail. A generic computation will fail if any single gate fails, so the probability of one or more errors on the R resource states employed in a computation must be much less than 1 to ensure that the computation succeeds with high probability. By the union bound, it is sufficient that the marginal probability of error on each resource state be much less than $1/R$. Strong correlations can reduce this requirement on marginal probabilities, but for independent errors the bound is necessary. Consequently, the proximate goal of magic-state distillation is to produce resource states such

that the marginal probability of error for any single state is bounded from above by some goal error probability, $e_g \ll 1/R$. In algorithms currently envisoned for quantum computers, R can easily be 10^{10} or more.

In order to obtain resource states with very low probabilities of error, it is necessary to use multiple rounds of distillation, where the input to each round is produced by the preceding one. We consider a sequence of such rounds where each is based on a single but possibly round-dependent distillation routine. In a round based on an m-to-n distillation routine, the output resource states from the preceding round are grouped into blocks of size m, and each block is then distilled to n states, which may, in general, have correlated errors.

The sequence of rounds is chosen to minimize the number of input resource states needed to produce a given number of output states with marginal probability of error e_g or less. In practice, we are interested in the case where the number of resource states to be prepared is very large, allowing us to consider only the asymptotic cost. The cost is defined as the number of input resource states used per output resource state produced. For one round of the 10-to-2 distillation routine, the cost is $\frac{10}{2a(p)}$, with marginal probability of error $e(p)$ on the output states conditioned on acceptance.

The one-round expression for marginal probability of error given in Sec. 4 assumes that the input resource states suffer from errors independently and with equal probability. Generally, however, the output of a distillation routine need not satisfy either restriction, which poses a concern for distillation sequences involving multiple rounds. If necessary, distillation routines can be output symmetrized by randomly permuting the output states, thereby ensuring that the output states from a given round all have the same error probability. Independence is a concern whenever a routine that outputs more than one state per instance is used, since errors on the states output by one instance of such a routine are usually not independent. For example, the probability of two errors in the output of the 10-to-2 distillation routine is of the same order as that for one error. Performing a distillation routine using such correlated states as input can substantially increase the output error probability. To avoid this effect, it is sufficient to ensure that no instance of a routine depends on more than one output from any previously executed instance of a routine. The following lemma and its corollary show that this strategy works without an increase in asymptotic cost.

Lemma 1 *Let $\mathcal{D}$ be an m-to-n output-symmetrized distillation routine with acceptance probability $a(p)$ and conditional output error probability $e(p)$ for each output state. Given a block of K independent resource states, each with error probability p, one can produce n blocks of output states where each block's states are independent within the block and have probability of error $e(p)$. Each block contains $a(p)\lfloor K/m \rfloor$ states on average.*

Proof. Partition the K resource states into $\lfloor K/m \rfloor$ sets of m states, discarding any remaining ones. Apply $\mathcal{D}$ to each set of m states, getting n states with probability $a(p)$ in each case. Conditional on acceptance, each output state has marginal probability of error $e(p)$, though these errors are not independent. Form n blocks by taking the jth output state from each successful distillation, for $j = 1, \ldots, n$. These blocks have the desired properties for a given pattern of distillation successes. Because the error probabilities of the jth output states of the successful distillations do not depend on the pattern of successes, any such arrangement of the output states that depends only on the pattern of successes preserves this independence □.

Corollary 1 *If, in Lem. 1, K is random with average $\langle K\rangle$, then the expected total number of output states is at least $a(p)n\left(\frac{\langle K\rangle}{m}-1\right)$. The average size of each of the n output blocks of independent states is at least $a(p)\left(\frac{\langle K\rangle}{m}-1\right)$.*

A multi-round distillation sequence can now be formulated as follows: Assume that after round $l-1$ there are N_{l-1} blocks of resource states, where within each block the states are independent with identitical error probabilities p_{l-1}, and the number of states in each block is $\langle K_{l-1}\rangle$ on average. Applying the procedure of Lem. 1 to each block with a $m_l \to n_l$ distillation routine $\mathcal{D}_l$ yields $N_l = N_{l-1}n_l$ output blocks, where each output block has $\langle K_l\rangle \geq a_l(p_{l-1})\left(\frac{\langle K_{l-1}\rangle}{m_l}-1\right)$ resource states on average, independent within a block and each with error probability $p_l = e_l(p_{l-1})$. The first round starts with K_0 independent resource states, each of which suffer an error with probability p_0. For large K_0, the constant offsets of -1 in the expressions are negligible. Consequently, the asymptotic cost c_l of resource-state production after round l satisfies $c_l = \frac{m_l}{n_l a_l(p_{l-1})}c_{l-1}$, where $c_0 = 1$. The error probability after round l satisfies $p_l = e_l(p_{l-1})$.

It is worth emphasizing that the number of outputs produced by an l-round distillation sequence using this approach is $N_l\langle K_l\rangle$, which, neglecting the probability of rejection, is an integer multiple of $N_l = \prod_{i=1}^{l} n_i$. If many of the output states are superfluous, the number of input states required by such a sequence can be much greater than c_l multiplied by the desired number of outputs. For example, using the construction of Lem. 1 but neglecting rejection, a distillation sequence involving l rounds of the 10-to-2 routine takes 10^l input states and outputs 2^l states for a cost of $c_l = 10^l/2^l = 5^l$, but at least 10^l inputs are required regardless of the number of outputs desired. Due to the large number of non-Clifford gates required for a non-trivial computation, this is unlikely to be a concern for the design of quantum computing architectures.

Nonetheless, it may sometimes be possible to restrain the growth in the number of output states as a function of the length of the distillation sequence by allowing states with dependent errors to be used for certain, carefully chosen, inputs of a single instance of a distillation routine. This happens to be the case for sequences involving the 10-to-2 distillation routine: Given pairs of $|H\rangle$ states with correlated errors as inputs where each pair is independent of the others, one may implement three of the four ${}^{C}H$ gates each using a pair and split the two additional pairs between the final ${}^{C}H$ gate and the inputs that are encoded. A correlated error on either of the latter two pairs is detected during syndrome decoding, and a correlated error on any other pair is detected by the encoded measurement, so a quadratic improvement in the error probability is obtained even for such inputs. Consequently, though the 10-to-2 routine outputs pairs of states with correlated errors, the number of output states for a distillation sequence involving only the 10-to-2 routine and single-output routines need never exceed two. A formal analysis of this approach would require tracking multiple error parameters. Since an excess of output states is unlikely to be a problem in practice, we focus on the simpler strategy of separating correlated outputs. This allows all calculations to proceed as if the output states of all routines were independent.

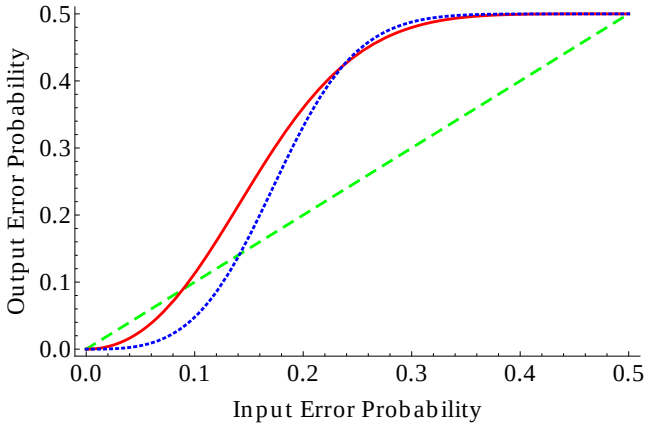

Fig. 5. Plots of the marginal error probabilities conditional on acceptance for the 10-to-2 (solid) and 15-to-1 (dotted) routines. The dashed line indicates the output if no distillation is performed. The thresholds for the two routines are determined by the first intersections with this line.

6 Comparative Performance

At present, practical fault-tolerant architectures require physical-gate error probabilities well below .01. In such a scenario, it should be possible to prepare encoded resource states with error probabilities of a few percent or less prior to distillation. Given such states as input, the most practical routine for $|H\rangle$ distillation developed to date is the 15-to-1 routine. In this section, we compare the performance of the 10-to-2 routine to that of the 15-to-1 routine and consider the effect of using them in concert.

Routines for magic-state distillation are typically judged on the basis of their threshold, that is, the error probability below which resource states can be successfully distilled. At the threshold, a distillation routine outputs resource states no better than the inputs. Thus, the threshold p_t for the 10-to-2 routine can be determined from Eq. (3) by considering solutions to $p_t = e(p_t)$. This yields a threshold of $p_t = 0.089$, which is substantially below the threshold of 0.141 for the 15-to-1 routine [2], but either threshold should be adequate for the error regime of interest. The curves for the marginal output error probability of the 10-to-2 and 15-to-1 routines are plotted in Fig. 5.

The efficiency of a distillation routine can be characterized, as detailed in Ref. [2], by the output error probability as a function of the number of resource states employed. In the limit of small initial error probability p, the output error probability for the 10-to-2 routine after l rounds of distillation is $\frac{1}{9}(9p)^{2^l}$. In the limit of both small p and many output states, l rounds of distillation require $k = 5^l$ input resource states per output. Consequently, taking l to be continuous, the asymptotic output error probability as a function of the number of input resource states expended is $\frac{1}{9}(9p)^{k^\xi}$, where $\xi = \frac{1}{\log_2(5)} \approx .43$. The corresponding exponent for the 15-to-1 routine is .4, so the 10-to-2 routine performs slightly better for this metric. However, these smooth functions hide the step discontinuities induced by using sequences of increasing integral lengths (as seen in Fig. 7) and can be misleading for practical comparisons.

Of greater utility to us is the cost, in resource states consumed per output state, required to obtain resource states of sufficiently high quality for useful quantum computations, given

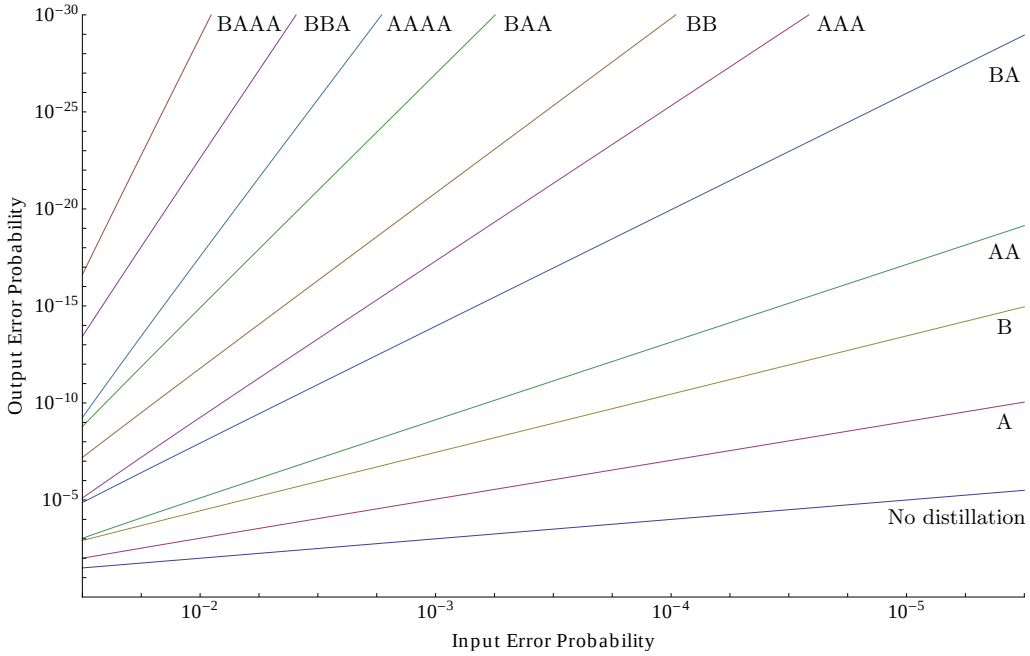

Fig. 6. Output error probability as a function of input error probability for various sequences of the 10-to-2 (A) and 15-to-1 (B) $|H\rangle$ distillation routines. Each curve is labeled by the associated sequence of routines, e.g., BAA denotes the 15-to-1 routine followed by two rounds of the 10-to-2 routine. The region directly underneath each labeled curve is the region in which the labeled strategy is preferred.

resource states with error probabilities in the range of 0.01 to 10^{-5}. The cost depends on the distillation sequence, which generally entails multiple rounds of distillation. For the purpose of optimizing the distillation sequence, one can consider arbitrary routines at each round. Here, we consider sequences involving the 10-to-2 and 15-to-1 distillation routines.

In Fig. 6 we plot the output error probability as a function of input error probability for various sequences. Data for the 15-to-1 routine was computed using the expressions corresponding to $a(p)$ and $e(p)$ in Eq. (35) and Eq. (36) of Ref. [2]. In the region plotted, distillation sequences with higher output error (lower curves) also require fewer input resource states per output state. Consequently, for a given output error goal, e_g, and input error probability p, the label of the nearest curve above the point (p, e_g) in the plot gives the best distillation sequence involving the 10-to-2 and/or 15-to-1 routines.

Table 1 shows the costs and improvements for a number of distillation sequences given an initial error probability of $p = 0.01$. Not surprisingly, the table shows that the 10-to-2 routine has a smaller cost, but the 15-to-1 routine has greater improvement in error probability per round. For distillations that use both routines, we find numerically that if 15-to-1 rounds are used they should be placed first. This is intuitively consistent with the higher threshold for the 15-to-1 routine, which suggests better performance at high error probabilities.

The cost improvements shown in Tab. 1 are illustrated more visually in Fig. 7, which shows the production cost, at a fixed input error probability $p = 0.01$ and as a function of e_g, of the best distillation sequence compared to the best sequence using only the 15-to-1 routine.

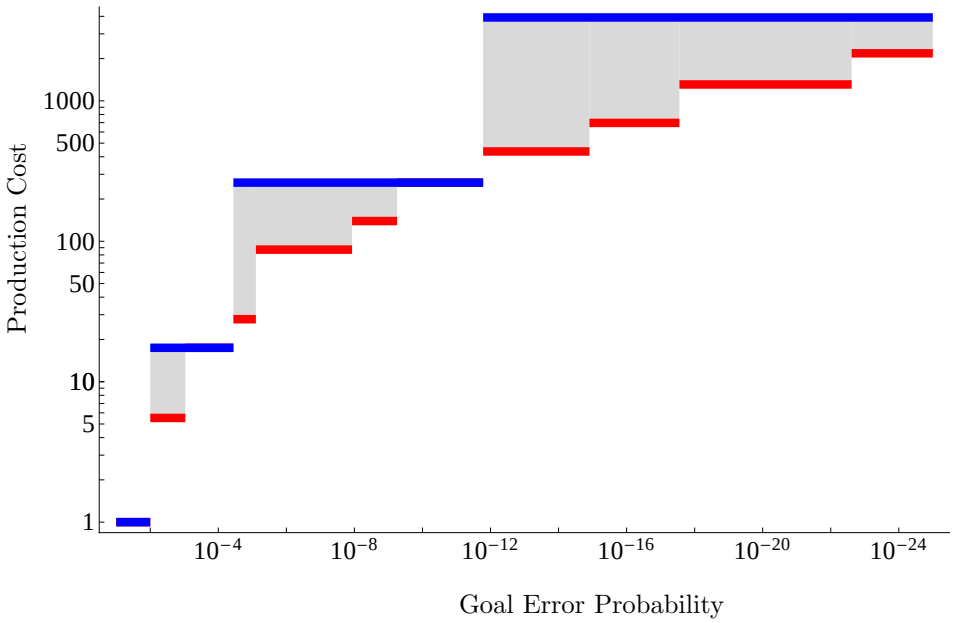

Fig. 7. Log-log plot of the cost required to produce output states satisfying a goal error probability (e_g) given input states with error probability 0.01. The upper horizontal segments show the cost for the best sequence involving only the 15-to-1 routine. The lower horizontal segments show the cost for the best sequence combining the 15-to-1 and 10-to-2 routines. The gray regions indicate the improvement obtained using the 10-to-2 routine.

For example, a goal error probability near 10^{-5} can be achieved by using either two rounds of the 15-to-1 routine at a production cost of 261.7 or two rounds of the 10-to-2 routine at a cost of 27.9. In this case, the improvement in production cost obtained by incorporating the 10-to-2 routine is a factor of 9.4.

7 Conclusions

Magic-state distillation enables universal quantum computing given only mediocre copies of a non-stabilizer state and high-quality Clifford operations. Considering the importance of Clifford-based techniques to the theory of fault tolerance, we expect that magic-state distillation will prove valuable for the practical implementation of quantum computers.

At the logical level, computationally useful quantum algorithms involve many non-Clifford gates, generally enough to account for a significant fraction of all gates employed. At least one high-quality magic state is required for the indirect implementation of each non-Clifford gate, so it is important to minimize the resources needed for the distillation of such states.

In this work, we contributed to the goal of resource reduction by introducing an $|H\rangle$ distillation routine that reduces the error probability for faulty $|H\rangle$ states from p to $O(p^2)$ and produces 2 output states using 10 input states. By judiciously combining this routine with the higher-order (p to $O(p^3)$) but higher-cost 15-to-1 routine from Refs. [2, 3], we showed that the number of faulty $|H\rangle$ states required to distill states of a given quality can be reduced by up to an order of magnitude.

As in Ref. [3], the general approach employed here was to verify encoded $|H\rangle$ states by using

Distillation scheme	Cost	Output error probability, $e(p)$	Cost improvement factor
A	5.5	9×10^{-4}	3.2
B	17.4	4×10^{-5}	1
AA	27.9	7×10^{-6}	9.4
BA	87.2	1×10^{-8}	3.0
AAA	139.3	5×10^{-10}	1.9
BB	261.7	2×10^{-12}	1
BAA	436.2	1×10^{-15}	9.0
$AAAA$	696.6	2×10^{-18}	5.6
BBA	1308.7	2×10^{-23}	3.0
$BAAA$	2180.8	1×10^{-29}	1.8

Table 1. Costs and output error probabilities at $p = 0.01$. The labels for the distillation schemes follow the convention given in Fig. 6. The cost improvement factor is with respect to the shortest sequence using only the 15-to-1 routine that achieves at least as good an output error probability.

transversally applied controlled-Hadamard gates to measure transversal logical Hadamard. We expect that this approach can be applied to larger codes, but it is not obvious that the gate cancellations in Fig. A.1 that make our scheme particularly efficient have an analogue for these codes. More generally, multiple-output distillation routines with larger numbers of inputs are promising candidates for reducing the resource requirements for magic-state distillation. Basing such routines on higher-dimensional systems offers the potential for further resource reductions, as indicated by the recent work of Campbell et al. on single-output qudit distillation routines [10]. The effect on the distillation cost of moving to larger codes and the ultimate bounds on the cost of magic-state distillation are interesting open problems.

Acknowledgments

We thank Scott Glancy for his help in bringing this work to fruition. We also thank Ben Reichardt and a referee for encouraging us to think more about the necessity of splitting up the outputs of distillation routines. This paper is a contribution of the National Institute of Standards and Technology and not subject to U.S. copyright.

References

1. G. Nebe, E. M. Rains and N. J. A. Sloane (2001), *The invariants of the Clifford groups*, Des. Codes and Cryptogr., 24, pp. 99–122, arXiv:math/0001038.
2. S. Bravyi and A. Kitaev (2005), *Universal quantum computation with ideal Clifford gates and noisy ancillas*, Phys. Rev. A, 71, pp. 022316, arXiv:quant-ph/0403025.
3. E. Knill (2004), *Fault-tolerant postselected quantum computation: Schemes*, arXiv:quant-ph/0402171.
4. B. Reichardt (2005), *Quantum universality from magic states distillation applied to CSS codes*, Quantum Inf. Process., 4, pp. 251-264, arXiv:quant-ph/0411036.
5. E. T. Campbell and D. E. Browne (2009), *On the structure of protocols for magic state distillation*, Lect. Notes Comput. Sc.: Theory of Quantum Computation, arXiv:0908.0838.
6. E. T. Campbell and D. E. Browne (2010), *Bound states for magic state distillation in fault-tolerant quantum computation*, Phys. Rev. Lett., 104, arXiv:0908.0836.

7. N. C. Jones, R. Van Meter, A. G. Fowler, P. L. McMahon, J. Kim, T. D. Ladd and Y. Yamamoto (2012), *Layered architecture for quantum computing*, Phys. Rev. X, 2, 031007, arXiv:1010.5022.
8. N. Isailovic, M. Whitney, Y. Patel and J. Kubiatowicz (2008), *Running a quantum circuit at the speed of data*, Intl. Symp. on Computer Architecture, IEEE Computer Society, pp. 177-188, arXiv:0804.4725.
9. M. A. Nielsen and I. L. Chuang (2001), *Quantum Computation and Quantum Information*, Cambridge University Press.
10. E. T. Campbell, H. Anwar and D. E. Browne (2012), *Magic state distillation in all prime dimensions using quantum Reed-Muller codes*, arXiv:1205.3104.

Appendix A

Fig. A.1 provides additional details about the circuit identities used in Fig. 2.

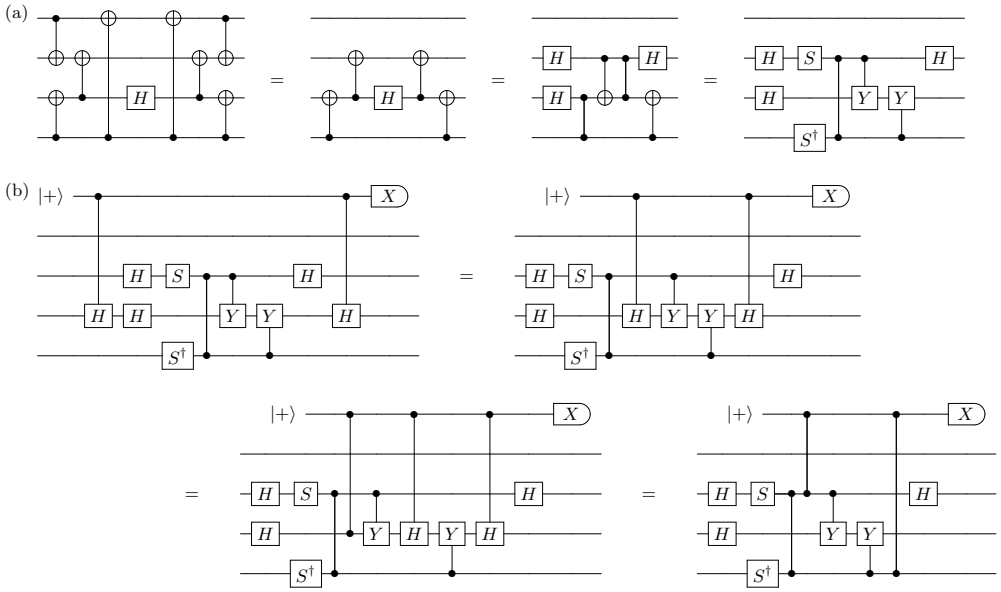

Fig. A.1. Additional details regarding the relations between circuits in Fig. 2. (a) A sequence of circuit identities deriving the form of the encoded Hadamard gate used in Fig. 2(b). The starting circuit implements H on the second logical qubit of the code $\mathcal{C}_4$ by decoding the logical qubits into the first and third physical qubits, applying H to the third qubit, and re-encoding. The first equivalence is obtained by commuting and cancelling pairs of ${}^{C}X$ gates. The second equivalence uses the decomposition of the ${}^{C}X$ gate into ${}^{C}Z$ and H gates several times as well as the identity $H^2 = I$. The final equivalence uses two facts: $ZX = iY$ and the order of a ${}^{C}X$ and a ${}^{C}Z$ gate with the same target can be exchanged if a ${}^{C}Z$ gate is added between the controls. (b) A sequence of circuit identities showing why the pair of ${}^{C}H$ gates targeting the fourth qubit in Fig. 2(b) can be eliminated. Other than reorganization of commuting gates, these equivalences rely on the fact that, because H anticommutes with Y, ${}^{C}H$ and ${}^{C}Y$ gates with the same target can be exchanged if a ${}^{C}Z$ gate between the two controls is added.