Tensor-network simulations of the surface code under realistic noise

Andrew S. Darmawan and David Poulin
Département de Physique & Institut Quantique, Université de Sherbrooke, Québec, Canada
(Dated: July 19, 2016)

The surface code is a many-body quantum system, and simulating it in generic conditions is computationally hard. While the surface code is believed to have a high threshold, the numerical simulations used to establish this threshold are based on simplified noise models. We present a tensor-network algorithm for simulating error correction with the surface code under arbitrary local noise. Our simulation is exact within statistical fluctuations and we use it to study the threshold and the sub-threshold behaviour of the amplitude-damping and systematic rotation channels. We also compare these exact results to those obtained by making standard approximations to the noise models.

The working principle behind quantum error correction is to 'fight entanglement with entanglement', i.e., protect the data against local interaction with the environment by encoding it into delocalized degrees of freedom of a many-body system. Thus, characterizing a fault-tolerant scheme is ultimately a problem of quantum many-body physics.

While simulating quantum many-body systems is generically hard, particular systems have additional structure that can be taken advantage of. For example, free-fermion Hamiltonians have algebraic properties that makes them exactly solvable. An analogy in stabilizer quantum error correction is Pauli noise, where errors are Pauli operators drawn from some fixed distribution. Due to their algebraic structure, Pauli noise models can be simulated efficiently using the stabilizer formalism [1]. Beyond Pauli noise, noise composed of Clifford gates and projections onto Pauli eigenstates can also be simulated efficiently using the same methods [2].

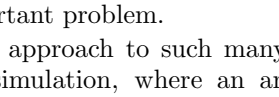
While such efficiently simulable noise models can be useful to benchmark fault-tolerant schemes, they do not represent most models of practical interest. For instance, qubits that are built out of non-degenerate energy eigenstates are often subject to relaxation, a.k.a. amplitude damping. Miscalibrations often result in systematic errors corresponding to small unitary rotations [3]. Given that these processes do not have efficient descriptions within the stabilizer formalism, understanding how a given fault-tolerant scheme will respond to them is a difficult and important problem.

The simplest approach to such many-body problems is brute-force simulation, where an arbitrary state in Hilbert space is represented as an exponentially large vector of coefficients. Using such methods, small surface codes (up to distance 3) have been simulated under non-Clifford noise [4]. In another study, brute-force simulation of the seven-qubit Steane code was performed without concatenation [5]. Simulation of such low distance codes allow comparison of noise at the logical level to the noise on the physical level, however it is difficult to infer quantities of interest such as thresholds or overheads from such small simulations. Another approach, akin to the use of tight-binding approximations in solid-state physics, is to approximate these noise processes with efficiently simulable ones [2, 6]. However, the accuracy of these approximations can be very poor as we will show below.

In this work, we import quantum many-body methods developed in the context of solid-state physics to study quantum error correction with realistic, non-Clifford noise models. Our construction hinges on the fact that the surface code is a projected-entangled-pair state (PEPS) with low bond dimension [7]. As a basic demonstration, we use our method to simulate the surface code under two non-Clifford local noise models: amplitude damping and systematic rotation. We assume that syndrome measurements are performed perfectly. We obtain thresholds for these noise models and also study error correction in the region of practical interest, where the noise strength is low relative to the threshold. These results are compared with those obtained using standard Pauli approximations, and significant discrepancies are observed. We have performed simulations on codes of up 153 qubits, while in contrast, previous studies using brute-force simulations were limited to about 25 qubits [4].

Tensor networks — For our purposes, a tensor $A_{i_1,i_2,\ldots,i_n}$ is an n-index array of complex numbers, where i_k runs from 0 to D_k-1, where D_k is called the bond dimension of i_k. We represent a tensor graphically as a box, and each index of the tensor as an edge emanating from that box. When two tensors are linked by an edge, the connected indices are summed over, for instance

$$\sum_{\alpha} A_{\alpha,i_1,i_2,\ldots} B_{\alpha,j_1,j_2,\ldots} =$$



.

It will often be convenient to group multiple indices together, using the notation $A_{i_1,i_2,\ldots,i_n} = A_{\boldsymbol{i}}$ where a bold subscript is taken to mean that the index is composed of multiple indices.

We will define an N-particle projected entangled pair state (PEPS) $|\psi\rangle$ to be a quantum state whose tensor of coefficients $\psi_{i_1,i_2\ldots i_N} = \langle i_1,\ldots,i_N|\psi\rangle$ is the contraction of a network of N tensors $\mathcal{T} = \{A^{(1)}_{i_1,\boldsymbol{\alpha}_1}, A^{(2)}_{i_2,\boldsymbol{\alpha}_2},\ldots,A^{(N)}_{i_N,\boldsymbol{\alpha}_N}\}$, each of which has one physical index labelled i and some number of virtual indices labelled $\boldsymbol{\alpha}$. We assume that the physical particles have

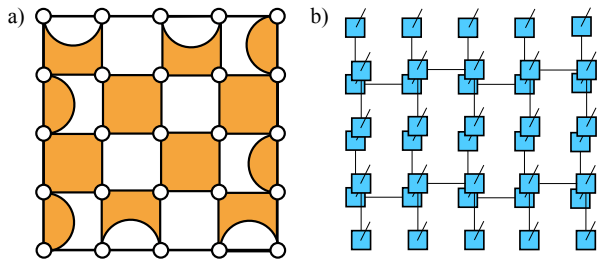


FIG. 1. a) Distance-5 surface code layout with qubits on the vertices of a rectangular lattice. Orange faces indicate x-checks, while white faces indicate z-checks. b) A tensor network representing an encoded $|0\rangle_L$ state on this layout. Each blue box is a Q^+ tensor, virtual indices all lie in the plane of the page, and physical indices are perpendicular to it. The bond layout is optimised for the exact contraction scheme we have used.

dimension d, that each virtual index has bond dimension D, and the number of virtual indices per tensor in $\mathcal{T}$ is less than a constant n. This implies that the tensor network description of $|\psi\rangle$ is memory efficient: while the tensor $\psi_{i_1,i_2\ldots i_N}$ has d^N entries, each tensor in $\mathcal{T}$ has at most dD^n entries and thus the whole set $\mathcal{T}$ can be specified with only NdD^n complex numbers.

The surface code tensor network — We consider the optimized layout of the surface code introduced in [8]. Qubits are placed on the vertices of a $W \times L$ rectangular lattice, as shown in Fig. 1a). For every orange face f, define an x-check operator $A_f = \prod_{i\in f} X_i$, where the product is taken over all qubits in f. Likewise, for every white face f define a z-check operator $B_f = \prod_{i\in f} Z_i$.

The logical qubit state $|0\rangle_L$ is defined to be the simultaneous $+1$ eigenspace of all check operators and of the logical operator $\overline{Z}$, where $\overline{Z}$ is a string of Z's along the left boundary of the code. Likewise, the state $|1\rangle_L$ is fixed by all check operators but is a -1 eigenstate of $\overline{Z}$, it can be obtained as $\overline{X}|0\rangle_L$, where $\overline{X}$ is a product of X operators along the bottom boundary of the code. Given that the product state $|0\rangle^{\otimes N}$ has non-zero overlap with $|0\rangle_L$, and that $|0\rangle^{\otimes N}$ is a $+1$ eigenstate of every B_f and $\overline{Z}$, we have

$$|0\rangle_L \propto \prod_{f\in F_B} \frac{1}{2}(I + A_f)|0\rangle^{\otimes N}\,, \tag{1}$$

where $\frac{1}{2}(I + A_f)$ is the projector onto the $+1$ eigenspace of A_f.

Let $W_{i,i',\boldsymbol{\alpha}}(C)$, be a tensor with two physical indices i, i' and an arbitrary number of virtual indices $\boldsymbol{\alpha} = \alpha_1, \alpha_2, \ldots$, that depends on a 2×2 matrix C. Each index of $W(C)$ has bond dimension two, and the only non-zero entries of $W(C)$ are

$$W_{i,i',\mathbf{0}}(C) = \delta_{i,i'} \quad \text{and} \quad W_{i,i',\mathbf{1}}(C) = C_{i,i'}, \tag{2}$$

where $\mathbf{0}(\mathbf{1})$ means that all virtual indices are set to 0 (1), and the symbol $\delta_{i,i'}$ denotes the Kronecker delta. For convenience, we define $Q^{\pm} = W(\pm X)$ and $R^{\pm} = W(\pm Z)$.

Consider the projector $\frac{1}{2}(I + A_f)$ acting on particles 1,2,3 and 4. It can easily be verified that the tensor $\langle i_1, i_2, i_3, i_4|I + A_f|i'_1, i'_2, i'_3, i'_4\rangle$ can be expressed as a contraction of four tensors

$$\sum_{\alpha_1,\alpha_2,\alpha_3} Q^+_{i_1,i'_1,\alpha_1} Q^+_{i_2,i'_2\alpha_1,\alpha_2} Q^+_{i_3,i'_3\alpha_2,\alpha_3} Q^+_{i_4,i'_4\alpha_3}\,. \tag{3}$$

We remark that tensor description for the projection onto the -1 eigenspace of A_f is identical to Eq. (3) but with any one of the Q^+ tensors replaced with Q^-. Projections onto eigenspaces of B_f operators can be defined analogously by replacing all Q's with R's. The tensor network corresponding to the product of projectors $\prod_{f\in F_B} \frac{1}{2}(I + A_f)$ in Eq. (1), is obtained by overlapping the tensor projector in Eq. (3) over the whole lattice.

The state $|0\rangle_L$ is then obtained by applying this projector to the state $|0\rangle^{\otimes N}$, which, in the tensor network picture, corresponds to fixing the second index to zero, thus effectively removing it. A tensor network $\mathcal{T} = \{A^{(1)}_{i_1,\boldsymbol{\alpha}_1}, A^{(2)}_{i_2,\boldsymbol{\alpha}_2}, \ldots, A^{(N)}_{i_N,\boldsymbol{\alpha}_N}\}$ for the state $|0\rangle_L$ is thus formed from contractions of Q^+ tensors, as shown in Fig. 1b).

While this tensor network describes the logical $|0\rangle_L$ state, we will need to be able to represent other states in order to fully characterise the transformation of the encoded information during error correction. Specifically, we want to compute the logical channel $\mathcal{E}_L$ that is applied to the logical qubit during a round of error correction. By the Choi-Jamiolkowski isomorphism, this channel can be inferred directly from the resulting output when a Bell state of the form $|\Psi^+\rangle = |0\rangle_L|0\rangle_a + |1\rangle_L|1\rangle_a$ is input, where the first qubit is encoded in a surface code and is subject to error correction, while the second qubit is unencoded and assumed to be noise free. We can obtain a tensor network description for $|\Psi^+\rangle$ by a simple modification to the tensor network describing $|0\rangle_L$. Consider the tensor

$$\overline{\text{CNOT}} := \sum_{\alpha_1,\alpha_2,\ldots,\alpha_L} Q^+_{i_1,i'_1,\alpha_1} Q^+_{i_2,i'_2\alpha_1,\alpha_2} \ldots Q^+_{i_L,i'_L\alpha_L,\alpha_a}\,, \tag{4}$$

which can be thought of as an L qubit operator (expressed in tensor form) with a single uncontracted virtual index α_a, which we call the ancilla index. If α_a is set to 0 then $\overline{\text{CNOT}}$ is simply the identity on its physical indices, while if $\alpha_a = 1$, $\overline{\text{CNOT}}$ is an L-fold tensor product of X. Therefore if $\overline{\text{CNOT}}$ is applied to the bottom row of the tensor network for $|0\rangle_L$, the resulting tensor network state $\psi_{i_1,i_2,\ldots,i_N,\alpha_a} = \langle i_1, i_2, \ldots, i_N, \alpha_a|\psi\rangle$ (which includes the ancilla index as a physical particle) will be the desired Bell state $|0\rangle_L|0\rangle_a + (\overline{X}|0\rangle_L)|1\rangle_a = |\Psi^+\rangle$.

The above tensor network definitions are for pure states. However as we will be considering non-unitary noise, we will want a tensor network description for density matrices. Given a tensor network

$\{A^{(1)}_{i_1,\boldsymbol{\alpha}_1}, A^{(2)}_{i_2,\boldsymbol{\alpha}_2}, \ldots, A^{(N)}_{i_N,\boldsymbol{\alpha}_N}\}$ for a pure state $|\psi\rangle$, we obtain the tensor network $\{B^{(1)}_{\boldsymbol{i_1},\boldsymbol{\alpha}_1}, B^{(2)}_{\boldsymbol{i_2},\boldsymbol{\alpha}_2}, \ldots, B^{(N)}_{\boldsymbol{i_N},\boldsymbol{\alpha}_N}\}$ for the density matrix $\langle i_1,\ldots,i_N|\psi\rangle\langle\psi|i'_1,\ldots,i'_N\rangle$ by defining $B^{(k)}_{\boldsymbol{i},\boldsymbol{\alpha}} = A^{(k)}_{i,\boldsymbol{\alpha}'} A^{(k)*}_{i',\boldsymbol{\alpha}''}$, where $\boldsymbol{i} = i, i'$ and $\boldsymbol{\alpha} = \boldsymbol{\alpha}', \boldsymbol{\alpha}''$ represent combined sets of physical and virtual indices respectively.

Simulation of error correction — We start with the density matrix for the half encoded Bell state described above. The noisy state is obtained by applying the desired CPTP map $\mathcal{E}$ to every qubit in the code. This corresponds to the local tensor update

$$B_{i,i',\boldsymbol{\alpha}} \leftarrow \sum_{j,j'} \mathcal{E}_{ijj'i'} B_{j,j',\boldsymbol{\alpha}}\,, \tag{5}$$

where $\mathcal{E}_{ijj'i'} := \langle i|\mathcal{E}(|j\rangle\langle j'|)|i'\rangle$.

During error correction, every check will be measured, yielding a set of measurement outcomes $s = (m_1, m_2, \ldots)$ corresponding to a syndrome s with probability $p_s = Tr(\Pi_s \rho)$, where Π_s is the projection onto the syndrome subspace corresponding to s and ρ is the noisy code state. We will assume that measurements are performed perfectly. In order to sample from the distribution p_s we imagine measuring checks sequentially. We can compute the conditional probability $p(m_k|m_1, m_2, \ldots, m_{k-1})$ for any k as follows. Let ρ_k be the state of the system after k check measurements and let P_{k+1} be the projection onto the $+1$ eigenspace of the $k+1$th check. The probabilities of obtaining an outcome of 1 and -1 are given respectively by $q = \frac{1}{2}Tr(P_{k+1}\rho_k)$ and $1-q$. To obtain a tensor network for $P_{k+1}\rho_k$, we apply the tensor corresponding to the check projector to the tensor network of ρ_k. For example, for a 4-qubit x-check, the projector is given by Eq. (3) and application to ρ_k corresponds to updating the tensors for the measured particles according to

$$B_{i,i',\boldsymbol{\alpha}} \leftarrow \sum_{j} Q^{+}_{i,j,\boldsymbol{\alpha}'} B_{j,i',\boldsymbol{\alpha}''}\,, \tag{6}$$

where the updated virtual indices $\boldsymbol{\alpha}$ contain the original virtual indices of B as well as those of the appended Q^+. Then, to evaluate $q = \frac{1}{2}Tr(P_{k+1}\rho_k)$, we take the trace of every tensor ($B_{\boldsymbol{\alpha}} \leftarrow \sum_i B_{i,i,\boldsymbol{\alpha}}$) then contract all virtual indices in the network.

This last step corresponds to contracting a square-lattice tensor network. Generically, this problem is $\#P$-complete and therefore no efficient algorithm is believed to exist [9]. While many efficient algorithms have been developed to obtain approximate solutions to this problem [10–15], in this work we have opted for an exact, albeit inefficient, contraction algorithm. Exact simulation will allow us to simulate the surface code well below threshold, where the logical error rates are extremely low (on the order of 10^{-15}). Also, while the algorithm is exponential in the code distance, it significantly outperforms the brute-force algorithm, which is exponential in the number of particles.

To contract the tensor network, we merge all tensors of a given column into a singe tensor, and then contract from left to right. The inefficiency of the algorithm stems from the fact that amount of memory required to store the tensor associated to a column is exponential in the number of inter-column bonds, which is proportional to the lattice width. However, this number can be substantially reduced by exploiting the invariance of the check-projector tensors in Eq. (3) under permutation of the particles, and always orienting the three bonds of the check such that there is only one inter-column bond per check. This optimised layout is illustrated for the $|0\rangle_L$ state in Fig. 1b). We also remark that, in sampling the syndrome sequentially, contractions can be reused between successive syndrome measurements, so it is rarely necessary to contract the entire network. The resulting complexity of our algorithm is $\mathcal{O}(LW^2 4^W)$.

When all check outcomes have are obtained, we have exactly sampled a syndrome s from the distribution p_s, and the tensor network encodes a noisy state with appropriate check projectors applied. The state is then returned to the code space via a Pauli operator, which we choose to be a product of X operators connecting each flipped B_f check to the left boundary and Z connecting each flipped A_f check to the top boundary. We call the resulting output state ρ_{out}.

At this point, a classical algorithm, called the decoder, is used to select one of four possible corrections $\overline{I}, \overline{X}, \overline{Y}, \overline{Z}$ to be applied to the state. The decoder uses information about the syndrome and the noise model to select the correction that minimises the overall logical error. For efficiency this is usually only done approximately [16–25]. In this work we perform exact decoding by choosing the correction that minimizes the distance between the computed logical channel (which our simulation algorithm allows us to compute) and the identity. The entire procedure (application of noise, check measurements, returning to the code space, correction) yields a CPTP map $\mathcal{E}_L$ acting on the encoded qubit which can be obtained from the tensor network encoding the state through standard Choi-Jamilkowski isomorphism.

Numerical results — Two non-Pauli noise models are considered: systematic rotation about the z-axis $\mathcal{E}_{SR}(\rho) = e^{-i\theta Z}\rho e^{i\theta Z}$ where $\theta \in [0, \pi)$ is the rotation angle, and amplitude damping $\mathcal{E}_{AD}(\rho) = \sum_i K_i \rho K_i^\dagger$, which has two Kraus operators

$$K_0 = |0\rangle\langle 0| + \sqrt{1-\gamma}|1\rangle\langle 1|\,, \quad K_1 = \sqrt{\gamma}|0\rangle\langle 1|\,, \tag{7}$$

where $\gamma \in [0, 1]$ is the damping parameter. As a test, we also performed simulations with the depolarising channel $\mathcal{E}_{DP}(\rho) = (1-\epsilon)\rho + \frac{\epsilon}{3}X\rho X + \frac{\epsilon}{3}Y\rho Y + \frac{\epsilon}{3}Z\rho Z$, which is a well studied Pauli channel.

We have also considered two different Pauli approximations of these channels. The Pauli twirl approximation (PTA) to an arbitrary channel expressed in the Pauli basis $\mathcal{E}(\rho) = \sum_{i,j} \chi_{i,j} P_i \rho P_j$ is the channel $\mathcal{E}^{\mathrm{PTA}} = \sum_i \chi_{i,i} P_i \rho P_i$, obtained by removing the off-diagonal ele-

	Exact	PTA	HPA
DP (ϵ)	$18.5 \pm 1.5\%$	$18.5 \pm 1.5\%$	$18.5 \pm 1.5\%$
AD (γ)	$39 \pm 2\%$	$39 \pm 2\%$	$21 \pm 1\%$
SR (θ)	> 0.40	0.34	0.11

TABLE I. Computed thresholds for three noise models and their Pauli approximations. PTA and HPA to SR were computed using the exact value of the threshold under bit flip noise. Pauli approximations of DP are identical to the exact channel, so only the exact channel was simulated.

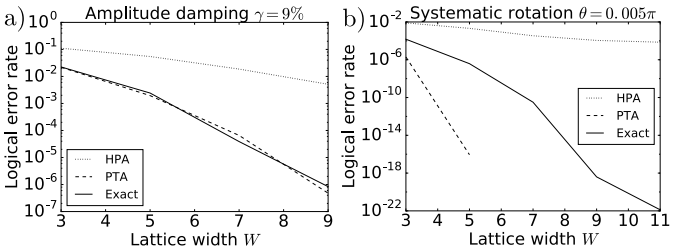


FIG. 2. Empirical logical error rate vs lattice dimension for two non-Pauli noise models. Solid lines represent exact channels while dashed and dotted lines represent the PTA and the HPA, respectively. The left plot is $\gamma = 9\%$ amplitude damping, while the right plot is $\theta = 0.005\pi$ systematic rotation. For each set of parameters, 1.32×10^6 syndromes were sampled to acquire statistics. For the PTA of SR with $W > 5$, error rates are numerically zero. Note that the exact average logical error could significantly differ from the empirical average observed over millions of syndrome measurements if failures are dominated by outliers.

ments of $\chi_{i,j}$ [26, 27]. It can produce noise models that are much better behaved, and thus provide poor insight into the performance of the real channel. For this reason, an honest Pauli approximation (HPA) was introduced [6], which seeks the channel $\mathcal{E}^{\mathrm{HPA}}$ which is as close as possible to the original channel yet produces a noisier output on every possible input. This approximation is thus expected to provide a pessimistic bound on the performance of a FT scheme under some noise process $\mathcal{E}$.

As a first application, we have used the simulation algorithm to estimate thresholds. For a given noise model, we define the threshold to be the largest value of the noise parameter below which increasing the lattice size results in a decrease in the logical error rate, as measured by the diamond distance of the logical channel from the identity. Computed thresholds are presented in table I. The largest lattice sizes simulated were 9×9 for depolarising, 11×11 for systematic rotation and 9×17 for amplitude damping. For amplitude damping we found that a non-square lattice, where the $\overline{X}$ logical operator runs along the long dimension of the code, performed significantly better than a square one. We remark that such lattices are well suited to our contraction algorithm: if the width of the lattice is fixed, then the time taken to perform the simulation scales only linearly with the length of the lattice.

Given that the depolarising channel is a Pauli channel, we can compare the threshold obtained from our data with that obtained using efficient simulations on larger lattices. We find that our threshold estimate agrees with the one obtained in [28] of $17\% \leq \epsilon \leq 18.5\%$ using maximum likelihood decoding on lattices up to 29×29.

The systematic rotation channel was simulated for rotation angles between $0.025\pi \leq \theta \leq 0.3\pi$. A large region showed below threshold behaviour, however no clear transition behaviour could be identified.

The thresholds of Pauli approximations generally did not agree well with the thresholds of the exact channels. One exception was the twirl approximation to amplitude damping, which yielded the same threshold as the exact channel (to within the accuracy of our data). As expected, the honest Pauli approximations provided pessimistic values of the threshold for non-Pauli channels.

As another application of our algorithm, we have performed simulations at fixed noise rates well below threshold. The results are presented in Fig. 2 for amplitude damping with $\gamma = 9\%$, and z-rotation with $\theta = 0.005\pi$. Again, the behaviour of the twirl approximation agrees quite well with that of the exact amplitude damping channel. However the behaviour of both approximations differed considerably from the exact z-rotation. For instance, for a code with $W = 5$, the PTA underestimates the logical error rate by a factor of about 10^{10}, while the HPA over-estimates the logical error rate by a factor of about 10^4. The observed discrepancies highlight the fact that the performance of the code under an efficiently simulable approximation to a noise model can differ significantly from its performance under the exact noise model and motivates the development of more efficient algorithms, such as the one presented here.

Conclusion — We have presented a simple tensor-network based algorithm for simulating the surface code under arbitrary local noise. The algorithm is exact within statistical fluctuations, allowing simulation of error correction well below threshold, where the logical error rate is extremely low. While the algorithm is exponential, it is efficient enough to allow simulation of systems with well over one hundred qubits, and we have used it to estimate thresholds of non-Pauli noise models.

The exponential complexity of the algorithm stems from a square-lattice tensor-network contraction. To simulate larger systems, efficient but approximate contraction algorithms could be used in place of exact contraction, however the accuracy of such algorithms would need to be established. Another approach to improve efficiency could be to take advantage of additional structure in the surface-code tensor-network. For instance, in the noiseless case, the network can be contracted exactly, both as a PEPS [29], and as a MERA [30].

While our simulation algorithm has assumed perfect check measurements, in practice, measurement errors are unavoidable. In surface-code error correction, measure-

ment errors are detected by performing multiple rounds of check measurements and observing how the syndrome evolves in time. This procedure could potentially be simulated using PEPS time-evolution algorithms [31–34], or by contracting 3D tensor networks [35].

Acknowledgements — We thank Marcus da Silva for discussions which initiated this project. This work was supported by the Army Research Office contract number W911NF-14-C-0048.

[1] D. Gottesman, *Stabilizer Codes and Quantum Error Correction*, Ph.D. thesis (1997).
[2] M. Gutiérrez, L. Svec, A. Vargo, and K. R. Brown, Phys. Rev. A **87**, 030302 (2013).
[3] J. Wallman, C. Granade, R. Harper, and S. T. Flammia, New J. Phys. **17**, 113020 (2015).
[4] Y. Tomita and K. M. Svore, Phys. Rev. A **90**, 062320 (2014).
[5] M. Gutiérrez, C. Smith, L. Lulushi, S. Janardan, and K. R. Brown, arXiv:1605.03604 (2016).
[6] E. Magesan, D. Puzzuoli, C. E. Granade, and D. G. Cory, Phys. Rev. A **87**, 012324 (2013).
[7] N. Schuch, I. Cirac, and D. Prez-Garca, Ann. Phys. **325**, 2153 (2010).
[8] H. Bombin and M. A. Martin-Delgado, Phys. Rev. A **76**, 012305 (2007).
[9] N. Schuch, M. M. Wolf, F. Verstraete, and J. I. Cirac, Phys. Rev. Lett. **98**, 140506 (2007).
[10] F. Verstraete and J. I. Cirac, arXiv:cond-mat/0407066 (2004).
[11] M. Levin and C. P. Nave, Phys. Rev. Lett. **99**, 120601 (2007).
[12] Z.-C. Gu and X.-G. Wen, Phys. Rev. B **80**, 155131 (2009).
[13] M. Lubasch, J. I. Cirac, and M.-C. Bañuls, New J. Phys. **16**, 033014 (2014).
[14] G. Evenbly and G. Vidal, Phys. Rev. Lett. **115**, 180405 (2015).
[15] S. Yang, Z.-C. Gu, and X.-G. Wen, arXiv:1512.04938 (2015).
[16] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, J. Math. Phys. **43**, 4452 (2002).
[17] J. W. Harrington, *Analysis of quantum error-correcting codes: symplectic lattice codes and toric codes*, Ph.D. thesis (2004).
[18] G. Duclos-Cianci and D. Poulin, Phys. Rev. Lett. **104**, 050504 (2010).
[19] J. R. Wootton and D. Loss, Phys. Rev. Lett. **109**, 160503 (2012).
[20] S. Bravyi and J. Haah, Phys. Rev. Lett. **111**, 200501 (2013).
[21] G. Duclos-Cianci and D. Poulin, Quant. Inf. Comp. **15**, 0721 (2014).
[22] A. Hutter, J. R. Wootton, and D. Loss, Phys. Rev. A **89**, 022326 (2014).
[23] F. H. Watson, H. Anwar, and D. E. Browne, Phys. Rev. A **92**, 032309 (2015).
[24] M. Herold, E. T. Campbell, J. Eisert, and M. J. Kastoryano, npj Quantum Information **1**, 15010 (2015).
[25] M. Herold, M. J. Kastoryano, E. T. Campbell, and J. Eisert, arXiv preprint arXiv:1511.05579 (2015).
[26] J. Emerson, M. Silva, O. Moussa, C. Ryan, M. Laforest, J. Baugh, D. G. Cory, and R. Laflamme, Science **317**, 1893 (2007).
[27] C. Dankert, R. Cleve, J. Emerson, and E. Livine, Phys. Rev. A **80**, 012304 (2009).
[28] S. Bravyi, M. Suchara, and A. Vargo, Phys. Rev. A **90**, 032326 (2014).
[29] S. J. Denny, J. D. Biamonte, D. Jaksch, and S. R. Clark, J. Phys. A **45**, 015309 (2012).
[30] M. Aguado and G. Vidal, Phys. Rev. Lett. **100**, 070404 (2008).
[31] H. C. Jiang, Z. Y. Weng, and T. Xiang, Phys. Rev. Lett. **101**, 090603 (2008).
[32] I. Piorn, L. Wang, and F. Verstraete, Phys. Rev. A **83**, 052321 (2011).
[33] R. Orús, Phys. Rev. B **85**, 205117 (2012).
[34] A. Werner, D. Jaschke, P. Silvi, M. Kliesch, T. Calarco, J. Eisert, and S. Montangero, Phys. Rev. Lett. **116**, 237201 (2016).
[35] Z. Y. Xie, J. Chen, M. P. Qin, J. W. Zhu, L. P. Yang, and T. Xiang, Phys. Rev. B **86**, 045139 (2012).