Quantum Science and Technology

RECEIVED
16 February 2018

ACCEPTED FOR PUBLICATION
14 May 2018

PUBLISHED
6 June 2018

PAPER

Lattice surgery on the Raussendorf lattice

Daniel Herr[1,2], Alexandru Paler[3], Simon J Devitt[4,5] and Franco Nori[1,6]

[1] Theoretical Quantum Physics Laboratory, RIKEN Cluster for Pioneering Research, Saitama 351-0198, Japan
[2] Computational Physics, ETH Zurich, 8093 Zurich, Switzerland
[3] Linz Institute of Technology, Johannes Kepler University, Linz, A-4040, Austria
[4] Centre for Quantum Software & Information (QSI), Faculty of Engineering & Information Technology, University of Technology Sydney, Sydney, NSW 2007, Australia
[5] Turing Inc., Berkley CA 94705, United States of America
[6] Department of Physics, University of Michigan, Ann Arbor, MI 48109–1040, United States of America

E-mail: daniel.herr@riken.jp

Keywords: quantum computing, fault tolerance, lattice surgery, braiding

Abstract

Lattice surgery is a method to perform quantum computation fault-tolerantly by using operations on boundary qubits between different patches of the planar code. This technique allows for universal planar code computation without eliminating the intrinsic two-dimensional nearest-neighbor properties of the surface code that eases physical hardware implementations. Lattice surgery approaches to algorithmic compilation and optimization have been demonstrated to be more resource efficient for resource-intensive components of a fault-tolerant algorithm, and consequently may be preferable over braid-based logic. Lattice surgery can be extended to the Raussendorf lattice, providing a measurement-based approach to the surface code. In this paper we describe how lattice surgery can be performed on the Raussendorf lattice and therefore give a viable alternative to computation using braiding in measurement-based implementations of topological codes.

1. Introduction

Fault-tolerant methods allow for quantum computation on systems that are prone to errors. The surface code is one of the most attractive choices for fault tolerance due to its nearest-neighbor interactions and its high error-threshold [1, 2]. For surface code-based architectures, qubits can be implemented using various approaches with different methods of computation [3]. Among these are, for example, defects [2] or twists [4] where computation is performed using braiding, or planar code patches where computation is performed using lattice surgery [5].

For many implementations of physical qubits, the surface code is the method of choice, but for linear-optics quantum computation [6–8] or other hardware architectures that utilize probabilistic connections [9] a measurement-based approach [10] is the better choice. The Raussendorf lattice [11] is a measurement-based approach to the surface code and, thus, the two methods have the same benefits in terms of fault-tolerant thresholds, ability to perform a universal set of gates, and ability to largely detach the specifics of an algorithmic implementation from the underlying physical hardware.

While braiding has been the method of choice for performing fault-tolerant computation, recently lattice surgery has been investigated by several works [12–17]. It has also been extended to different implementations than the surface code, such as the color-codes [18]. For the Raussendorf lattice, however, only braiding has been investigated in depth. This paper closes this gap by describing how lattice surgery can be performed on the Raussendorf lattice.

First we give brief reviews on measurement-based quantum computation and error correction using both the surface code and the Raussendorf lattice. Furthermore, we show the translation between these two error correction methods. Then, the elementary operations using lattice surgery on the Raussendorf lattice are described.

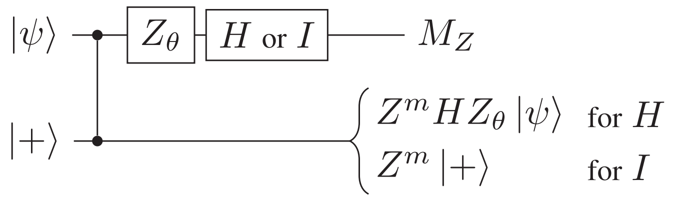

Figure 1. This figure shows the circuit which teleports the initial state $|\psi\rangle$ to the next qubit in measurement-based quantum computing. The initialization step and controlled-phase gate, corresponds to the creation of a graph state. This is followed by a measurement operation which teleports the information from one qubit to the next. Any quantum algorithm can be implemented using a sequence of these teleportation operations. Here $Z = Z_{\theta=\pi}$, H is a Hadamard gate, I is the identity and M_Z is a measurement in Z-basis.

2. Brief review

In this first part we will briefly describe how computation can be performed using a special graph state configuration called the Raussendorf lattice. We will then motivate the link between the Raussendorf lattice and surface code quantum computation. This will provide the necessary background in order to translate lattice surgery to the Raussendorf lattice.

2.1. Graph states

Graph states can be treated as resources which are used in measurement-based quantum computation [19] in order to perform information processing. The name of graph states stems from an underlying undirected graph $G = (V, E)$ whose vertices $V \subset \mathbb{N}$ represent individual qubits. The set of edges $E \subset [V]^2$ in this graph indicates entanglement between its vertices.

Given a graph, one can obtain a physical state by first initializing all qubits (vertices) to $|+\rangle$. For each edge a controlled-phase gate has to be applied on its two vertices:

$$= \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & -1 \end{bmatrix}.$$

An alternative way of looking at this graph state is given in terms of stabilizer measurements. The graph state is the simultaneous eigenstate with eigenvalue $+1$ for stabilizers S_i with $i \in V$ given by

$$S_i = X_i \prod_{j \in \mathrm{Nbh}(i)} Z_i \qquad \text{for all} \quad i \in V. \tag{1}$$

Here, the definition of the neighborhood of a graph is given by $\mathrm{Nbh}(i) = \{j|(i, j) \in E\}$.

2.2. Measurement-based quantum computation

The basic idea in measurement-based quantum computation is to to perform single-qubit measurements on this graph state. These act as teleportation operations which move the quantum state from one qubit to the next while by-product operations are applied. In figure 1 the circuit for such an operation is shown. One should note that the first two steps, the initialization and the controlled-phase gate, correspond to the graph state generation. The remaining gates can be combined into a measurement in an arbitrary basis. The rotation $Z_\theta = \exp(\mathrm{i}\theta Z/2\pi)$ is included to perform non-clifford gates on individual qubits. Finally, measurements teleport the information from one qubit to the next. Hadamard and Z_θ operations can also be combined into a single rotated-basis measurement. Hereafter, we will always treat those operations as a single rotated-basis measurement.

However, depending on the outcome of this measurement erroneous Pauli-operators are applied and need to be tracked classically. Since these measurements are probabilistic, a classical algorithm needs to track the outcomes of previous measurements and change the basis of subsequent measurements. Furthermore, a by-product Hadamard operation is applied to the state after each teleportation operation. This causes the state to change from Z-basis to X-basis after an odd number of measurements and changes back to Z-basis for an even number of measurements.

A measurement in the Z-basis without the application of the Hadamard operation removes the measured qubit from the lattice and no information will be teleported through this qubit. Thus, a measurement in this basis can be used to completely isolate different parts of a graph state from each other. Introducing defects to the lattice with Z-basis measurements is the main idea to implement logical qubits.

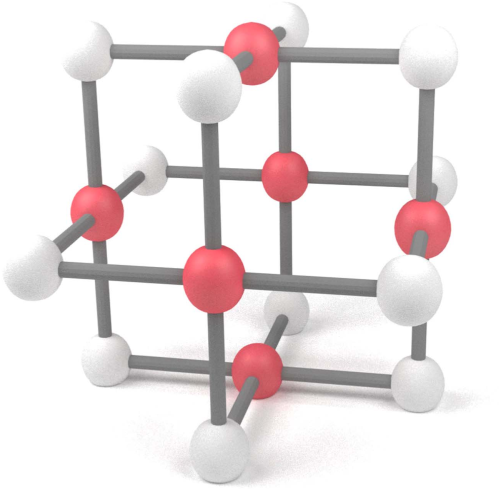

Figure 2. Unit cell of the Raussendorf lattice: the spheres represent individual photons and the connections between them show the entanglement given by the definition of a graph states. The vertices in the middle of the unit cell's faces are colored in red and contribute to a single parity check. The qubits colored in white are in the middle of dual lattice faces which also correspond to parity check operations.

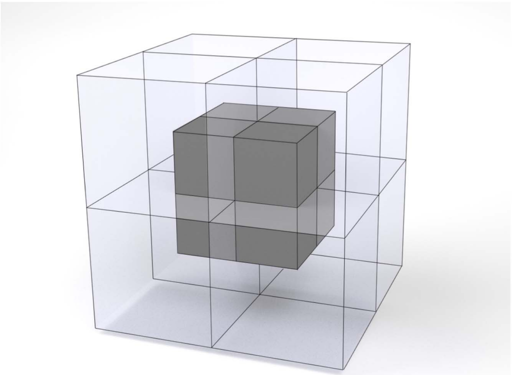

Figure 3. The dual unit cell is represented by the dark-gray box in the center. It can be created by stacking 8 primal unit cells (translucent boxes) together.

2.3. The Raussendorf lattice and the surface code

The Raussendorf lattice [20] is a 3D graph state that possesses a specific lattice structure. Figure 2 shows a unit cell of this lattice. With the stabilizer definition given in equation (1) one can see that the product of all stabilizers from the qubits colored in red corresponds to a simultaneous X-parity check of all the faces. Thus, we can define the X-stabilizer measurements for the Raussendorf lattice. This means that if no error occurred, this stabilizer measurement should give a parity of $+1$.

In order to introduce Z-stabilizers of the Raussendorf lattice, we need to consider the dual lattice of the Raussendorf lattice (as opposed to the primal lattice, that has been considered so far). This is a self-similar lattice that is shifted by $\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$. It is visualized in figure 3, where the translucent boxes are given by primal unit cells and the dark box represents the dual unit cell. The faces of the dual cell correspond to the edges of the primal cells and give rise to chains of Z-stabilizers.

If an error occurs, two of these stabilizer measurements will show a parity of -1. From these syndromes one can deduce which error occurred and how to correct for it.

2.4. Planar code

The planar code is a 2D error-correcting code whose syndromes are continuously measured to detect errors. The only difference to the surface code is how the boundary is treated. A layout of the planar code can be seen in figure 4. The qubits in this figure can be divided into two categories: syndrome qubits which are continuously

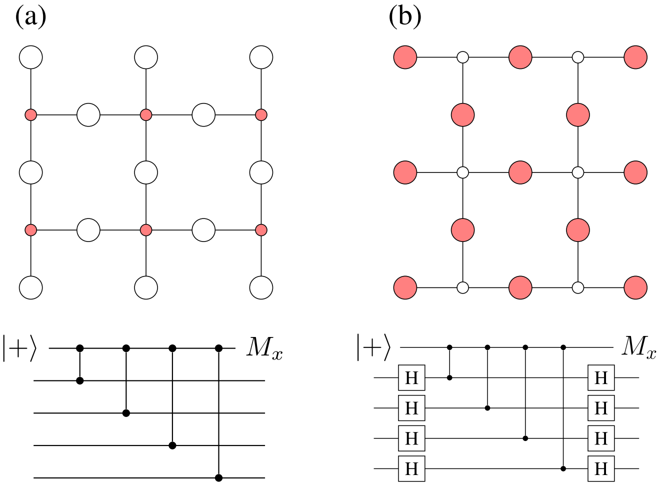

Figure 4. These figures show the two stabilizer measurements of the surface code. The small circles correspond to ancilla qubits which measure the error syndromes. In (a) only Z-stabilizers are shown, whereas in (b) only X-stabilizers are shown. Merging these two stabilizers gives the complete error detection for the surface code. However, this example, where the different stabilizers are treated separately, illustrates the connection between the surface code and the Raussendorf lattice. This can be seen by looking at even (a) and odd (b) time-slices in the Raussendorf lattice. Below the lattice depictions, a single stabilizer measurement between the ancillary qubit $|+\rangle$ (top) and its surrounding data qubits is shown for both cases.

measured and thus help detect errors, and data qubits which store the logical state of the system. Furthermore, there are two types of syndrome measurements in different bases.

The equivalence between surface codes and the Raussendorf lattice has already been shown in [21, 22] and generalized to arbitrary Calderbank–Shor–Steane stabilizer states in [23]. Therefore, we will only motivate why this equivalence holds and introduce quantities that we rely on by translating lattice surgery to the Raussendorf lattice.

To show this equivalence, we will now describe how to obtain the Raussendorf lattice from a planar code. Figure 4 shows Z-stabilizer measurements in (a) and X-stabilizer measurements in (b), which were separated into two distinct time steps. Alternating between the lattices (a) and (b) and connecting data qubits in neighboring time-slices, one obtains the Raussendorf lattice. Thus, every even time-slice in the Raussendorf lattice can be associated with Z-stabilizer measurements and every odd time-slice can be associated with X-stabilizer measurements.

In the Raussendorf lattice, one can explain this procedure in terms of teleportation. The nodes in each time-slice get measured in the X-basis and the logical state is teleported to the next time-slice. This is the reason for the terminology of 'time-slices'. Looking at the circuit diagrams for the planar code stabilizer checks shown in figure 4, one can already see the similarity between the definition of graph states and these stabilizers. Each Z-stabilizer adds a red-colored ancillary qubit to the graph using controlled-phase gates and the parity is measured afterwards. The same happens to the X-stabilizers where Hadamard operations on the data qubits are required for a change in basis. These Hadamard operations are readily obtained due to the teleportation protocol described in figure 1. Thus, in each odd time-slice a Hadamard operator has been added to the data qubits and the basis transformation follows naturally from the teleportation rules.

3. Lattice surgery on the Raussendorf lattice

Now that the equivalence between the surface code and the Raussendorf lattice has been motivated, we can translate the lattice surgery protocols. Lattice surgery is composed of many planar code patches. Each of these planar code patches encodes one logical qubit and interactions between these qubits are performed using boundary operations between neighboring patches. These boundary operations are called merge and split operations and act as parity checks. In addition to these operations, state injection is needed to perform gates that are not supported by surface codes. With these, universality is achieved. A description within the context of planar codes was given in the original paper for lattice surgery [5]. Or description is different to this paper due to an additional physical dimension. This causes the planar code patches to become boxes and their edges to become faces.

The key idea of our approach is to carve out boxes from the fully-connected Raussendorf lattice which is given by the hardware. These boxes each represent a logical qubit, as did patches in the planar code. Merges

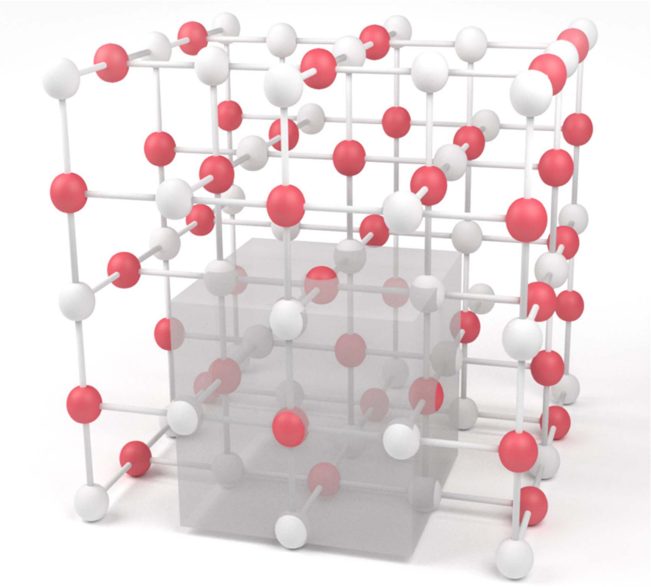

Figure 5. 3D representation of a Raussendorf lattice box of the patch illustrated in figure 4. The gray area corresponds to a single unit cell which was illustrated in 2. To separate this box from the surroundings, all qubits around the box need to be measured in the Z-basis.

occur when we stop carving out boxes and splits occur when we introduce more cuts into the lattice to split a box into separate logical boxes. Below, we describe in detail all necessary operations for universality.

3.1. Boxes inside the Raussendorf lattice

The qubits of the Raussendorf lattice have to be measured in the Z-basis if they do not belong to a logical qubit box. Conversely, nodes that contribute to a logical qubit need to be measured in the X-basis. Thus, a single box inside the Raussendorf lattice can be obtained by measuring all qubits surrounding it in the Z-basis. In figure 5 one such box is shown. All visible qubits are measured in the X-basis and the surrounding qubits have to be measured in the Z-basis. There is a transition that changes X-measurements of the box into Z-measurements, which disentangle the box from its surroundings. Depending on where this transition is, one can distinguish two types of faces: rough and smooth faces. A rough face is cutting through primary unit cells, whereas a smooth face is composed of faces of the unit cells. From figure 5 one can see that a single box consists of two smooth faces vertically, and two rough faces horizontally. The faces in the time direction need to be determined during initialization and measurement.

In figure 6 we show how four of these patches can be arranged in the large Raussendorf lattice. The qubits on the boundary are measured in Z-basis, which completely disentangles each patch from the others. Boundary operations such as merges will be described later and use these qubits to add interactions between neighboring qubits.

3.2. Logical Pauli-operators

Logical X- and Z-operators in the original lattice surgery description can be realized by performing a physical X or Z-operation along chains spanning from opposite edges of the patch/box. While these operations could in principle also be implemented physically, they do not need to be applied because classical software can permute these operations and reinterpret measurement results.

Logical X- or Z-operations can be treated the same way teleportation errors are handled on a physical qubit level. Due to by-product Pauli operations, figure 1 shows, that with a probability of 50%, teleport operations of physical qubits have an additional error-operator. The outcome of our measurement indicates whether this error-operator was applied or not. Thus, we can treat a logical X-operation by inverting the measurement results along a chain on primary faces. The Z-operation can be performed by inverting the measurement results along a chain on a time-slice of dual faces. The logical X- and Z-operators are shown in figure 7.

3.3. Initialization and measurement

In the original description of lattice surgery [5] patches could be initialized into the logical $|0\rangle$ state by preparing all physical qubits in $|0\rangle$. Similarly, the logical $|+\rangle$ state can be obtained by preparing all physical qubits in $|+\rangle$. In the Raussendorf lattice a patch can be created by changing the measurements from Z- to X-basis. Therefore,

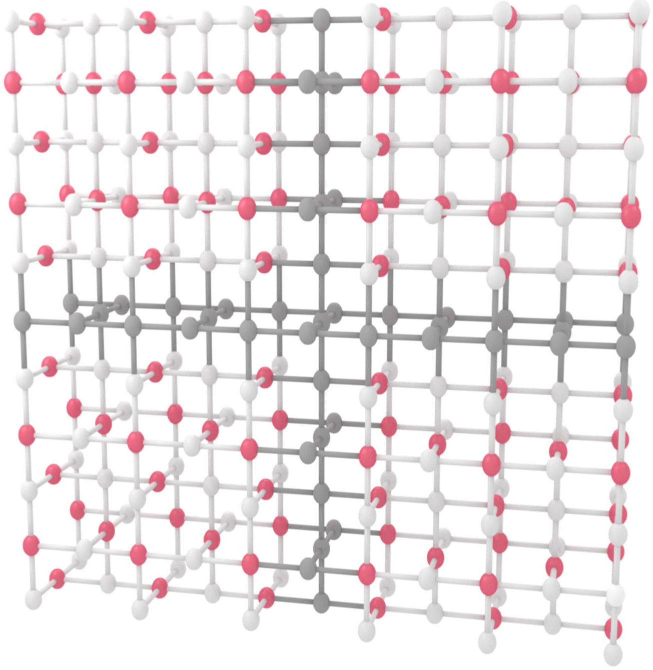

Figure 6. Four planar code patches are embedded in the Raussendorf lattice with Z-basis measurements on the boundary between the patches (dark color).

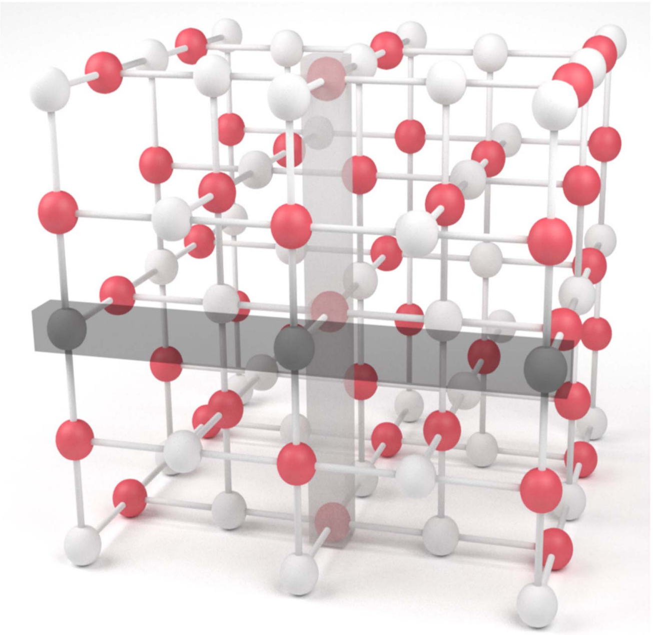

Figure 7. Logical X and Z-operators are chains of physical X or Z-operations. For a logical Z-operation this chain runs from one rough edge to the opposing rough edge (horizontally), whereas for X-operations it runs from smooth edge to smooth edge (vertically).

these measurements stop removing qubits from the graph and start connecting subsequent qubits to each other. Due to the definition of graph states, physical qubits are always prepared in a $|+\rangle$ state. Despite this constraint, we are still able to initialize the system into the logical $|0\rangle$ or $|+\rangle$ state. The important point here is in which time-slice the measurements are changed. If the patch is created on the faces of the primal lattice (even time-slices), one will end up with a logical $|+\rangle$ state, because this would correspond to a $|+\rangle$ state initialization of each qubit. If one chooses to switch the measurement basis on the faces of the dual lattice, each qubit will be teleported to the next primal nodes with an additional Hadamard operator (see 1) and therefore this will result in an initialization of the logical $|0\rangle$ state.

For measurements on logical qubits, the inverse of the initialization has to be performed. Thus, one will switch the measurement basis from X-measurements to Z-measurements. Again, the time-slice this happens determines whether an additional Hadamard gate was applied and whether the basis of the measurement was X or Z. For a measurement in the Z-basis one needs to measure an even time-slice, which corresponds to a measurement along primal faces. A measurement in the X-basis has to occur on an odd time-slice, which corresponds to a measurement-plane along dual faces. The measurement result can be inferred from the total parity along a chain of measurements spanning from one edge to the next. Because of the logical operators shown in figure 7 these chains need to be oriented horizontally (vertically) for a logical Z-measurement (logical X-measurement).

3.4. Merges and splits

In the following we describe how merge and split operations from the original proposal [5] can be adapted to the Rausendorf lattice.

Measurements on boundary qubits between the two boxes need to be switched from Z to X-measurements to perform a merge operation. Data qubits on this connecting edge need to be initialized in $|0\rangle$ ($|+\rangle$) for a merge along rough (smooth) boundaries. This requires again to swap the measurement basis at different times for smooth and rough merges. For a smooth merge, one needs to swap the basis starting at a time-slice that is made of primal faces, while for a rough merge this needs to happen on a time-slice composed of dual faces. A visualization of this can be seen in figure 8.

The merge operation acts as a parity measurement on the two logical qubits. For a rough merge this parity corresponds to simultaneous X-measurement (XX) and for a smooth merge it corresponds to a ZZ-measurement on the logical states. To obtain the measurement outcome of this parity check, the faces along the boundary between the two patches have to be checked. Because of this parity measurement, the output state is non-deterministic, and the number of logical qubits is decreased by one.

The output state of a rough merge between logical qubits $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ and $|\phi\rangle = \alpha'|0\rangle + \beta'|1\rangle$ can be summarized in the following equation [5]:

$$\begin{aligned}|\psi\rangle \text{Ⓜ}_r|\phi\rangle &= \alpha|\phi\rangle + (-1)^M\beta|\overline{\phi}\rangle\\ &= \alpha'|\psi\rangle + (-1)^M\beta'|\overline{\psi}\rangle,\end{aligned}$$

where $M = \{0, 1\}$ is the measurement outcome of the parity check and $|\overline{\phi}\rangle = \sigma_x|\phi\rangle$.

A smooth merge between between logical qubits $|\psi\rangle = a|+\rangle + b|-\rangle$ and $|\phi\rangle = a'|+\rangle + b'|-\rangle$ can be summarized in the following equation [5]:

$$\begin{aligned}|\psi\rangle \text{Ⓜ}_r|\phi\rangle &= a|\phi\rangle + (-1)^M b|\overline{\phi}\rangle\\ &= a'|\psi\rangle + (-1)^M b'|\overline{\psi}\rangle,\end{aligned}$$

where M again gives the measurement result and $|\overline{\phi}\rangle = \sigma_z|\phi\rangle$.

3.5. Encoding states

To be able to perform universal computation, state injection has to be performed. This requires specially prepared states as a resource. Encoding these logical states is done by using faulty single-qubit operations which require a distillation of the resulting logical state to reduce errors. Only then can these states be used as a resource for state injection. While distillation and injection itself are done on a purely logical level, a method for encoding arbitrary states needs to be devised for the Raussendorf lattice. For injection, it is common to use the following two states, because they complement the surface code and enable universality:

$$\begin{aligned}|Y\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + \mathrm{i}|1\rangle)\\ |A\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + \mathrm{e}^{\mathrm{i}\frac{\pi}{4}}|1\rangle).\end{aligned} \tag{2}$$

The $|Y\rangle$ state will mainly be used for P-gates and Hadamard operations while the $|A\rangle$ state is needed for the implementation of the T-gate. The procedure to encode logical states is based on the original lattice surgery description which we modified to be implementable in the Raussendorf lattice. It can be summarized in the following steps:

1. A single physical qubit is prepared as a magic state: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $|\psi\rangle$ is equal to either $|A\rangle$ or $|Y\rangle$ described in equation (2).
2. Using CNOTs, a 3-qubit entangled state is created resulting in $|\psi\rangle = \alpha|000\rangle + \beta|111\rangle$.
3. These three qubits are used as data qubits along a logical Z-operator chain for a distance-3 planar patch.

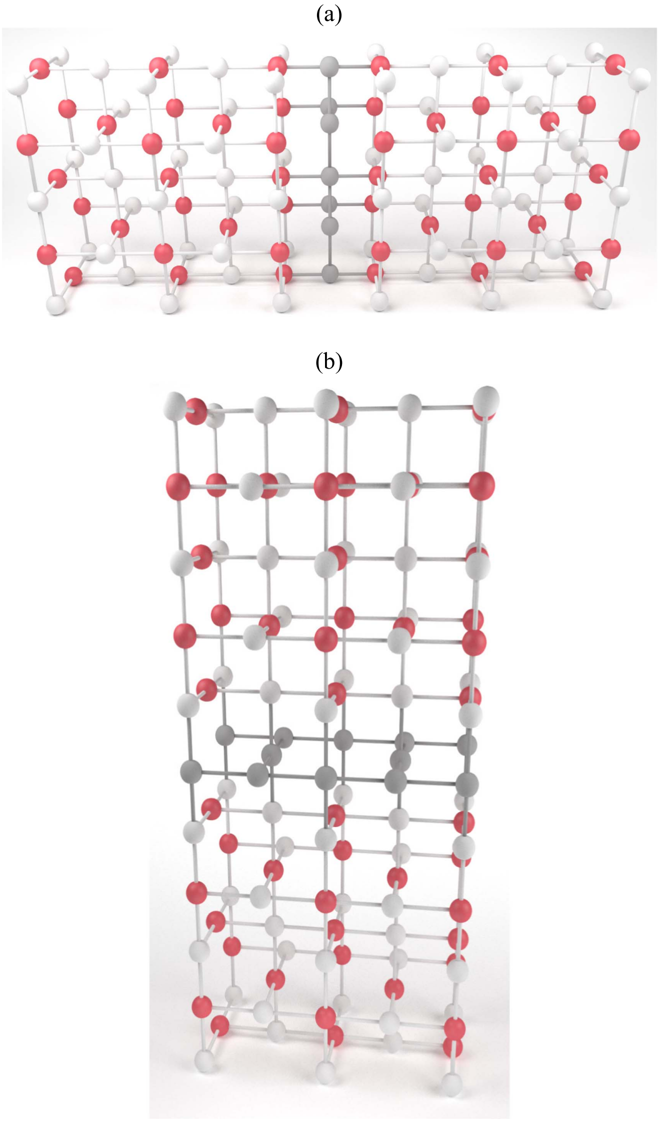

Figure 8. During a merge operation the measurement basis along the edge between two LS patches is turned on. In (a) this is shown for a rough merge, where each qubit along the boundary needs to be initialized in the $|0\rangle$ state. Thus, the measurement has to be flipped on an odd time-slice. In (b) this procedure is done for a smooth merge. Here the qubits on the boundary need to be initialized in the $|+\rangle$ state and thus the measurement basis needs to be flipped in an even time-slice.

4. Then, with all other qubits of this patch initialized to $|0\rangle$, syndrome measurements are turned on.

The resulting state of this procedure is a logical distance-three planar patch that encodes an erroneous magic state. Using merges, this patch can be brought to the desired code distance.

To implement this procedure in the Raussendorf lattice, a single-qubit needs to be prepared in a magic state. This is where a measurement in an arbitrary basis comes into play (see figure 1). Figure 9 shows the method of encoding an arbitrary magic state in a distance-three box. The measurement basis has to be changed from Z- to X-basis on primal faces for five qubits. Here, a vertical line of qubits is measured in the X-basis, except for the center qubit (indicated by the red color in figure 9). This qubit is measured in a rotated-basis. In order to inject a $|Y\rangle$ state (needed for the S-gate) a rotated-basis measurement with $\theta = \pi/2$ needs to be chosen. For a T-gate, the $|A\rangle$ state has to be prepared by a rotated measurement of $\theta = \pi/4$.

The center qubit and the outer qubits (all colored qubits in figure 9) therefore teleport the state from step 2: $|000\rangle + \exp(\mathrm{i}\theta/2\pi)|111\rangle$ to the next time-slice. In the second time-slice the measurement is performed in the X-basis for all qubits. This would correspond to an initialization of 0 for the remaining data qubits. Error correction on this distance-3 code is now possible, and using merges this state can be brought to arbitrary sizes.

This procedure encoded an arbitrary but faulty state in a logical qubit. Magic state distillation algorithms [24, 25] can now be used to reduce the noise and teleportation protocols can then implement S- and T-gates. Even, the synthillation protocols given in [26] can be used here.

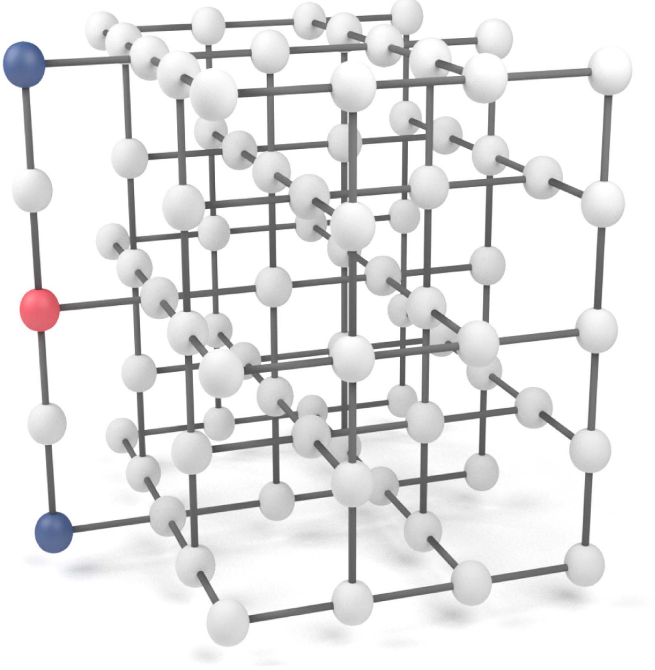

Figure 9. Here, a magic state is encoded into a distance-three planar code, within the Raussendorf lattice. The time-slices are measured from the left to the right. The first measurements happen on the column that has the red-colored qubit in the middle. All qubits are measured in the standard X-basis, only the qubit colored in red is measured in a rotated-basis. This results in a superposition state of the red and blue qubits, which is due to the measurements teleported back one layer. All other qubits need to be initialized to the $|0\rangle$ state. This is achieved by switching from a Z-basis to a X-basis measurement on an odd time-slice (dual faces).

3.6. Hadamard

A logical Hadamard operation is also possible. We present a modified version of the initial transversal Hadamard operation devised for the surface code [2] and lattice surgery [5]. However, in order to recombine the modified logical state with its surroundings, we need to employ operations that are not native to the Raussendorf lattice itself. Looking at the lattice generation protocol of [8] we can modify the lattice by choosing a different method for the lattice renormalization.

Our proposal for a logical Hadamard operation starts with a physical Hadamard operation applied to each data qubit. This is achieved by connecting primal nodes with primal nodes as shown in figure 10. Due to the missing temporal layer of nodes, an additional by-product Hadamard is applied. Furthermore, data qubits are shifted by half a unit cell down and left. This is required in order to be able to recombine the lattice with other patches later on.

In comparison to figure 7, their logical operators have now changed the measurement basis for physical qubits. This means that the logical Z-operator is a horizontal chain of physical X-operations instead of physical Z-operations. The logical X-operator is now described by a chain of vertical Z-operations.

To complete the Hadamard translation, this patch has to be rotated. In the original lattice surgery proposal [5] such a rotation has already been proposed and can simply be translated to the Raussendorf lattice. In figure 11 this rotation is performed by switching the measurement basis on dual faces from Z to X for all qubits depicted in dark-gray. (This corresponds to a $|0\rangle$-initialization for qubits on the surface code.)

The resulting patch is still a distance-d planar code. After d-rounds of error correction all qubit measurements that do not form a square with the dark-colored ones can be turned to Z-basis measurements. This results in a rotated patch, such that a previous horizontal logical operator is now vertical and vice versa. Additionally, this rotation cancels the shift by half a unit cell that was introduced before, such that the patch is back in proper alignment with the rest of the Raussendorf lattice.

The logical Z-operator, that previously was a horizontal chain of physical Z-operators, has now been mapped to a chain of vertical X-operations. Thus, the logical Z and X operators have been swapped and a proper Hadamard operation has been applied.

3.7. Hadamard and ICM

For some hardware models, connections between primal nodes might not be allowed. Thus, we present another approach which is based on the decomposition of a Hadamard into rotational gates $H = R_z\left(\frac{\pi}{2}\right)R_x\left(\frac{\pi}{2}\right)R_z\left(\frac{\pi}{2}\right)$.

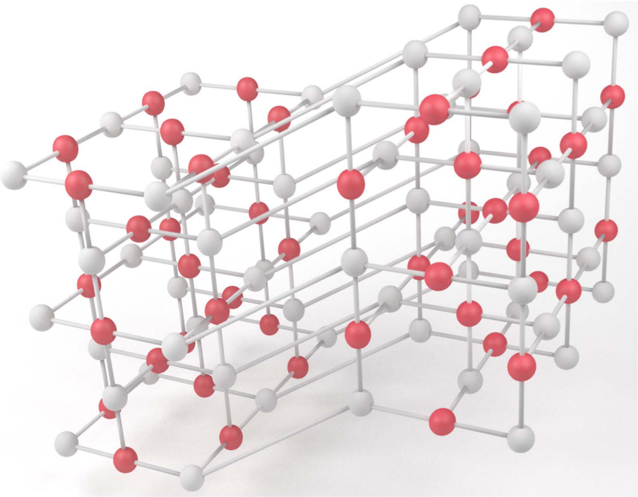

Figure 10. This figure shows how data qubits from two primal faces can be connected. The nodes from the dual lattice are removed during the lattice generation or renormalization step. This structure archives two goals. First, it applies a Hadamard operation to each of the data qubits. Second, the data qubits are shifted by $(-1/2,\ -1/2)$, which allows the recombination with the rest of the Raussendorf lattice, when the Hadamard operation is finished.

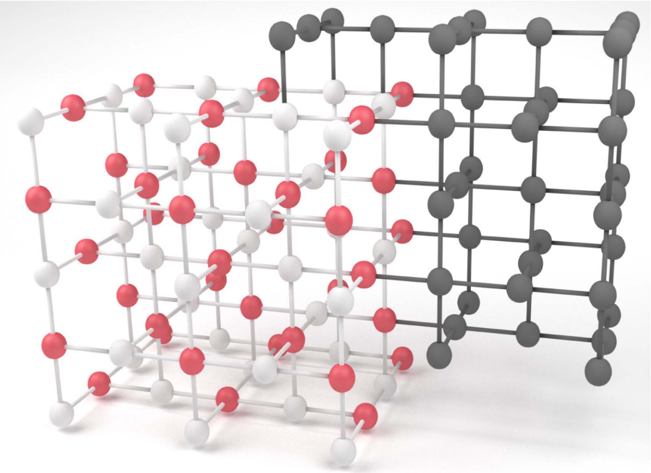

Figure 11. This procedure rotates the patch by $\pi/2$. The chains of logical operators are rotated such that they are facing the same direction as logical operators of other patches.

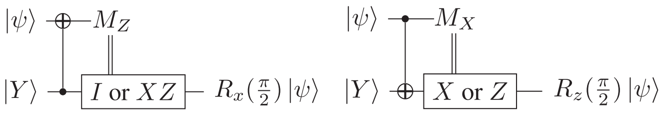

Figure 12. Teleportation operations needed to implement a Hadamard operation. The circuit to the left implements a X-rotation and the circuit to the right implements a Z-rotation.

These rotational gates can be implemented using the measurement procedure shown in figure 12, which treats the application of Hadamards on a purely logical level.

Unfortunately, this procedure needs state injection and is more costly due to the distillation protocols that are required. Once the necessary $|Y\rangle$ states have been encoded and distilled, the teleportation protocols given in figure 12 will implement the rotation operations required by the Hadamard operation.

A convenient representation that translates any circuit using teleportation gates into a deterministic circuit is the ICM model [27, 28], which stands for (I)nitialization, (C)NOT and (M)easurements. In this representation, any operation is decomposed into teleportation operations, and non-deterministic results are handled using selective target and selective source teleportation methods [28, 29]. This results in a deterministic circuit, despite being dependent on the probabilistic outcome of its teleportation operations.

A further extension of this representation is the inverted ICM representation [13]. The difference between ICM and inverted ICM is that the former has arbitrary basis initialization and restricted-basis measurements, whereas the latter has only a restricted initialization (Z or X) and arbitrary basis measurements. In the inverted ICM representation, an error-corrected graph state which is specific to the algorithm is created and can be readily realized by merges and splits of lattice surgery. Afterwards, measurements perform the computation of the quantum algorithm. A complete discussion is available in [13].

4. Conclusion

So far the literature has only described braiding as a method of computation on the Raussendorf lattice. In this paper we showed that a lattice surgery implementation can also perform quantum computation in the Raussendorf lattice and we described how to implement all fundamental operations. An implementation of transversal Hadamards proved to be hard without changing the lattice structure. We gave two approaches, both with different drawbacks. In conclusion, lattice surgery can be implemented on the Raussendorf lattice such that future quantum computation [30] on the Raussendorf lattice can use the beneficial aspects of lattice surgery.

Acknowledgments

DH is supported by the RIKEN IPA program and AP is supported by the Linz Institute of Technology project CHARON, grant number LITD13361001.

SJD acknowledges support from the JSPS Grant-in-aid for Challenging Exploratory Research and from the Australian Research Council Centre of Excellence in Engineered Quantum Systems EQUS (Project CE110001013).

FN is partially supported by the MURI Center for Dynamic Magneto-Optics via the AFOSR Award No. FA9550-14-1-0040, the Army Research Office (ARO) under grant number 73315PH, the AOARD grant No. FA2386-18-1-4045, the CREST Grant No. JPMJCR1676, the IMPACT program of JST, the RIKEN-AIST Challenge Research Fund, the JSPS-RFBR grant No. 17-52-50023, and the Sir John Templeton Foundation.

ORCID iDs

Daniel Herr https://orcid.org/0000-0002-0849-8339
Alexandru Paler https://orcid.org/0000-0002-1536-8858
Simon J Devitt https://orcid.org/0000-0002-5901-1391
Franco Nori https://orcid.org/0000-0003-3682-7432

References

[1] Wang D S, Fowler A G and Hollenberg L C L 2011 Surface code quantum computing with error rates over 1% *Phys. Rev.* A **83** 020302
[2] Fowler A G, Mariantoni M, Martinis J M and Cleland A N 2012 Surface codes: towards practical large-scale quantum computation *Phys. Rev.* A **86** 032324
[3] Brown B J, Laubscher K, Kesselring M S and Wootton J R 2017 Poking holes and cutting corners to achieve Clifford gates with the surface code *Phys. Rev.* X **7** 021029
[4] Bombin H 2010 Topological order with a twist: ising anyons from an Abelian model *Phys. Rev. Lett.* **105** 030403
[5] Horsman C, Fowler A G, Devitt S and Meter R V 2012 Surface code quantum computing by lattice surgery *New J. Phys.* **14** 123011
[6] Knill E, Laflamme R and Milburn G J 2001 A scheme for efficient quantum computation with linear optics *Nature* **409** 46–52
[7] Kieling K, Rudolph T and Eisert J 2007 Percolation, renormalization, and quantum computing with nondeterministic gates *Phys. Rev. Lett.* **99** 130501
[8] Herr D, Paler A, Devitt S J and Nori F 2017 A local and scalable lattice renormalization method for ballistic quantum computation arXiv:1710.04387
[9] Nemoto K, Trupke M, Devitt S J, Stephens A M, Scharfenberger B, Buczak K, Nöbauer T, Everitt M S, Schmiedmayer J and Munro W J 2014 Photonic architecture for scalable quantum information processing in diamond *Phys. Rev.* X **4** 031022
[10] Raussendorf R and Briegel H J 2001 A one-way quantum computer *Phys. Rev. Lett.* **86** 5188–91
[11] Raussendorf R, Harrington J and Goyal K 2006 A fault-tolerant one way quantum computer *Ann. Phys.* **321** 2242
[12] Yoder T J and Kim I H 2017 The surface code with a twist *Quantum* **1** 2
[13] Herr D, Nori F and Devitt S J 2017 Lattice surgery translation for quantum computation *New J. Phys.* **19** 013034
[14] Herr D, Nori F and Devitt S J 2017 Optimization of lattice surgery is NP-hard *NPJ Quantum Inf.* **3** 35
[15] Litinski D and Oppen F V 2017 Lattice surgery with a twist: simplifying clifford gates of surface codes *Quantum* **2** 62
[16] Nagayama S, Satoh T and Van Meter R 2017 State injection, lattice surgery, and dense packing of the deformation-based surface code *Phys. Rev.* A **95** 012321

[17] Nautrup H P, Friis N and Briegel H J 2017 Fault-tolerant interface between quantum memories and quantum processors *Nat. Commun.* **8** 1321

[18] Landahl A J and Ryan-Anderson C 2014 Quantum computing by color-code lattice surgery arXiv:1407.5103

[19] Raussendorf R, Browne D E and Briegel H J 2003 Measurement-based quantum computation with cluster states *Phys. Rev.* A **68** 022312

[20] Raussendorf R, Harrington J and Goyal K 2006 A fault-tolerant one-way quantum computer *Ann. Phys., NY* **321** 2242–70

[21] Raussendorf R and Harrington J 2007 Fault-tolerant quantum computation with high threshold in two dimensions *Phys. Rev. Lett.* **98** 190504

[22] Briegel H J, Browne D E, Dur W, Raussendorf R and Van den Nest M 2009 Measurement-based quantum computation *Nat. Phys.* **5** 19–26

[23] Bolt A, Duclos-Cianci G, Poulin D and Stace T M 2016 Foliated quantum error-correcting codes *Phys. Rev. Lett.* **117** 070501

[24] Bravyi S and Haah J 2012 Magic-state distillation with low overhead *Phys. Rev.* A **86** 052329

[25] Campbell E T, Anwar H and Browne D E 2012 Magic-state distillation in all prime dimensions using quantum reed-muller codes *Phys. Rev.* X **2** 041021

[26] Campbell E T and Howard M 2017 Unifying gate synthesis and magic state distillation *Phys. Rev. Lett.* **118** 060501

[27] Paler A, Devitt S J, Nemoto K and Polian I 2014 Mapping of topological quantum circuits to physical hardware *Sci. Rep.* **4** 4657

[28] Paler A, Polian I, Nemoto K and Devitt S J 2017 Fault-tolerant, high-level quantum circuits: form, compilation and description *Quantum Sci. Technol.* **2** 025003

[29] Paler A, Devitt S J and Fowler A G 2016 Synthesis of arbitrary quantum circuits to topological assembly *Sci. Rep.* **6** 30600

[30] Rudolph T 2017 Why I am optimistic about the silicon-photonic route to quantum computing *APL Photonics* **2** 030901