Adiabatic graph-state quantum computation

B Antonio[1], D Markham[2] and J Anders[1,3]

[1] Department of Physics and Astronomy, University College London, Gower Street, London WC1E 6BT, UK
[2] CNRS LTCI, Département Informatique et Réseaux, Telecom ParisTech, 23 avenue d'Italie, CS 51327, F-75214 Paris CEDEX 13, France
[3] Department of Physics and Astronomy, University of Exeter, Stocker Road, Exeter EX4 4QL, UK
E-mail: robert.antonio.10@ucl.ac.uk

Abstract

Measurement-based quantum computation (MBQC) and holonomic quantum computation (HQC) are two very different computational methods. The computation in MBQC is driven by adaptive measurements executed in a particular order on a large entangled state. In contrast in HQC the system starts in the ground subspace of a Hamiltonian which is slowly changed such that a transformation occurs within the subspace. Following the approach of Bacon and Flammia, we show that any MBQC on a graph state with *generalized flow (gflow)* can be converted into an adiabatically driven holonomic computation, which we call *adiabatic graph-state quantum computation* (AGQC). We then investigate how properties of AGQC relate to the properties of MBQC, such as computational depth. We identify a trade-off that can be made between the number of adiabatic steps in AGQC and the norm of $\dot{H}$ as well as the degree of H, in analogy to the trade-off between the number of measurements and classical post-processing seen in MBQC. Finally the effects of performing AGQC with orderings that differ from standard MBQC are investigated.

Keywords: measurement-based quantum computation, adiabatic quantum computation, holonomic quantum computation, depth complexity, Hamiltonian complexity, quantum information, adiabatic theorem

1. Introduction

Quantum computation provides an advantage over classical computation in solving certain problems in less time. For example, known classical algorithms for factoring a number take exponential time in the number of inputs *N*, while a quantum algorithm polynomial in *N* exists [1]. Three widely studied methods for implementing universal quantum computation are circuit-based quantum computation (see e.g. [2]), in which a series of unitary gates are applied to a number of input qubits; measurement-based quantum computation (MBQC) [3], in which an entangled resource state is prepared and measurements are performed to drive a transformation on a portion of the state; and adiabatic quantum computation (AQC) [4], in which the solution to a problem is encoded in the ground state of a Hamiltonian, and this ground state is reached adiabatically (i.e. slowly with respect to the minimum energy gap) starting from the ground state of some easily prepared Hamiltonian. These models have been shown to be equivalent to each other in the following sense; for a given computation, the number of gates required in the circuit model scales polynomially with the number of measurements on a graph state in MBQC [3] (the size of the graph state) and polynomially with the inverse energy gap of the equivalent AQC computation [5].

Bacon and Flammia proposed the direct translation of MBQC on a cluster state into an adiabatically driven holonomic quantum computation (HQC) evolution [6], which they call *adiabatic cluster-state quantum computation*. In this model, the initial adiabatic Hamiltonian is made up of the stabilizers of the cluster-state and the computation proceeds by replacing the discontinuous measurements of MBQC with continuous adiabatic transformations. While the evolution is adiabatic, Bacon and Flammia's model is not an example of an AQC. In AQC the computational task is to reach the unique ground state of a problem Hamiltonian, in contrast, in Bacon and Flammia's model the ground state is degenerate, the subspace is known, but the evolution generates transformations within this degenerate subspace. This is an example of a HQC [7–9]. There are several other works combining ideas from MBQC and AQC. In ancilla-controlled adiabatic evolution [10–12], computation is carried out by a combination of adiabatic passage and measurement. In adiabatic topological quantum computation [13], defects in a topological code are adiabatically deformed to perform logical operations. There are also examples of adiabatically driven computations on non-stabilizer states such as symmetry-protected states of matter [14] and generalized cluster states [15].

Here we extend Bacon and Flammia's model to general graph states with a property called *generalized flow (gflow)*. This new *adiabatic graph-state quantum computation* (AGQC) allows us to investigate how the properties of MBQC change when we replace non-deterministic measurements by deterministic adiabatic transformations. Beyond the application of the different models, it is of general interest to have methods of translating computations from one model to another so that intuition, understanding and techniques from one model can be applied to the others.

The random outcomes of the measurements in MBQC require a classical adaption of future measurements to achieve a deterministic outcome. This interplay allows for interesting trade-offs between classical and quantum time in MBQC and has given rise to new concepts such as blind quantum computation [16] and verified universal computation [17]. Furthermore it can be used to demonstrate a gap in quantum depth complexity between MBQC and the circuit model [18]. Here we are interested in how this trade-off manifests itself in AGQC. We find that it becomes a trade-off between the number of adiabatic steps and the degree of the initial

Hamiltonian as well as the norm of the time derivative of the Hamiltonian. (The *degree* is the number of sites that each summand in the Hamiltonian acts on non-trivially.)

Whether or not large degree operators act as a useful resource in this model is important from a fundamental point of view as well as a practical one. Our results suggest that large degree operators are not a useful computational resource in this model. We see that in order to decrease the number of adiabatic steps, the Hamiltonian degree needs to rise. However, this does not bring the benefit one may hope as the time needed for adiabatic passage through each step increases by the same amount so that the total time scales the same. Furthermore, under the assumption that simulating high degree operators shrinks the energy gap (which is the case for all known methods e.g. [19, 20]), implementation using fixed degree Hamiltonians will incur prohibitive time costs. This means that the optimal way of performing MBQC does not correspond to the optimal way to perform AGQC. These results are also interesting from the perspective of a subtlety that arises within the application of the adiabatic theorem. In our computation, the time taken for each step is governed not by the energy gap (which remains constant), but the norm of differential of the Hamiltonian.

Finally we are interested in how the time order associated to computation through MBQC appears in our model. One may expect that replacing random measurements with deterministic adiabatic substitutions will allow different ordering of the computational steps. In particular Clifford operations can be done in a single step in MBQC, so we expect this behaviour to manifest itself in AGQC. However we find that, in certain cases, the computational steps can be performed in a limited order. Surprisingly the most limited ordering occurs for certain Clifford operations.

The paper is structured as follows; in section 2 we provide background on AQC, MBQC, and adiabatic cluster-state quantum computation. In section 2.1 we generalize adiabatic cluster-state quantum computation to any graph state which has *gflow*, and investigate what trade-off exists in this model in analogy to the trade-offs in MBQC. Finally we discuss the role of the ordering of measurements in AGQC in section 4 and conclude in section 5.

2. Background

2.1. Adiabatic HQC

Consider a system with a time-varying Hamiltonian $H(t)$ and ground state $|E_0(t)\rangle$. If at $t = 0$ we prepare a system in $|E_0(0)\rangle$, and change the Hamiltonian slowly enough, then at time τ we will finish in the state $|E_0(\tau)\rangle$ with high probability. In this case, 'slow' means that the evolution satisfies the adiabatic criterion [21], which roughly says that the system will remain in the ground state provided that the evolution satisfies

$$\frac{\langle E_m(t) | \dot{H}(t) | E_0(t) \rangle}{E_m(t) - E_0(t)} \ll 1 \tag{1}$$

for all m and $0 \leqslant t \leqslant \tau$. After such an adiabatic evolution, the final state is $e^{i\gamma_B(\tau)} e^{-i \int_0^\tau E(t') dt'} |E_0(\tau)\rangle$, where $-i \int_0^\tau E(t') dt'$ is the *dynamical* phase and $i\gamma_B(\tau) = -\int_0^\tau \langle E_0(t') | \dot{E}_0 \rangle dt'$ is the *Berry* or *Pancharatnam* phase [22, 23]. The dynamical phase vanishes under cyclic evolutions, where $H(0) = H(\tau)$ and can be removed by a local gauge transformation $|\tilde{E}_0(t)\rangle = e^{-i \int_0^\tau E(t') dt'} |E_0(t)\rangle$, however the Berry phase does not vanish

under cyclical evolutions or local gauge transformations, and depends only on the path taken through parameter space during the evolution. The Berry phase has proved important when describing many phenomena in condensed matter systems, such as the anomalous quantum hall effect [24].

Berry phases can also be generalized to situations where $H(t)$ has ground space of d degenerate energy levels, labelled $|E_0^\alpha\rangle$, where $1 \leqslant \alpha \leqslant d$. In this case, the same process will not in general lead to a global phase, but instead transforms the ground space by a rotation $U_{\alpha\beta}(t) = \mathcal{T} \exp\left(-\int_0^t \langle E_0^\alpha(t')|\dot{E}_0^\beta(t')\rangle \, \mathrm{d}t' \right)$. This is called a *holonomy* [25], and for systems over which we have adequate control it is possible to use these holonomies to produce universal rotations on information encoded in the ground space, and so provides a way to perform quantum computation. This method is called HQC [7, 8]. It is also possible to perform HQC, where $H(0) \neq H(\tau)$, this is known as *open-loop* HQC [9].

Typically HQC is performed using an adiabatic evolution (although this is not a necessary condition [26]), but is a very different way of computation compared to the 'standard' adiabatic quantum computation (AQC) protocol proposed by Farhi *et al* [27]. The AQC protocol starts with a system prepared in the ground state of a simple initial Hamiltonian H_0, such as a uniform magnetic field, and then the Hamiltonian is slowly changed to a complicated 'problem Hamiltonian' H_p whose ground state encodes the problem to be solved (e.g., finding the ground state of an Ising spin glass is NP-hard [28]). In HQC, it is not the ground state itself that encodes the answer to the problem but the sequence of operations that have been performed within a degenerate subspace. The eigenstates of the Hamiltonian can be completely known at all times, as can the energy gap profile. AQC has a built in noise reduction mechanism since there is always an energy gap between the ground and states, however achieving fault tolerance in AQC is still an ongoing problem (see e.g. [29]). HQC also has some gap protection (although due to the degeneracy in the ground state the protection is not the same as for AQC), and some protection from control errors (see e.g. [30]). There are also known schemes to implement fault-tolerant HQC [31].

Since HQC involves degenerate ground spaces, the form of the adiabatic theorem shown in equation (1) is not appropriate since it is derived for singly degenerate states. For the purposes of this paper the following form of the adiabatic theorem is valid [32]; consider a linear interpolation between two Hamiltonians H_0 and H_p. The time dependent Hamiltonian of this transition is $H(t) = (1 - \frac{t}{\tau})H_0 + \frac{t}{\tau}H_p$ so that the transition is finished at $t = \tau$. To simplify the notation, we introduce a parameter $s = \frac{t}{\tau}$, and we denote the eigenvalues of $H(s)$ as $E_n(s)$. Then if we start in the ground state of H_0, the final state will be ε close in the l_2 norm to the ground subspace of H_p provided the adiabatic run time τ satisfies

$$\tau \geqslant \max_{0 \leqslant s \leqslant 1}\left(\frac{c(\delta)\| \dot{H} \|^{1+\delta}}{\varepsilon \Delta^{2+\delta}} \right), \tag{2}$$

where $\Delta = \min_n |E_n(s) - E_0(s)|$, $0 < \delta \leqslant 1$, $\|M\|$ is the operator norm, defined as the largest absolute eigenvalue of M, and $c(\delta)$ is a parameter depending only on δ. Although it is tempting to set $\delta \to 0$, this isn't possible without the adiabatic time diverging, since $\lim_{\delta \to 0} c(\delta) = \infty$ [32]. So δ is taken as some fixed, small positive number.

2.2. Measurement-based quantum computation

In MBQC an entangled resource state is measured sequentially and adaptively. We consider graph states as our entangled resource states (see e.g. [33] for other possible resources). To create this resource state, qubits are prepared in a $|+\rangle = (1/\sqrt{2})(|0\rangle + |1\rangle)$ state, and controlled-phase (CZ) operations are performed between neighbouring qubits. To perform a computation on this resource state, single qubit measurements in bases $\{|+_{\theta_j}\rangle\langle+_{\theta_j}|, |-_{\theta_j}\rangle\langle-_{\theta_j}|\}$ are performed, where $|\pm_\theta\rangle := \frac{1}{\sqrt{2}}(|0\rangle \pm e^{i\theta}|1\rangle)$, and where θ_j is the measurement angle for qubit j. The measurement will have a random outcome ± 1, however by adapting future measurements on other qubits this randomness can be corrected for. The deterministic output of the computation is either encoded in the quantum state of the unmeasured qubits (that is, the quantum output), or in the classical measurement outcomes [3, 34], with the former case giving a unitary evolution on the encoded information.

For example, consider a system of two qubits, A and B. Qubit A is prepared in a state $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$, whilst qubit B initially prepared in state $|+\rangle$. Performing a CZ gate between them entangles the input and results in the state $|\psi_{AB}\rangle = \alpha|0\rangle|+\rangle + \beta|1\rangle|-\rangle$. Note that by preparing A in state $|\phi\rangle$ instead of $|+\rangle$, we are able to encode information in the chain, so we call A the input to the chain, and since B is the system where the information will be at the end of the computation, we call this the output. We can rewrite the state $|\psi_{AB}\rangle$ as

$$\left|\psi_{AB}\right\rangle = \frac{1}{\sqrt{2}}\left|+_\theta\right\rangle\left(\alpha\left|+\right\rangle + e^{-i\theta}\beta\left|-\right\rangle\right) + \frac{1}{\sqrt{2}}\left|-_\theta\right\rangle\left(\alpha\left|+\right\rangle - e^{-i\theta}\beta\left|-\right\rangle\right)$$

$$= \frac{1}{\sqrt{2}}\left|+_\theta\right\rangle\tilde{H}U_z(\theta)|\phi\rangle + \frac{1}{\sqrt{2}}\left|-_\theta\right\rangle X\tilde{H}U_z(\theta)|\phi\rangle, \tag{3}$$

where $U_z(\theta)$ is a rotation about the z-axis by angle θ, and $\tilde{H}$ is a Hadamard gate. Now consider performing a $|\pm_\theta\rangle$ measurement on qubit A. If the outcome is $|+_\theta\rangle$, the state of qubit B is $\tilde{H}U_z(\theta)|\phi\rangle$, and if the outcome is $|-_\theta\rangle$ the state of qubit B is $X\tilde{H}U_z(\theta)|\phi\rangle$. If we apply a Pauli X correction on qubit B when the measurement outcome is $|-_\theta\rangle$, then both outcomes will be the same. In this way corrections allow for a deterministic implementation of the unitary operation $\tilde{H}U_z(\theta)|\phi\rangle$.

More general computation in MBQC can be depicted using graphs. Qubits prepared in the $|+\rangle$ state are represented by vertices V on a graph G, and the edges E represent which pairs of qubits have been acted on by a CZ gate. The state resulting from these operations is called a *graph state*, $|G\rangle$. The graph state of the cluster state, which is a universal resource for MBQC [3], is the two-dimensional (2D) square lattice (see figure 1). Graph states can also be defined using the stabilizer formalism [35], where it is defined as the state which satisfies the stabilizer eigenequations

$$K_v|G\rangle = |G\rangle \ \forall \ v \in V, \tag{4}$$

where a stabilizer generator is associated to each vertex v,

$$K_v = X_v \prod_{w \sim v} Z_w. \tag{5}$$

Here X_v, Y_v, Z_v are the Pauli matrices acting on site v, and the notation $v \sim w$ means that v and w are connected by an edge. The K_v generate the *stabilizer group* $S = \langle\{K_v\}_v\rangle$. The same group can be found by choosing different generators, for example $S = \langle\{K_\alpha K_v\}_v\rangle$ (for some arbitrary
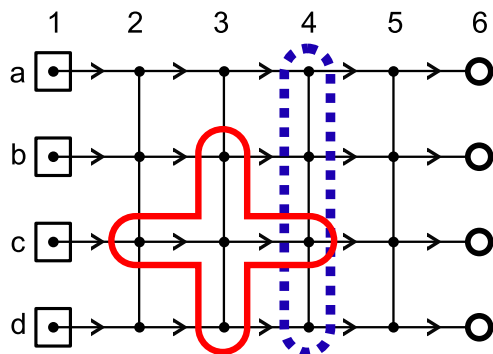
Figure 1. An illustration of the definitions in section 2.2, applied to a cluster state with depth 5. Inputs are represented by vertices with squares, and outputs are vertices with hollow circles. Arrows indicate *gflow* lines, the dotted circle indicates a single layer of qubits and the solid red cross contains all the vertices which contribute to $|g(v)|$ for vertex 2c (so $|g(2c)| = 5$).

fixed vertex α), and indeed the graph state is stabilized by the set of generators $\{K_\alpha K_v\}_v$ as well. This flexibility of choice will be useful later on.

As in the example above for a quantum input/quantum output computation, we can extend the definition to include input qubits, labelled I. In this case the stabilizer generators (5) are defined on non-inputs only. This is referred to as an *open* graph state. During a computation all vertices are measured except the output vertices, labelled O. In this work we are concerned with computations for quantum inputs and outputs, so we will be using open graph states from now on.

In order for a measurement pattern on a graph state to be able to be correctable such that the output is the same regardless of the measurement outcomes, it is sufficient (although not in general necessary) for the graph to have *gflow* [36, 37]. *Gflow* is an incredibly useful tool in MBQC that has been used to study parallelism [16, 18], the translation between MBQC and the circuit model [16, 38] and the emergence of causal order in MBQC [39]. *Gflow* allocates a time ordering over the vertices on a graph state and a *gflow function* $g(v)$ which tells us which vertices are affected by the measurement outcome of vertex v and which qubits can be used to correct for this. It is defined for measurements in any of the three planes, (X, Y), (X, Z) or (Y, Z), in a generalization of the example presented earlier. In this paper, we focus on measurements in the (X, Y) plane, as results for measurements in other planes will follow in a similar fashion, although, for completeness we present the definition of *gflow* for all planes.

The notation $v < w$ is used to represent that vertex v is measured before vertex w, and $v = w$ to indicate that v and w can be measured at the same time. We say that a set of vertices U is oddly (evenly) connected to a vertex v if there is an odd (even) number of edges connecting U and v. The definition of *gflow* is then:

Definition 1 Gflow. Given an open graph state G with inputs I, outputs O, edges E and vertices V, we say it has *gflow* if there exists a *gflow function* g and a time ordering $<$ over V such that, for all $v \in V$ which are not outputs:

(G1) All qubits w in $g(v)$ are in the future of v, i.e. $v < w$ for all $w \in g(v)$.

(G2) If $w \leqslant v$, and $v \neq w$, then w is evenly connected to all qubits in $g(v)$.

(G3) — (X, Y) plane: $v \notin g(v)$, and $g(v)$ is oddly connected to v.

— (X, Z) plane: $v \in g(v)$, and $g(v)$ is oddly connected to v.

— (Y, Z) plane: $v \in g(v)$, and $g(v)$ is evenly connected to v.

Given a *gflow* g, the associated correction required after measurement of vertex ν with result r_ν is

$$\left(\prod_{\mu \in g(\nu)} K_\mu \right)^{r_\nu}. \tag{6}$$

In [40] it is shown that if an open graph has *gflow* then it is possible to run an MBQC from input I to output O. Indeed if we ask that the corrections work for all measurement angles on the planes, the existence of a *gflow* is necessary and sufficient. Finding out if a graph has *gflow* can be done in polynomial time [41]. Furthermore, the *gflow* defines a valid measurement pattern, which may allow some qubits to be measured at the same time. Any qubits which can be measured simultaneously are said to be in the same layer of the computation. More formally;

Definition 2 Layers. A layer of a computation is defined as any (non-output) qubits in a measurement pattern which can be measured at the same time.

We denote the layers as L_k, and we use $L(v)$ to denote the layer that vertex v is in. For example, for the *gflow* defined on the graph in figure 1, the layers are given by $L_k = \{a_k, b_k, c_k, d_k, e_k\}$, for $k < 6$, and $L(a_k) = L(b_k) = L_k$ etc. We further denote $v \leqslant L_k$ for vertices in L_k or earlier layers, similarly for $v \geqslant L_k$. Using the concept of layers, we can define the depth for MBQC;

Definition 3 Depth. The depth of an MBQC with *gflow* is the number of rounds of measurements in the measurement pattern, or equivalently the number of layers in a measurement pattern.

In general this depth will be different depending on which *gflow* we are using (there can be more than one—indeed we will see in section 3.2 an example where many *gflow*s can be realized). Since we can think of *gflow* as a directed graph superimposed on an undirected graph, an equivalent and perhaps more intuitive definition is that the depth is the longest possible path along these directed edges.

The depth is effectively the time needed for the quantum part of the computation. To decrease this quantum time one is interested in pushing as many measurements together as possible to reduce the number of computational steps [16, 18]. This is achieved by what is called the *maximally delayed gflow* [41]. However, this is done at the expense of increasing the classical time needed to process the measurement results, which increases for larger *gflow*. The techniques of *gflow* thus give rise to a trade-off between the classical and quantum time for the computation [16, 18]. As we will see in the example in section 3.2 this trade-off can be great, so that for some cases all the time of the computation can be shifted to the classical processing except some constant quantum part. This trade-off is characterized in [18] in terms of circuits with fanout.

We end this subsection with some further definitions which will be important in the theorem later on, and are illustrated in figure 1.

Table 1. An illustration of the method to perform single qubit operations in adiabatic cluster-state quantum computation in 1D; arrows indicate an adiabatic transition from one operator to the other.

	T_1	T_2	T_3	...	T_{N-2}	T_{N-1}
	↓					
Step 1	X_1	T_2	T_3	...	T_{N-2}	T_{N-1}
		↓				
Step 2	X_1	X_2	T_3	...	T_{N-2}	T_{N-1}
			↓			
⋮				⋮		
					↓	
Step $N-2$	X_1	X_2	X_3	...	X_{N-2}	T_{N-1}
						↓
Step $N-1$	X_1	X_2	X_3	...	X_{N-2}	X_{N-1}

Definition 4 $|g(v)|$**.** The size $|g(v)|$ of a *gflow* is the number of qubits for which the product of the stabilizers over $g(v)$, $\prod_{v \in g(v)} K_v$, is non-trivial.

E.g. for the cluster-state, there is only one vertex in $g(v)$, and since each stabilizer generator acts on up to five qubits, the size of the *gflow* is 5.

Definition 5 Gflow lines. A *gflow line* is a directed edge from a vertex v to an element of $g(v)$. The set of *gflow* lines generates a directed graph over the set of vertices of the original graph (see for example figure 1).

2.3. Adiabatic cluster-state quantum computation

An adiabatically driven open-loop holonomic approach to MBQC was proposed in [6]; adiabatic cluster-state quantum computation. The approach is similar to that used in [31] for closed-loop HQC. First note that generating a cluster state is equivalent to initializing in the ground state of the Hamiltonian:

$$H_0 \equiv -\gamma \sum_{v \in V} K_v, \tag{7}$$

where γ parametrizes the strength of the interactions and K_v are cluster-state stabilizer generators. Indeed this is also true if we replace the $\{K_v\}$ by any set of generators of the stabilizer group. This will be useful later when extending to general graph states with *gflow*.

The computation protocol in [6] proceeds first by preparing the system in the ground state of H_0, which could be done by preparing the system in the ground state of a uniform magnetic field and adiabatically changing to H_0. Then each stabilizer generator is replaced by a rotated Pauli-X operator $X^{\theta_v} = | +_\theta \rangle\langle +_\theta | - | -_\theta \rangle\langle -_\theta | = \mathrm{e}^{-\mathrm{i}\theta_v Z_v/2} X_v \mathrm{e}^{\mathrm{i}\theta_v Z_v/2} = \mathrm{e}^{-\mathrm{i}\theta_v Z_v} X_v$ (see table 1), in analogy to the measurements in MBQC. This can be done one-by-one, or at the same time. If done one-by-one, the time for each individual replacement is independent of the computation, so that the total time scales with N. If all the replacements are done at the same

time it becomes difficult to get analytical bounds on the time, however numerical studies suggest that the energy gap scales inversely with the number of qubits [42].

This model allows for implementation of single qubit rotations and CNOT gates, and so is universal for quantum computation [43]. For the purposes of this paper, we will often use twisted stabilizer generators K_n^θ, defined as

$$K_n^{\theta_n} = X_n^{\theta_n} \prod_{n \sim m} Z_m. \tag{8}$$

We use these to form an initial Hamiltonian $H_0 = -\gamma \sum_{v \in \mathcal{V}} K_v^{\theta_n}$. Then instead of adiabatically replacing the stabilizer generators by $X_n^{\theta_n}$ operators, they are replaced by X_n operators, and the resulting computation is the same. The advantage of using the twisted picture is that certain results become clearer compared to the untwisted version. Note that, although this choice of stabilizer generators does not really affect the analogous MBQC protocol, in adiabatic cluster-state quantum computation these stabilizer generators are different physical Hamiltonians which must get realized.

We now look at the single qubit rotation to illustrate the main ideas and introduce a few concepts. Details of doing a controlled-NOT gate are in the appendix, which completes the set of universal gates [6]. To perform single qubit rotations in the adiabatic cluster-state model, consider a twisted one-dimensional (1D) cluster state. The stabilizer generators for such a state are

$$K_v^{\theta_v} = Z_{v-1} X_v^{\theta_v} Z_{v+1} \quad v = 2, \ldots, N-1; \quad K_N = Z_{N-1} X_N^{\theta_N}. \tag{9}$$

To fit with the notation used in [6], we will use slightly different operators $\{T_v\}$, where $T_v := K_{v+1}^{\theta_{v+1}}$. The initial Hamiltonian is then

$$H_0 \equiv -\gamma \sum_{v=1}^{N-1} T_v. \tag{10}$$

H_0 has a doubly degenerate ground state, which we can use to encode a qubit. For example, we can define logical states $|0\rangle_L$ and $|1\rangle_L$ as

$$|0\rangle_L := \prod_{v \sim w} \mathrm{CZ}_{(v,w)} |0\rangle_1 \bigotimes_{n>1} \left| +_{\theta_n} \right\rangle_n, \quad |1\rangle_L := \prod_{v \sim w} \mathrm{CZ}_{(v,w)} |1\rangle_1 \bigotimes_{n>1} \left| +_{\theta_n} \right\rangle_n \tag{11}$$

and similarly $| \pm \rangle_L := \frac{1}{\sqrt{2}} (|0\rangle_L \pm |1\rangle_L)$ and $| \pm i\rangle_L := \frac{1}{\sqrt{2}} (|0\rangle_L \pm i|1\rangle_L)$. Preparing the ground state in a superposition $\alpha|0\rangle_L + \beta|1\rangle_L$ corresponds to attaching an input $\alpha|0\rangle + \beta|1\rangle$ to the cluster state in the MBQC picture. The protocol proceeds by adiabatically replacing the T_1 with X_1 (see table 1), so that the time-dependent Hamiltonian is

$$H(s) = -\gamma(1-s)T_1 - \gamma s X_1 - \gamma \sum_{n=2}^{N-1} T_n, \tag{12}$$

where $0 \leqslant s \leqslant 1$. After this process, the information is encoded in the new degenerate ground space of $H(1)$, and the information originally encoded in $\{|0_L\rangle, |1_L\rangle\}$ has been transformed.

Rather than following how the information transforms by following how the ground state evolves, one can follow how the computation proceeds in terms of the logical operators X_L, Y_L, and Z_L, which for H_0 are

$$X_L \equiv X_1 Z_2, \qquad Y_L \equiv Y_1 Z_2, \qquad Z_L \equiv Z_1. \qquad (13)$$

Following the transformations of these operators is equivalent to following how the ground states transform, and can be simpler to deal with. In this picture, encoding an arbitrary input state is equivalent to applying the field $H_{\text{field}} = \alpha X_L + \beta Y_L + \gamma Z_L$.

To see how the information is transformed by the adiabatic substitution in (12), the logical operators are multiplied by stabilizers (since they act as identity) until they commute with X_1 (i.e., the logical operators are put in a form which is conserved during the adiabatic transformation). X_L already commutes with X_1, but Z_L doesn't so we multiply by T_1:

$$Z_L \rightarrow Z_1 T_1 = X_2^{\theta_2} Z_3. \qquad (14)$$

Then, after setting $X_1 \rightarrow \mathbb{1}$ since we are in the $+1$ eigenstate of X_1, the logical operators become

$$X_L \rightarrow Z_2, \qquad Z_L \rightarrow X_2^{\theta_2} Z_3. \qquad (15)$$

Now defining new logical operators in the same way as in equation (13):

$$X'_L \equiv X_2 Z_3, \qquad Z'_L \equiv Z_2. \qquad (16)$$

Expressing X_L and Z_L in terms of these new logical operators, we can see that the information has been moved one step along the chain and transformed by $U_2^{(L)} \tilde{H}^{(L)}$, where $\tilde{H}^{(L)}$ is a Hadamard operation acting in the logical subspace, and $U_v^{(L)} := \exp[-i\theta_v Z_L/2]$ is a logical Z rotation (cf the example in section 2.2). At the next step we replace T_2 with X_2, and so on from left to right down the chain (see table 1 for an illustration of this). Finally the information is encoded in the Nth qubit, with the information transformed by the operation $U_{\text{tot}}^{(L)}$, where

$$U_{\text{tot}}^{(L)} = \tilde{H}^{(L)} \prod_{v=N-2}^{2} \left(U_v^{(L)} \tilde{H}^{(L)} \right), \qquad (17)$$

and since $U_v^{(L)} = \exp[-i\theta_v Z_L/2]$, this will depend on the sequence of angles $\{\theta_v\}$ used (following [6] we set $\theta_1 = 0$ for convenience).

Since we are doing adiabatic transformations, the speed of the computation is limited by the ratio of the energy gap and $\|\dot{H}(s)\|$. For the Hamiltonian in (12), this is given by (see appendix B)

$$\tau \geqslant \tau_0 := \frac{c(\delta)}{\varepsilon 2^{1+\delta/2} \gamma}, \qquad (18)$$

where $0 < \delta \leqslant 1$, and ε is the error in the adiabatic evolution. For the remainder of this paper we will compare adiabatic evolution time to this time τ_0, so this is our definition of one unit of time for the adiabatic computation.

3. Adiabatic graph-state quantum computation

In the previous section we reviewed the results of [6, 42], where they show that universal quantum computation is possible using adiabatic substitutions, instead of measurements, on the cluster state. Here we generalize this method to other graph states using tools from MBQC and show that any MBQC measurement pattern on a graph state can be converted into an adiabatically driven adiabatic HQC of the form in [6], such that the same computation is

performed. We call this AGQC. We will first consider doing step-by-step transitions, then explore how the trade-off between quantum and classical time in MBQC manifests itself in AGQC.

3.1. Translation of MBQC patterns with gflow to adiabatic computation

In AGQC, given an open graph state with *gflow* $(g, <)$, and measurement angles $\{\theta_v\}$, we start in the ground state of the initial Hamiltonian

$$H_0 = -\gamma \sum_v T_v, \tag{19}$$

where the T_v are products of the twisted stabilizer generators,

$$T_v := \prod_{w \in g(v)} K_w^{\theta_w}, \tag{20}$$

$\forall \, v \notin O$. Hence the ground state corresponds to the twisted open graph state (an open graph state where the vertices are rotated according to the measurements). Preparing the system in the ground state of H_0 could be done by starting in the ground state of a magnetic field, and adiabatically evolving to H_0. This initial Hamiltonian can also be computed efficiently from the graph and *gflow*, since each T_v can be calculated in a time that scales as $O(\log(\max_v |g(v)|))$ using the methods in [44]. For simplicity we take the number of inputs to be equal to the number of outputs $|I| = |O|$, but all statements and proofs can be easily extended to the cases $|O| > |I|$ ($|O| < |I|$ is not allowed as this would mean information is lost). The final Hamiltonian will be $H_f = -\gamma \sum_v X_i$, and the transition will be done in steps, as in the cluster-state case. Indeed if the graph is a cluster (2D lattice) our model reduces to adiabatic cluster-state quantum computation (and $T_v = K_{g(v)}$, as $g(v)$ contains only one element).

To perform the computation, the T_v can be replaced one-by-one in an order that doesn't violate the *gflow*, or those in the same layer can be replaced all at once. Replacing them one-by-one will take N steps of time τ_0. When replacing layer by layer, the adiabatic transition for the kth step is governed by the interpolation Hamiltonian

$$H_{L_k}(s) = -\gamma \left(\sum_{v < L_k} X_v + \sum_{v > L_k} T_v \right) - \gamma \left(\sum_{v \in L_k} (1 - s) T_v + s X_v \right) \tag{21}$$

$$= (1 - s) H_{k-1} + s H_k, \tag{22}$$

where

$$H_k = \gamma \left(- \sum_{v \leqslant L_k} X_v - \sum_{v > L_k} T_v \right). \tag{23}$$

Note that $[X_u, T_v]_{u \neq v} = 0$ for any $u, v \in L_k$. Using this property and the analysis in appendix B, we see that the time taken to perform the adiabatic evolution of Hamiltonian $H_{L_k}(s)$ scales with $\Omega(|L_k|^{1+\delta})$, where $0 < \delta \leqslant 1$. Interestingly the dependence of the time on $|L_k|$ does not come from the energy gap Δ, which remains constant and independent of $|L_k|$, rather it comes from the norm $\|\dot{H}\|$, which scales as $|L_k|$. In this way we can replace all operators T_v in the same layer simultaneously, but the adiabatic runtime for each layer scales as the size of each

layer $|L_k|$. This point highlights a subtlety of the adiabatic theorem as applied here—the time is dominated by the norm $\|\dot{H}\|$, not the energy gap Δ as is more commonly the case.

Now consider what happens to the information when we replace all T_v operators defined above with an X_v operator, in the order given by *gflow*. This can be seen by multiplying logical operators with stabilizers in such a way that the logical operators commute with all the adiabatic 'measurements' [6] (as illustrated for the 1D chain graph in section 2.3). If there are any Z_v or Y_v operators which appear in a logical operator α, the *gflow* conditions guarantee that multiplying these terms by T_v will either give the identity or an X_v operator at vertex v, and the new logical operator will commute with any X_w for $w \leqslant v$. This means that it is possible to update the logical operators α_L, to $\tilde{\alpha}_L$ such that $[\tilde{\alpha}_L, X_v] = 0$ for all v which are not outputs, and where $\alpha = X, Y, Z$. The output of the computation is encoded in these final logical operators, after setting $X_v \rightarrow \mathbb{1}_v$ for all v which are non-outputs.

To see that this performs the same computation as in MBQC, consider performing MBQC on a twisted graph state. If we start with the logical operators X_L, Z_L of the MBQC resource state, and update these logical operators α_L, to $\tilde{\alpha}_L$ such that $[\tilde{\alpha}_L, X_v] = 0$ for all v, then all measurements in the X basis commute with these operators. Therefore if we start in the $+1$ eigenstate of the logical operators, after the measurements the final state will be the $+1$ eigenstate of $\tilde{\alpha}_L$ (after corrections have been applied). Indeed, the procedure outlined above for updating the logical operators is essentially the same process given by *gflow* for tracing the logical operators in MBQC in the Heisenberg picture as in [45], thus the computation is clearly identical.

This is summarized in the following theorem.

Theorem 1. *Any measurement based computation on an open graph state $|G\rangle$ of N qubits which is the ground state of a Hamiltonian H and which has gflow g and depth d can be efficiently converted into an adiabatically driven computation for which*

- *The adiabatic computation can be done in d steps, where the energy gap for each step is the same, and $\|\dot{H}\| = |L_j|$ for the jth step. Thus the time to perform the jth step is $\Omega(|L_j|^{1+\delta})$, where $0 < \delta \leqslant 1$.*
- *The maximum degree of the initial Hamiltonian, $k_{\max}$, is equal to the maximum gflow size : $k_{\max} = |g(v)|_{max}$.*
- *The initial Hamiltonian can be computed efficiently.*

For example, for adiabatic cluster-state quantum computation on a rectangular graph qubits with r rows and d columns, $g(v) = v + r$, $T_v = K_{v+r}$, $|g(v)| \leqslant 5$ and there are d layers, with each layer containing r qubits, so the time for each layer scales as $\Omega(|r|^{1+\delta})$.

In the above theorem we have performed the computation step-by-step mimicking the measurement pattern in MBQC. We could also replace all of the T_v operators at the same time regardless of which layers they are in, and the resulting computation would be the same, however the results in [42] suggests that the energy gap for such an evolution would shrink polynomially in the depth. Since it is desirable to keep the gap as large as possible to provide protection against errors, and since it is hard to find (analytically or numerically) the energy gaps of systems other than simple 1D chains when doing a one step transition, we do not follow this approach here.

Note that, for all known universal graphs there is a *gflow* for which $|g(v)|$ is bounded, so typically the degree will also be bounded. However this is not necessarily the case for all families of graphs; in some cases it can scale with the number of inputs (we will see an example of this in section 3.2). This raises an important question: Do we regard this increase in degree as a free resource, or is there a cost associated with it? The evidence to date would suggest that the latter is true, since typically in nature we only see two-body interactions, with higher degree interactions resulting as a low energy approximation. We take this approach in our model, and assume that degree is bounded and such high degree Hamiltonians must be simulated. There are known methods for constructing such large degree operators from two-local operators using *perturbation gadgets* [19, 46–48]. In particular we use the results from [19], that we can create k-local Hamiltonians H_k using a perturbative Hamiltonian acting on rk ancilla qubits and n computational qubits (r is the number of terms in Hamiltonian with degree k, which we consider as being fixed). The result is that the effective Hamiltonian, apart from some overall energy shift, is

$$\tilde{H}_{\text{eff}} = \frac{-k(-\lambda)^k}{(k-1)!} H_k \otimes P_+ + O\left(\lambda^{k+1}\right), \tag{24}$$

where P_+ is a projector on the space of r ancilla qubits, projecting each one into the $|+\rangle$ state, and the perturbation converges provided that $\lambda < \frac{k-1}{4k}$.

This energy gap decreases exponentially with k, therefore if we make the reasonable assumption that interactions in nature are limited to two (or a finite number) of bodies, then if the degree is allowed to scale with N, this imposes a prohibitive cost in that the minimum energy gap of the system shrinks exponentially, and so therefore the adiabatic time grows exponentially. This is not a general result, since (as far as the authors are aware) there is no general theorem saying that the gap must shrink when approximating k-body Hamiltonians, but it is a intuitive result that we would expect to be true.

3.2. Trade-offs in AGQC

In the previous subsection we saw how any MBQC computation with *gflow* can be mapped to a HQC. We now explore how the trade-off between quantum and classical time seen in MBQC [37] translates into this adiabatically driven model. First, as an illuminating example, consider the graph in figure 2 which was presented in [37] and gives rise to a trade-off between classical and quantum times in MBQC. Many different *gflows* can be defined on this graph, in particular a family of *gflows* can be defined as

$$g^r(v) = \begin{cases} \{N+v, \ldots, N+v+r-1\}, & \text{if } v+r-1 \leqslant N, \\ \{N+v, \ldots, 2N\}, & \text{if } v+r-1 > N, \end{cases} \tag{25}$$

where $1 \leqslant r \leqslant N$. For a *gflow* g^r, r measurements can be performed simultaneously, interspersed by classical processing. Corrections on qubits will be of the form $X^{s_1+s_2+\cdots s_m}$, where the s_m are binary variables accounting for the measurement outcomes. Thus the classical processing involves a binary sum of the results of these r measurement outcomes, and so takes time $O(\log r)$ [49]. There is also classical processing required on the outputs, which involves the same number of terms to be added and so can also be done in time $O(\log r)$ (each output requires addition of r binary variables or less, and these additions can be done in parallel). Since
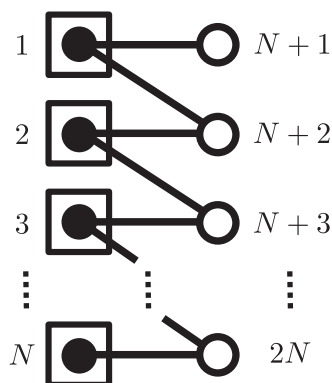
Figure 2. A graph for which many different *gflows* are applicable. Vertices represent qubits and edges indicate that two qubits are entangled. Vertices inside boxes are input qubits, whilst hollow circles are outputs (which are not measured).

we can perform r measurements simultaneously, the measurement depth d^r is given by $d^r = \left\lceil \frac{N}{r} \right\rceil$, and the size of the *gflow* is $|g^r(v)| \leqslant r + 2$.

The two extreme cases are where $r = 1$ or $r = N$. The former is just where each measurement is performed one-by-one ($d^1 = N$), with no addition of binary variables in between. The latter is where we can perform all measurements simultaneously ($d^N = 1$), but we must perform corrections on the outputs which take time $O(\log N)$. g^N is called the *maximally delayed flow* associated with this graph, whilst g^1 is the minimally delayed flow [41]. Since classical computation is typically a cheap resource, it is usually desirable to shift as much computation into classical processors as possible, and so in MBQC the optimal *gflow* to choose would be the maximally delayed *gflow*.

Following the conversion of these *gflows* into an adiabatically driven computation, we can perform the computation in d adiabatic steps, where the time to perform the jth layer scales as $\Omega(|L_j|^{1+\delta})$, and the maximum degree is given by the influencing volume. Thus g^r is converted into an adiabatic computation which has d^r steps, and each step takes $\Omega(r^{1+\delta})$ time, and the Hamiltonian degree $k = r + 2$. In the most extreme case, g^N is converted into an adiabatic computation which takes one step, but this step takes $\Omega(N^{1+\delta})$ time to complete, and the Hamiltonian degree is proportional to N. In all cases the total time for computation is $\Omega(N^{1+\delta})$.

In this way the trade-off in quantum time (d^r) versus classical time ($\log(r)$) that is facilitated by the use of *gflow* in MBQC, translates to a trade-off between the number of steps (d^r) and the degree of the initial Hamiltonian (r), as well as the norm $\| \dot{H} \|$ ($\leqslant r + 2$). However, even though the minimum energy gap is kept constant, the fact that the norm $\| \dot{H} \|$ scales with N means that there is no overall gain in time for the adiabatic computation. This result highlights the subtlety in the adiabatic theorem; although the minimum energy gap is the same for graphs of different flow, the size of $\| \dot{H} \|$ changes and leads to a dependence of the adiabatic time on the number of elements in a *gflow*. We also see that whereas in MBQC there is an advantage in using maximally delayed *gflow*, there is no such advantage for the AGQC case. Indeed, given that the maximally delayed *gflow* is accompanied by large (possibly unbounded) degree operators, and these Hamiltonians are likely to incur an exponentially decreasing energy gap, in AGQC it is better to use the minimally delayed *gflow* instead.

In MBQC it is only possible to perform measurements in one step for certain computations. A natural question to ask is whether or not the trade-off seen above in the application of theorem 1 to the zig-zag graph extends to more general computations, and can be extended to allow any computation to proceed in a constant time at the expense of high degree. If there was a way of implementing arbitrary computations instantly at the expense of increasing the degree only, then under reasonable assumptions this would imply an upper bound on the energy gap required for simulating such high degree operators. Such an upper bound would give an important fundamental limitation on the cost of large degree Hamiltonians, which does not currently exist.

To perform an AGQC on an arbitrary graph in one step, following the logic leading to theorem 1, we must find a transformation of the stabilizers $T_v \to \tilde{T}_v$ such that all of the $\tilde{T}_v$ stabilizers satisfy the commutation relations

$$\left[\tilde{T}_v, X_w \right] = 0 \ \text{ for all } \ v \neq w \text{ in } V \backslash O. \tag{26}$$

Then all of the stabilizers $\tilde{T}_v$ can be replaced at the same time, provided the adiabatic runtime scales as $\Omega(|N|^{1+\delta})$.

This transformation can be achieved by multiplying the $\tilde{T}_v$ stabilizers together. *Gflow* always allows such a procedure, since whenever a stabilizer T_w contains a Z_v or Y_v operator, these can be multiplied by T_v, giving identity or X_v, respectively, such that the resulting stabilizer commutes with X_v. Here we define a specific choice of such a procedure to make all stabilizers commute with all Pauli X operators. To update stabilizer T_v, we start in layer L_{n+1}, where $L_n = L(v)$, and proceed as follows; (1) if T_v contains a Z_w or Y_w term such that $w \in L_{n+1}$, multiply by T_w. (2) Proceed to the next layer, as determined by the time order. Iterate until the outputs are reached. (3) The final updated stabilizers are denoted $\tilde{T}_v$.

However, creating a Hamiltonian in this way is polynomially equivalent to simulating the computation by following the evolution of the logical operators (see e.g. [45]), which we have already used in our discussion of making general measurement patterns adiabatic. Or to put it another way, this procedure is only efficient for certain classes of operations. For Clifford operations the procedure takes polynomial time; if we have N qubits in total, we will have to perform one updating sweep per qubit, each sweep involves a search over at most N stabilizers to see whether they commute or anti-commute, and the cost of testing if a stabilizer commutes or not will be $O(N)$ since each stabilizer contains at most N terms, so the overall procedure takes $O(N^2)$ steps. For general angles the procedure takes an exponential amount of time, since at every step we replace a Z_v operator with a $Z_v T_v = \mathrm{e}^{-\mathrm{i}\theta_{v+1} Z_{v+1}} X_{v+1} Z_{v+2}$ term, so every update converts a Z term into three new terms after expanding the exponent. If we start off with a stabilizer containing n Z operators, then after r sweeps we will have $O(3^r n)$ operators to search through.

In addition, during this update procedure, we multiply every Z_w operator by $T_v = \prod_{w \in g(v)} K_w$. So following this procedure, each $\tilde{T}_v$ operator will have Pauli X's in positions $g(w)$ for every Z_w that has been have corrected for. The only parts that will contribute to the degree of $\tilde{T}_v$ are situated at vertices which can be arrived at by following a path on which a non-*gflow* line is preceded and followed by a *gflow* line. Following the discussion in section 3.1 we expect the simulation of these Hamiltonians to be very prohibitive.

As noted in section 3.1, an alternative way to perform all operations in one step is to just replace all stabilizers at once, without changing the form of the stabilizers, and we would expect

the energy gap to shrink polynomially in N for AGQC [42]. This alludes to another complementary trade-off between the energy gap and the number of steps in the computation.

In summary, we see that in AGQC it is always possible to perform the computation in one step, at the expense of increasing $\| \dot{H} \|$ and the degree of operators in the initial Hamiltonian. This means that there is no decrease in the overall computation time. In addition, the process of finding the initial Hamiltonian is closely linked to the classical simulation of the computation itself, and thus can also only be efficiently calculated for simple cases such as Clifford computations. We see that reducing the number of steps in the computation has no advantage in overall computation time. Furthermore, given the exponentially decreasing energy gap for the known methods of simulating large-degree operators, this suggests that the optimal way to perform AGQC is to keep the degree as low as possible, which in general means as many steps as possible. In the language of MBQC, this corresponds to using the minimally-delayed *gflow* for the computation.

4. Reordering the computation

In general, performing measurements in the wrong order in MBQC results in random outcomes to the computation. In some cases however the order of measurement does not matter, such as when all qubits are measured in the Pauli basis. A natural question to ask is whether or not this property also applies to AGQC. In this section we will go back to the original formulation of adiabatic cluster-state quantum computation, and look at what order the adiabatic substitutions can be performed in. Firstly we will consider the most obvious case, where the stabilizers are exactly the same as in [6], one stabilizer is replaced by one Pauli operator at each step. Then we consider a less constrained method, and discuss how these two approaches lead to different behaviour.

4.1. Re-ordering the adiabatic computation with fixed number of terms in Hamiltonian

We start with an N-qubit 1D chain with $N - 1$ stabilizer generators (i.e. one qubit is encoded in the chain), and we consider replacing one stabilizer generator by one Pauli operator in a different order to the corresponding measurement pattern in MBQC. As a physical motivation we may imagine that we have some experimental apparatus which is limited to only applying the T and X operators, and which can turn them on in any combination. We would like to find orders of replacements which keep the system in the 2D logical subspace, and we might expect from MBQC that changing the order will in some way disrupt the computation. As an example, consider a four-qubit chain with all angles set to zero (i.e. with untwisted stabilizer generators). The system is initialized in the ground state of the Hamiltonian

$$H_0 = -T_1 - T_2 - T_3. \tag{27}$$

Clearly it is possible to replace $T_2 \rightarrow X_2$ out of order, since $[T_1, X_2] = 0$. However, if T_3 is replaced with X_3 out-of-order, this leads to the time-dependent Hamiltonian

$$H(s) = -T_1 - T_2 - (1 - s)T_3 - sX_3, \quad \text{with } 0 \leqslant s \leqslant 1. \tag{28}$$

At $s = 1$, this Hamiltonian has a ground state degeneracy of 4, i.e. the degeneracy doubles. However, there is a constraint in that the operator $T_1 T_3$ commutes with $H(s)$ for all s and so the eigenstate of $T_1 T_3$ is conserved throughout the evolution. Therefore, since the system starts in the +1 eigenspace of $T_1 T_3$, it will also end in the +1 eigenspace of $T_1 T_3$. So although the gap closes,

the transitions between these degenerate eigenstates are forbidden and so the logical subspace is preserved.

Now consider replacing T_1 in (28) with X_1. Throughout this evolution, $T_1 T_3$ no longer commutes with $H(s)$, and so unless there is a stabilizer generator that can be multiplied with $T_1 T_3$ to make it commute with $H(s)$, the evolution is no longer constrained to the +1 eigenspace of $T_1 T_3$. It is easy to check that there are no remaining stabilizer generators which satisfy this property, since this requires a generator with a Z_1 term (and T_1 cannot be used since it doesn't commute with X_3 or X_1). Since at the very start of the $T_1 \rightarrow X_1$ evolution, the energy gap is zero and the subspace is no longer preserved, this means the information now leaks out of the subspace unless the evolution time $\tau \rightarrow \infty$. Thus the computation fails at the final step for this ordering. Note that this happens whether or not T_2 is replaced by X_2; the key part was the fact that there were no stabilizer generators left to multiply $T_1 T_3$ with to make it commute with $H(s)$.

Extending this to larger chains, whenever T_n is replaced with X_n such that $n > 2$, the evolution is still constrained to the +1 eigenspace of $T_{n-2} T_n$. If this 'hidden' stabilizer anticommutes with the next Pauli replacement X_m, then provided we can multiply by T_{m-2} the subspace is still preserved. But since at some point we need to replace $T_1 \rightarrow X_1$ or $T_2 \rightarrow X_2$, and there are no other stabilizer generators to multiply with to make sure that the 'hidden' stabilizer still commutes with $H(s)$, the computation fails as there will be zero energy gap and a non-zero matrix element to leak out of the 2D logical subspace.

Similar behaviour can be seen when the angles $\{\theta_v\}$ are all odd multiples of $\pi/2$ (corresponding to Y measurements). Again, using untwisted stabilizer generators, consider a Hamiltonian on three sites, $H = T_1 + T_2$. After replacing $T_2 \rightarrow Y_2$, the system is still in a 2D subspace since $[T_1 T_2, H(s)] = 0$ and so the subspace where $T_1 T_2$ has eigenvalue +1 is preserved. However, after replacing $T_1 \rightarrow Y_1$, there are no stabilizer generators which can be chosen, and so the computation fails at the last step. This argument can similarly be extended to larger chains, and just like in the above case we will find that once we reach the boundary the computation time has to go to infinity to avoid leakage.

Similar behaviour is also seen for the CNOT gate proposed in [6] and discussed in appendix A. Consider two rows of qubits, labelled a and b, with three columns numbered from left to right. The initial Hamiltonian is

$$
\begin{aligned}
H_{\mathrm{CNOT}} = &-T_{a1} - T_{a2} - T_{b1} - T_{b2} \\
&- Z_{a1} X_{a2} Z_{a3} Z_{b2} - Z_{a2} Z_{b1} X_{b2} Z_{b3} - Z_{a2} X_{a3} - Z_{b2} X_{b3}.
\end{aligned}
\tag{29}
$$

Although $T_{a2} \rightarrow X_{a2}$ can be replaced out of order without leaking out of the four-dimensional subspace (since the operator $T_{b1} T_{a2}$ is conserved), then when $T_{a1} \rightarrow X_{a1}$ is replaced afterwards, the computation fails. Thus we have seen that, with this approach, the only replacements in 1 dimension which can preserve the subspace are replacements with X_n with $n \leqslant 2$, and replacements by Y_n with $n \leqslant 1$, and no out-of-order measurements are possible for the CNOT gate.

What about more general angles? Since both X_n and Y_n measurements fail when $n > 2$, and any general measurement is a superposition of X and Y measurements, we might expect that any general measurement fails for $n > 2$. So we try replacements starting on the second site for general angles, i.e. replacements of the form

$$-\sum_{n=1}^{N-1} T_n \rightarrow -T_1 - X_2^{\theta_2} - \sum_{n=3}^{N-1} T_n. \tag{30}$$

The energy gap Δ_1 for such a process is given by

$$\Delta_1(\theta_2, s) = \sqrt{2\left(1 - s + s^2\right) + \Gamma(\theta_2, s)} - \sqrt{2\left(1 - s + s^2\right) - \Gamma(\theta_2, s)}, \tag{31}$$

where $\Gamma(\theta, s) \equiv \sqrt{2s^2 \cos 2\theta_2 + (4 - 8s + 6s^2)}$. For a given θ_2, this reaches a minimum at $s = (1 - \frac{1}{2}\cos\theta_2)$. Thus for all θ_2, the minimum energy gap $\Delta_1^{\min}$ is given by

$$\Delta_1^{\min}(\theta_2) = \Delta_1\left(\theta_2, 1 - \frac{1}{2}\cos\theta_2\right). \tag{32}$$

Over all possible θ_2 values, $\Delta_1^{\min}(\theta_2)$ is largest for $\theta_2 = l\pi$, $s = 1/2$, and goes to zero when $\theta_2 = (2l + 1)\pi/2$, $s = 1$, where l is an integer.

So we see that, for angles θ_2 such that the minimum energy gap remains non-zero, it is possible to perform the computation out-of-order. While the information remains in a protected subspace, it is also necessary to check if the information is transformed in the expected way. To see this, we start with the logical operators $X_L \equiv X_1 Z_2$, $Z_L \equiv Z_1$ (excluding Y_L since $Y_L = iZ_L X_L$). Using the method in [6], these operators are multiplied by stabilizers to make them commute with the 'measurements'. The appropriate transformation for the above case would be

$$X_L \rightarrow X_L T_2 = X_1 X_3^{\theta_3} Z_4,$$

$$Z_L \rightarrow Z_L T_1 = e^{-i\theta_2 Z_2} X_2 Z_3 \rightarrow e^{-i\theta_2 Z_2 T_2} X_2 Z_3 = e^{-i\theta_2 X_3^{\theta_3} Z_4} X_2 Z_3. \tag{33}$$

This is exactly the same transformation we would make in the normal computation, except two steps have been performed at once. Both of these logical operators are in a form which commute with X_1 and X_2, so they commute with the time-dependent Hamiltonian, and so, for example, if the system starts in the +1 eigenstate of X_L, it will end up in the +1 eigenstate of $X_3^{\theta_3} Z_4$.

In summary, we have seen that if we simply replace T_n stabilizer generators with X_n operators one-by-one on a 1D chain, we are extremely restricted in what we can do, and reordering is only possible using measurement angles which are not odd multiples of $\frac{\pi}{2}$. In particular, in a 1D chain, we can only start by replacing the stabilizer at the first site or the second site. The information stored in the chain is transformed in the same way in both cases, however the latter only works for $\theta_2 \neq (2n + 1)\frac{\pi}{2}$, and the energy gap depends on θ_2 so the speed of the adiabatic substitution must vary.

4.2. Re-ordering without a fixed number of terms in the Hamiltonian

We have seen that the initial approach to re-ordering the operations works only for a limited case. This is perhaps expected, as the way we have performed the out-of-order operations so far is not a proper reflection of what happens in MBQC. For instance, in MBQC measuring a qubit destroys any entanglement on edges connected to that qubit. This is clearly not true above, since we end up with Hamiltonians containing terms such as $T_1 + T_2 + X_3$. So a more natural way to perform the out-of-order measurements would be to remove all entanglement to measured qubits, or just remove all anticommuting terms entirely [50]. Take the chain considered above:

$$H_0 = -T_1 - T_2 - T_3. \tag{34}$$

Now if T_3 is replaced by X_3, any other operators which anticommute with X_3 are also removed, i.e. the final Hamiltonian is

$$H_1 = -T_2 - X_3. \tag{35}$$

Notice that the operator T_1T_3 is still conserved, as in the previous subsection. Instead of replacing T_1 with X_1, $-X_1$ can be added to the Hamiltonian, and the system is still constrained to be within a 2D subspace. This would also work if we had instead just removed all entanglement to site 3 (except T_2 can be left untouched since it commutes with X_3), i.e.

$$H_1 = -Z_1X_2 - T_2 - X_3. \tag{36}$$

The energy gap will still be the same when we introduce X_1 in equation (35) to when we replace $Z_1X_2 \rightarrow X_1$ in equation (36), since the Hamiltonian is of the form $(1-s)Z_1 \otimes A + sX_1$ and so has energy gap $2\eta = 2\sqrt{1 - 2s + 2s^2}$ (appendix B).

Similar results hold for Y measurements or the CNOT gate, so that Clifford operations can be performed in any order. Note that if more than one operator is replaced at the same time, the number of anticommuting terms increases, and based on the numerical studies in [42], if the number of anticommuting terms scales with n we would expect the energy gap to be polynomial in $1/n$.

In summary, we have seen that Clifford operations can be done in any order, provided that either the stabilizer generators are modifed so that the entanglement to 'measured' sites is destroyed, or any stabilizer generators which anticommute with the measurement operator are removed entirely. This is in contrast to the method in the previous subsection, in which the stabilizer generators did not reflect what happens in MBQC, and in which the re-ordering is limited to performing alternating measurements from left to right. In both of the approaches considered, Clifford operations cannot be performed all in one step, since we would expect the energy gap to decrease polynomially in the number of stabilizer generators we replace.

5. Conclusions

We have shown that any measurement pattern with *gflow* can be converted into an adiabatically driven HQC, such that the number of adiabatic steps is equal to the depth in MBQC, and each step takes time proportional to the size of each layer. This opens up the possibility of future results about efficiency and trade-offs in MBQC being used in AGQC. For example, there is still little understood about how to view the efficient simulatability of Clifford gates from the perspective of *gflow*. The framework developed here offers a natural route to translate future possible results in this area, which may have implications for the efficiency of AGQC as well as the simulation of many-body Hamiltonians. In addition, since fault-tolerant schemes for HQC on stabilizer codes exist [31], we would expect it to be possible to extend this to AGQC. Beyond computation itself, the inherent interplay between classical and quantum processing in MBQC has led to the cryptographic protocols of blind quantum computation [16] and verified universal computation [17]. Our translation captures much of this interplay by using *gflow* as a main tool, so one may hope that it can help translate these protocols and ideas across to HQC.

We have also found that, in analogy to the trade-off between quantum and classical time in MBQC, there is a trade-off between the number of adiabatic steps taken and the size of $\|\dot{H}\|$, together with the degree of the initial Hamiltonian. One interesting point is that the trade-off is

only between the number of steps and $\| \dot{H} \|$, but does not involve the energy gap, which highlights the subtleties of using the adiabatic theorem. It is perhaps surprising that, even if we could simulate high-degree operators without an exponentially shrinking energy gap, the overall adiabatic time would still scale with N. We might have expected that the free availability of large degree operators would allow some computational speed up, but the fact that this isn't the case suggests that there may not be any reason *in principle* why it should be prohibitive to simulate large degree operators. It is also interesting that the increase in degree does not appear to provide any computational advantage in this model. Since known methods of simulating high-degree operators result in an exponentially small energy gap in AGQC it is optimal to use the minimally delayed flow, in contrast to MBQC where the maximally delayed flow is preferable.

We considered the influence of adiabatic measurements in a different order from the corresponding MBQC pattern. When stabilizers are replaced in a different order, the computation fails with the exception of a few, special cases. When however the operations are performed in such a way that the terms that anticommute with the measurement are all adiabatically removed, then all Clifford operations can be performed in any order.

Finally we stress that these results do not cover all possible methods of performing MBQC, as there are some graph states without *gflow* which still yield deterministic computations [37]. For theorem 1 we require correcting sets to be known which allow us choose stabilizers $\{T_v\}$ such that $\{T_v, X_v\} = 0$, and $[T_v, X_w] = 0$ for all $v \neq w$. For extensions of *gflow* where such correcting sets are known our arguments can be carried forward simply, for example for Pauli flow [37]. However, more generally, the lack of characterization of possible correcting sets means it is not easy to guarantee these conditions are met so our procedure may not work. There are also more resource states for MBQC such as those investigated in [33], which could have interesting properties.

Appendix A. Adiabatic CNOT gate

To perform universal quantum computations an entangling gate on the encoded information is needed, such as a controlled-NOT gate. Consider two rows of qubits, labelled a and b, with three columns numbered from left to right (see figure A1). The protocol starts with an initial Hamiltonian in which all angles $\{\theta_n\}$ are 0, and which the inputs are on qubits a_1 and b_1, and the outputs are on qubits a_3 and b_3:

$$H_{\text{CNOT}} = -Z_{a1}X_{a2}Z_{a3}Z_{b2} - Z_{a2}Z_{b1}X_{b2}Z_{b3} - Z_{a2}X_{a3} - Z_{b2}X_{b3}. \qquad (A.1)$$

Adiabatically replacing all of these stabilizers with X operators results in a CNOT gate acting on the encoded information [6]. Each replacement of a single stabilizer by a local Pauli operator
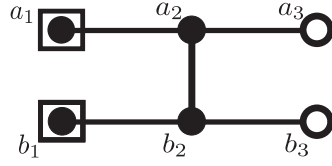
Figure A1. An illustration of the graph used to perform a CNOT gate in adiabatic cluster-state quantum computation.

still has the same adiabatic time as in the previous section, i.e., the adiabatic evolution time takes the form τ_0, provided these substitutions are done progressing from left to right. A controlled-Z rotation can also easily be achieved using an adiabatic scheme based on the gates in [37], in which only three qubits are required to perform a gate on two qubits.

Appendix B

In this section we will prove the following lemma.

Lemma 6. *Consider two Hamiltonians acting on a graph with vertices V and gflow* $(g, <)$;

$$H = -\gamma \left(\sum_{u<U} X_u + \sum_{v \geqslant U} T_v \right), \qquad H' = -\gamma \left(\sum_{u \leqslant U} X_u + \sum_{v > U} T_v \right), \qquad (B.1)$$

where $T_v := \prod_{w \in g(v)} K_w^{\theta_w}$. *At time* $s = 0$ *we prepare a state in the ground subspace of H, and adiabatically change the Hamiltonian to H′ using an interpolation of the form* $H(s) = (1 - s)H + sH'$. *Then provided* $[T_u, X_v]_{v \neq u} = 0$ *for all* $u, v \in U$, *it is possible to finish in the ground subspace of H′ with high probability provided that the adiabatic time* τ *can scale as* $\tau = \Omega(|U|^{1+\delta})$, *where* $0 \leqslant \delta \leqslant 1$ *[32]*.

To prove this, the spectrum of $H(s)$ and the spectrum of $\dot{H}(s)$ must be derived and inserted into the expression for the adiabatic time in equation (2). The time dependent Hamiltonian in this case is

$$H(s) = -\gamma \sum_{u \in U} \left[(1 - s)T_u + sX_u \right] - \gamma \sum_{u<U} X_u - \gamma \sum_{u>U} T_u. \qquad (B.2)$$

First note that X_v commutes with all stabilizers T_w with $w \geqslant v$, $w \neq v$, and anticommutes with T_v. This follows from the conditions of *gflow*, since a product of stabilizer generators $\prod_{y \in g(w)} K_y$ has an even number of connections to vertices $v \leqslant w$ (which means an even number of Z operators, which cancel to give identity), whilst T_v has an an odd number of connections to vertex v (which means there is one Z_v term T_v). This is true whether or not we are considering twisted stabilizers. Therefore we have $\{T_v, X_v\} = 0$, and $[T_w, X_v] = 0$ for all $w \geqslant v$, which means that the terms in the second two sums commute with each other and with the first sum. Then since $[T_u, X_v]_{v \neq u} = 0$ for all $u, v \in U$, each of the summands in the left hand sum commute with each other. We therefore know that the eigenvalues of $H(s)$ will be formed from combinations of the eigenvalues of the $\gamma[(1 - s)T_u + sX_u]$ terms, plus multiples of ± 1 from the remaining terms. To be exact, since the terms in $H(s)$ are commuting normal operators with non-overlapping spectral projections, following the analysis in [51] the eigenvalues of $H(s)$ will

be all possible combinations of the eigenvalues of the individual commuting terms. The eigenvalues of the individual $[(1 - s)T_u + sX_u]$ terms are $\pm\sqrt{(1 - s)^2 + s^2} := \pm\, \eta$, so the eigenvalues of $H(s)$ are $-|U|\gamma\eta, -(|U| - 2)\gamma\eta, \ldots, (|U| - 2)\gamma\eta, |U|\gamma\eta$ plus some integer multiples of ± 1 from the remaining X_n, T_n terms. The energy gap between the ground state and first excited state is therefore 2η. This is minimal at $s = \frac{1}{2}$, at which point the gap is $\sqrt{2}\,\gamma$.

The time-derivative of the Hamiltonian is

$$\dot{H}(s) = \gamma \sum_{u \in U} \left(T_u - X_u\right). \tag{B.3}$$

Using the same reasoning as above, the eigenvalues of this derivative are $\{-|U|\gamma, -(|U| - 2)\gamma, \ldots, (|U| - 2)\gamma, |U|\gamma\}$, so the norm of $\dot{H}(s)$ is just $|U|\gamma$. Inserting the minimal energy gap and $\|\dot{H}(s)\|$ into equation (2), we then find that a system prepared in the ground state of H which is adiabatically changed to H' using a linear interpolating function will be ε-close in the l_2 norm to the ground subspace of H' provided that the adiabatic time τ scales as

$$\tau \geqslant \left(\frac{c(\delta)\,|U|^{1+\delta}}{\varepsilon\, 2^{1+\delta/2}\gamma}\right) = \Omega\left(|U|^{1+\delta}\right). \tag{B.4}$$

References

[1] Shor P W 1997 *SIAM J. Comput.* **26** 1484–509
[2] Nielsen M A and Chuang I L 2004 *Quantum Computation and Quantum Information* (Cambridge: Cambridge University Press) ISBN 0521635039
[3] Raussendorf R and Briegel H J 2001 *Phys. Rev. Lett.* **86** 5188–91
[4] Farhi E, Goldstone J, Gutmann S and Sipser M 2000 arXiv:quant-ph/0001106
[5] Aharonov D and Ta-Shma A 2007 *SIAM J. Comput.* **37** 47–82
[6] Bacon D and Flammia S T 2010 *Phys. Rev.* A **82** 30303
[7] Zanardi P and Rasetti M 1999 *Phys. Lett.* A **264** 94–99
[8] Pachos J, Zanardi P and Rasetti M 1999 *Phys. Rev.* A **61** 010305
[9] Kult D, Åberg J and Sjöqvist E 2006 *Phys. Rev.* A **74** 022106
[10] Childs A M and Wiebe N 2012 *Quantum Inf. Comput.* **12** 901–24
[11] Kieferova M and Wiebe N 2014 arXiv:1403.6545
[12] Hen I 2014 arXiv:1401.5172
[13] Cesare C, Landahl A J, Bacon D, Flammia S T and Neels A 2014 arXiv:1406.2690
[14] Williamson D J and Bartlett S D 2014 arXiv:1408.3415
[15] Brell C G 2014 arXiv:1408.6237
[16] Broadbent A, Fitzsimons J and Kashefi E 2009 *50th Annual IEEE Symp. on Foundations of Computer Science FOCS '09* pp 517–26
[17] Fitzsimons J F and Kashefi E 2012 arXiv:1203.5217
[18] Browne D, Kashefi E and Perdrix S 2011 Computational depth complexity of measurement-based quantum computation *Theory of Quantum Computation, Communication and Cryptography (Lecture Notes in Computer Science)* (Berlin: Springer) pp 35–46
[19] Jordan S P and Farhi E 2008 *Phys. Rev.* A **77** 62329
[20] Bravyi S, DiVincenzo D P, Loss D and Terhal B M 2008 *Phys. Rev. Lett.* **101** 070503
[21] Born M and Fock V 1928 *Z. Phys.* **51** 165

[22] Berry M V 1984 *Proc. R. Soc.* A **392** 45–57
[23] Pancharatnam S 1956 *Proc. Indian Acad. Sci.* A **44** 247–62
[24] Xiao D, Chang M C and Niu Q 2010 *Rev. Mod. Phys.* **82** 1959–2007
[25] Wilczek F and Zee A 1984 *Phys. Rev. Lett.* **52** 2111–4
[26] Aharonov Y and Anandan J 1987 *Phys. Rev. Lett.* **58** 1593–6
[27] Farhi E, Goldstone J, Gutmann S, Lapan J, Lundgren A and Preda D 2001 *Science* **292** 472–5
[28] Barahona F 1982 *J. Phys. A: Math. Gen.* **15** 3241–53
[29] Young K C, Sarovar M and Blume-Kohout R 2013 *Phys. Rev. X* **3** 041013
[30] Carollo A, Fuentes-Guridi I, Santos M F and Vedral V 2003 *Phys. Rev. Lett.* **90** 160402
[31] Oreshkov O, Brun T and Lidar D 2009 *Phys. Rev. A* **80** 022325
[32] Reichardt B W 2004 *Proc. 36th Annual ACM Symp. on Theory of Computing STOC '04* (New York: ACM) pp 502–10
[33] Gross D and Eisert J 2007 *Phys. Rev. Lett.* **98** 220503
[34] Anders J and Browne D 2009 *Phys. Rev. Lett.* **102** 050502
[35] Hein M, Dür W, Eisert J, Raussendorf R, den Nest M V and Briegel H J 2005 *Quantum Computers, Algorithms and Chaos* vol 162 (Amsterdam: IOS Press) pp 115–218
[36] Danos V and Kashefi E 2006 *Phys. Rev. A* **74** 052310
[37] Browne D, Kashefi E, Mhalla M and Perdrix S 2007 *New J. Phys.* **9** 250
[38] da Silva R D, Pius E and Kashefi E 2013 arXiv:1301.0351
[39] Raussendorf R, Sarvepalli P, Wei T and Haghnegahdar P 2011 arXiv:1108.5774
[40] Browne D E and Briegel H J 2006 arXiv:quant-ph/0603226
[41] Mhalla M and Perdrix S 2008 *Automata, Languages and Programming* (*Lecture Notes in Computer Science* vol 5125) ed L Aceto, I Damgård, L Goldberg, M Halldórsson, A Ingólfsdóttir and I Walukiewicz (Berlin: Springer) pp 857–68
[42] Bacon D, Flammia S T and Crosswhite G M 2013 *Phys. Rev. X* **3** 021015
[43] Barenco A, Bennett C H, Cleve R, DiVincenzo D P, Margolus N, Shor P, Sleator T, Smolin J A and Weinfurter H 1995 *Phys. Rev. A* **52** 3457–67
[44] Aaronson S and Gottesman D 2004 *Phys. Rev. A* **70** 052328
[45] Markham D and Kashefi E 2013 arXiv:1311.3610
[46] Kempe J, Kitaev A and Regev O 2004 *Proc. 24th FSTTCS* vol 35 pp 372–83
[47] Oliveira R and Terhal B M 2008 *Quantum Inf. Comput.* **8** 900–24
[48] Bartlett S D and Rudolph T 2006 *Phys. Rev. A* **74** 40302
[49] Furst M, Saxe J and Sipser M 1984 *Math. Syst. Theory* **17** 13–27
[50] Fitzsimons J 2013 private communication
[51] Nielsen M A 2005 The fermionic canonical commutation relations and the Jordan–Wigner transform (http://michaelnielsen.org/blog/?p=233)