Introduction to Quantum Computing
Lecture 13: The Stabiliser Formalism

Petros Wallden

School of Informatics, University of Edinburgh

2nd November 2017
Stabiliser Codes

- Important class of QECC – analogous to classical linear codes
- Need to introduce the stabiliser formalism
- Powerful method for understanding a wide class of operations. Leads to theorem about the limitations of these operations.
- Given state $|\psi\rangle$, there are unitary operators $A$ that leave it invariant (stabilise it) – $+1$ eigenvectors; $A|\psi\rangle = |\psi\rangle$
- Exists a collection of operators that uniquely determine the state. Example: $|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$

The operators \{X_1 X_2, Z_1 Z_2\} stabilise the state, and uniquely (up to global phase) determine it

$X_1 X_2 |\psi\rangle = Z_1 Z_2 |\psi\rangle = |\psi\rangle$

- Many quantum states can be easier described with the stabiliser operators than the state itself
- Many QECC (Shor) can be described simpler with stabilisers
- Errors on qubits and operations: Hadamard, phase, and control-NOT gate and measurements in the computational basis, easier described with the stabiliser formalism
- Group is a collection of elements $G = \{ e_1, \cdots, e_n \}$ (in general $n$ can be infinite), along with an operation $\cdot$ called group law (in our case it is matrix multiplication) with the following properties

(i) **Closure**: $e_i \cdot e_j = e_k$ for some $k$

(ii) **Associativity**: $e_i \cdot (e_j \cdot e_k) = (e_i \cdot e_j) \cdot e_k$

(iii) **Identity element**: Exist element $e_0$ such that $e_0 \cdot e_i = e_i \cdot e_0 = e_i$ for all $e_i$

(iv) **Inverse**: For each element $e_i$ there exists element $e_i^{-1}$ such that $e_i e^{-1}_i = e_i^{-1} e_i = e_0$

- The **Pauli Group**

  for one qubit has as elements the following matrices and as a group operation the matrix multiplication $G_1 := \{ \pm 1, \pm i, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ \}$

  (Q: Check that it is a group)

- The general $n$-qubit Pauli group $G_n$ consists of all $n$-fold tensor products of Pauli matrices, again with the factors $\pm 1, \pm i$

  (e.g. elements of the 2-qubit Pauli group are $X_1 \otimes X_2, iX_1 \otimes Z_2, -1 \otimes Y_2$ etc)
- Consider subgroup $S$ of $G_n$. Is a subset of $G_n$ such that it still forms a group (closed and contains the identity)
- Let $V_S$ be the set of $n$-qubit states which are fixed (stabilised) by every element of $S$
We call:
1. $V_S$ the vector space stabilised by $S$
2. $S$ the stabiliser of the space $V_S$
- One can see that $V_S$ is the intersection of all the $n$-qubit states fixed by each operation in $S$
Example: The 3-qubits case. Let $S = \{1, Z_1Z_2, Z_2Z_3, Z_1Z_3\}$
The space that is stabilised by this stabiliser is $V_S = \text{span}(\{|000\rangle, |111\rangle\})$
(i.e. the space spanned by those two vectors. Check!)
- Represent “economically” a group in terms of its generators

Generator: A set of elements \( \text{Gen} = \{g_1, \cdots, g_m\} \) of a group \( G \) is called generator if every element in \( G \) can be expressed as product of elements \( \text{Gen} \). We then denote the group \( G = \langle g_1, \cdots, g_l \rangle \) (more compact way representing a group)

A vector is stabilised by a group \( G \) if and only if it is stabilised by all the generators

Example: In the example given before for the group

\( S = \{1, Z_1Z_2, Z_2Z_3, Z_1, Z_3\} \) the generator of this group is

\( S = \langle Z_1Z_2, Z_2Z_3 \rangle \)

to see this note: \( (Z_1Z_2)(Z_1Z_2) = 1 \) and \( (Z_1Z_2)(Z_2Z_3) = Z_1Z_3 \)

We can see that the (basis) states \( |000\rangle, |111\rangle \) are stabilised by the elements of the generator
- For a subgroup $S$ of the $G_n$ n-qubit Pauli group to stabilise an non-trivial vector space $V_S$ two conditions must hold:

1. $-1 \notin S$
2. All elements of $S$ should commute

(check that this is the case for the examples given!)
- Can use the stabiliser formalism to describe the (unitary) evolution of vectors.

- Apply $U$ to a state that is stabilised by the group $S$.

Let $|\psi\rangle \in V_S$ and $g \in S$.

$$U |\psi\rangle = U g |\psi\rangle = Ug U^\dagger (U |\psi\rangle)$$

We can see that the unitary evolved state $U |\psi\rangle$ is stabilised by elements $Ug U^\dagger$.

The vector space $UV_S$ is stabilised by the group $USU^\dagger = \{Ug U^\dagger | g \in S\}$.

i.e. to see the evolution, it suffices to see how the generators are affected.
Let's see what is the effect of known unitary operators on Pauli matrices. For example the Hadamard $H$:

$$HXH^\dagger = Z; \quad HYH^\dagger = -Y; \quad HZH^\dagger = X$$

The following gates map Pauli to Pauli:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>controlled-NOT</td>
<td>$X_1$</td>
<td>$X_1X_2$</td>
</tr>
<tr>
<td></td>
<td>$X_2$</td>
<td>$X_2$</td>
</tr>
<tr>
<td></td>
<td>$Z_1$</td>
<td>$Z_1$</td>
</tr>
<tr>
<td></td>
<td>$Z_2$</td>
<td>$Z_1Z_2$</td>
</tr>
<tr>
<td>$H$</td>
<td>$X$</td>
<td>$Z$</td>
</tr>
<tr>
<td></td>
<td>$Z$</td>
<td>$X$</td>
</tr>
<tr>
<td>$S$</td>
<td>$X$</td>
<td>$Y$</td>
</tr>
<tr>
<td></td>
<td>$Z$</td>
<td>$Z$</td>
</tr>
<tr>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
</tr>
<tr>
<td></td>
<td>$Z$</td>
<td>$-Z$</td>
</tr>
<tr>
<td>$Y$</td>
<td>$X$</td>
<td>$-X$</td>
</tr>
<tr>
<td></td>
<td>$Z$</td>
<td>$-Z$</td>
</tr>
<tr>
<td>$Z$</td>
<td>$X$</td>
<td>$-X$</td>
</tr>
<tr>
<td></td>
<td>$Z$</td>
<td>$Z$</td>
</tr>
</tbody>
</table>

- These are the **Clifford** gates
- Circuits that involve gates from the above set, acting on states stabilised by subsets of $G_n$ can be described very efficiently. Normally one needs $2^n$ numbers to describe the amplitudes of the evolved $n$-qubit state. Stating how the generator changes can be described using numbers linear in $n$. e.g. the state with stabiliser $\langle Z_1, Z_2, \cdots, Z_n \rangle$ if acted with Hadamard gates $H_1 \otimes \cdots \otimes H_n$ it is mapped to the state $\langle X_1, X_2, \cdots, X_n \rangle$

- Can prove that any unitary operation mapping elements of $G_n$ to elements of $G_n$ can be composed from Hadamard $H$, phase gate $S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$ and controlled-NOT.

**Definition:** The set $U$ such that $UG_n U^\dagger = G_n$ is called the **normaliser** of $G_n$ and denoted $N(G_n)$
Gottesman-Knill Theorem

Suppose a quantum computation involving only:
(i) state preparations in the computational basis, (ii) Hadamard gates, (iii) phase gates, (iv) controlled-NOT gates, (v) Pauli gates, and (vi) measurements of observables in the Pauli group; together with (vii) the possibility of classical control conditioned on the outcome of such measurements.
Then it can be **efficiently simulated on a classical computer**.

- Not all quantum operations are of this type. Example that is not of this type is the gate $R_{\pi/4} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
- Quantum circuits that involve gates that are not Clifford, cannot be easily described within the stabiliser formalism
- Many important examples involve Clifford gates. Encoding, decoding, error-detection and recovery for stabiliser QECC can be done only with the normaliser gates.
- First we identify the space spanned by the codewords (logical qubits) \( V_S \) and describe it using its stabiliser \( S \subset G_n \)
- Then one can prove when a set of errors is correctable:

  Let \( \{ E_i \} \) be the set of possible errors. This set of errors is correctable if for all pairwise products \( E_i E_j \) it is the case that either \( E_i E_j \) belongs to \( S \) or anti-commute with at least one element \( S_i \) of \( S \)

- If it anti-commutes, then measuring \( S_i \) distinguishes perfectly between the two errors \( E_i, E_j \): 
  
  \[ \langle \psi |_L E_i \times E_j |\psi \rangle_L = 0 \]

- If \( E_i E_j \) belongs in \( S \), then the effect of \( E_i \) and \( E_j \) on code-words is identical (degenerate code) and the same action corrects both errors:
  
  \[ \langle \psi |_L E_i \times E_j |\psi \rangle_L = 1 \]
- To detect the errors we measure the generators of the stabiliser and we get a $+1$ or $-1$ outcome. For each $-1$ outcome, we find one error that has this syndrome and apply the suitable correction (see example and tutorial)

Example: Shor’s code in stabiliser form

The generators are given by the following table

<table>
<thead>
<tr>
<th>Name</th>
<th>Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g_1$</td>
<td>$ZZI I I I I I I$</td>
</tr>
<tr>
<td>$g_2$</td>
<td>$IZZ I I I I I$</td>
</tr>
<tr>
<td>$g_3$</td>
<td>$I I I ZZ I I I I$</td>
</tr>
<tr>
<td>$g_4$</td>
<td>$I I I I ZZ I I I$</td>
</tr>
<tr>
<td>$g_5$</td>
<td>$I I I I I I Z Z I$</td>
</tr>
<tr>
<td>$g_6$</td>
<td>$I I I I I I I Z Z$</td>
</tr>
<tr>
<td>$g_7$</td>
<td>$XXXXXX I I I I$</td>
</tr>
<tr>
<td>$g_8$</td>
<td>$I I I I X X X X X$</td>
</tr>
</tbody>
</table>
- Can check that both codewords (logical bits) are stabilised by all the generators

\[
|0\rangle_L \rightarrow \frac{(|000\rangle+|111\rangle)(|000\rangle+|111\rangle)(|000\rangle+|111\rangle)}{2\sqrt{2}}
\]

\[
|1\rangle_L \rightarrow \frac{(|000\rangle-|111\rangle)(|000\rangle-|111\rangle)(|000\rangle-|111\rangle)}{2\sqrt{2}}
\]

- Can check that the set of all single-qubit errors is correctable. All pairs of single qubit errors anti-commute with at least one generator (or belong to the stabiliser)

Examples:

1. \(X_1\) and \(Y_4\) anticommutes with \(g_1\).
   \(\{X_1Y_4, Z_1Z_2\} = 0\)

2. \(Z_1\) and \(X_4\) anticommutes with \(g_7\).
   \(\{Z_1X_4, X_1X_4\} = 0\)

3. \(Z_1\) and \(Z_2\) belongs to the stabiliser (is actually \(g_1\)).
- The error-detection is measuring the generators and getting values (syndromes) $\pm 1$
eq 1\) e.g. If there is a flip at the first qubit the stabiliser $g_1$ will give outcome $-1$ while all the others will be unaffected. This means that the error was a bit flip and can be corrected with applying a Pauli $X$ at the first qubit. (the error will change the stabiliser to the $\langle -g_1, g_2, \cdots, g_8 \rangle$ and by applying the correction $X_1$ we recover the stabiliser $\langle g_1, g_2, \cdots, g_8 \rangle$)

- For a given error $E_i$ to determine what syndrome it would give we do the following $E_i g_j E_i^\dagger = \beta_j g_j$, where $\beta_j \in \{+1, -1\}$. The syndrome of the error $E_i$ is $(\beta_1, \beta_2, \cdots, \beta_8)$

**Summary**: Measure the syndromes and obtain a string of $\{\pm 1\}$. Then check which error $E_i$ gives this string $(\beta_1, \beta_2, \cdots, \beta_8)$. Then apply the suitable correction.