Note on Random Clifford Circuit Simulation with Measurement

We wish to simulate a mixed state evolution of a code state evolving under a random Clifford circuit with random, local measurements. We are only interested in the entanglement growth of the circuit, and not the actual measurement results themselves, so we can ignore the phase of the stabilizers, and only keep track of their one-hot representation. This provides a group isomorphism between the quotient group $\mathcal{P}(N)/\mathbb{Z}_4$, which is abelian, and $2N$ dimensional vectors over $\mathbb{Z}_2$, with multiplication and addition defined as usual, modulo two.

The current numerics for simulating the evolution of the code state keeps track of the stabilizers in a matrix S with $r = |\mathcal{G}|$ columns, where r is the number of stabilizers, and uses Gaussian elimination when simulating measurements, which in general takes $\mathcal{O}(N^3)$. Here, we wish to provide an algorithm[1] which keeps track of a larger matrix S with $2N$ columns, but performs measurements in $\mathcal{O}(N^2)$. This matrix has full rank, its columns form a representation (isomorphism) of the quotient group. The new matrix is

$$S = \{|g_1\rangle, ... |g_r\rangle, |x_1\rangle ... |x_{N-r}\rangle, |h_1\rangle, ... |h_r\rangle, |z_1\rangle ... |z_{N-r}\rangle\} \tag{1}$$

The $\{|g_i\rangle\}$ are the stabilizers and define the code state ρ_S. The $\{|h_i\rangle\}$ are the de-stabilizers, which anti-commute with their corresponding stabilizer and commute with the rest. The other columns are the remaining x and z bits which haven't yet been incorporated into the (de)stabilizer set. By keeping track of these additional bits, we can perform measurements quickly without resorting to Gaussian elimination. The commutation relations of all the bits can be summarized as follows:

$$|s_i\rangle, |s_j\rangle \in S \implies \begin{array}{l} [s_i, s_j] = 0 \iff |i-j| \neq N \\ \{s_i, s_j\} = 0 \iff |i-j| = N \end{array} \tag{2}$$

The advantage of keeping track of additional bits is that it is easy to determine if a measurement g that commutes with the stabilizer set $\mathcal{G} = \{g_1 ... g_r\}$ is contained in the stabilizer group $\mathcal{S}$ or not. The two situations correspond to trivial and logical errors, respectively. To distinguish between the cases, we need to check if

$$g = \prod_{i \in A} g_i \iff |g\rangle = \sum_{i \in A} |g_i\rangle \tag{3}$$

where $A \subset \{1, ...r\}$. Originally, we solved this as a linear equation using row reduction or gaussian elimination, which takes $\mathcal{O}(N^3)$ time in practice. From the commutation relations, it is clear that if g is in the image of $\mathcal{G}$, then

$$A = \{i | \{h_i, g\} = 0\} \tag{4}$$

Thus the cases can be distinguished by simply checking if $|g\rangle = \sum_{i \in A} |g_i\rangle$. This takes $\mathcal{O}(N^2)$ time since it takes $\mathcal{O}(N)$ time to check if two elements commute or anti-commute, and we need to calculate the commutator with all r destabilizers.

Updating S in $\mathcal{O}(N^2)$

First, to simplify notation, let $\bar{i} = \begin{cases} i + N & i \leq N \\ i - N & i > N \end{cases}$ refer to the column which anti-commutes with i.

We describe how to update S each step:

1. Unitary evolution: to evolve by g, update each column of S as usual

$$S \to (\mathbb{I} + |g\rangle \langle g| \epsilon) S \tag{5}$$

2. Correctable measurement: in this case, let $|g_p\rangle = |s_p\rangle$, $p \leq r$, be the first stabilizer to anti-commute with g. For every column $|s_j\rangle \in S$ s.t. $\{s_j, g\} = 0$, $j \neq p$, update $|s_j\rangle \to |s_j\rangle + |s_p\rangle$. This makes it so that the only column that anticommutes with the measurement g is the stabilizer s_p. Note that doing this only messes up the commutation equations for the destabilizer $|s_{\bar{p}}\rangle = |h_p\rangle$. Then, update $s_{\bar{p}} \to s_p$ and $s_p \to g$. Now s_p and

[1] This algorithm, due to Aaronson and Gottesman, is described here: https://arxiv.org/abs/quant-ph/0406196

$s_{\bar{p}}$ anti-commute, and the remaining columns s_k, $k \neq p, \bar{p}$, commute with s_p since the first step made it so that all the other columns commute with g. This takes $\mathcal{O}(N^2)$ time since we have to update $\mathcal{O}(N)$ columns, each update taking $\mathcal{O}(N)$ time.

3. Trivial error: No update is needed.

4. Logical error: In this case, g anti-commutes with one of the remaining bits s_p, $r+1 \leq p \leq N$ or $N+r+1 \leq p \leq 2N$. Otherwise, g would commute with each of the stabilizers $\mathcal{G}$ and the additional bits x_i, z_i, which means its not in the destabilizer set or any of the remaining bits, which means its in the stabilizer set which would make it a trivial error. Let s_p be one of the remaining bits which anti-commutes with g. Then, as in correctable measurement case, multiply all the columns which anticommute with g by s_p so that only s_p anti-commutes with g. Then, as before, $\left|s_{\bar{p}}\right\rangle \to \left|s_p\right\rangle$ and $\left|s_p\right\rangle \to |g\rangle$. This restores the commutation relations, but now $\left|s_p\right\rangle, \left|s_{\bar{p}}\right\rangle$ must be added to the stabilizer and destabilizer set, respectively. This can be done by swapping columns p ($\bar{p}$) and $r+1$ ($N+r+1$) and incrementing r. This has the same running time as if we have a correctable measurement.