

# Ph 219c/CS 219c

## Exercises

Due: Wednesday 29 April 2009

### 7.1 Generating the Clifford Group

Recall that the  $n$ -qubit *Pauli group* is defined as

$$\mathcal{P}_n = \{\mathbf{I}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}\}^{\otimes n} \times \{\pm 1, \pm i\} \quad (1)$$

where  $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$  are the  $2 \times 2$  Pauli matrices. That is, each element of  $\mathcal{P}_n$  is (up to an overall phase  $\pm 1, \pm i$ ) a tensor product of Pauli matrices and identity matrices acting on the  $n$  qubits. The  $n$ -qubit *Clifford group*  $\mathcal{C}_n$  is the *normalizer* of the Pauli group – a unitary operator  $\mathbf{U}$  acting on  $n$  qubits is contained in  $\mathcal{C}_n$  iff

$$\mathbf{U}\mathbf{M}\mathbf{U}^{-1} \in \mathcal{P}_n \text{ for each } \mathbf{M} \in \mathcal{P}_n . \quad (2)$$

That is,  $\mathbf{U}$  acting by conjugation takes a tensor product of Pauli matrices to a tensor product of Pauli matrices. Actually, an element of the Clifford group is defined as this action by conjugation, so that the overall phase of  $\mathbf{U}$  is not relevant.

In this exercise, you will show that the Clifford group can be generated by three quantum gates: the single-qubit gates  $\mathbf{H}$  and  $\mathbf{P}$ , and the two-qubit gate CNOT =  $\Lambda(\mathbf{X})$ . Here  $\mathbf{H}$  denotes the Hadamard gate

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (3)$$

(a rotation by  $\pi$  about the axis  $\hat{x} + \hat{z}$ ), and  $\mathbf{P}$  denotes the phase gate

$$\mathbf{P} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \quad (4)$$

(a rotation by  $\pi/2$  about the  $\hat{z}$  axis). It follows that quantum circuits constructed from these gates can be efficiently simulated by a classical computer, because the action of  $\mathcal{C}_n$  on  $\mathcal{P}_n$  can be succinctly described and easily updated after each gate.

- a) Compute how  $\mathbf{H}$ ,  $\mathbf{P}$ , and  $\Lambda(\mathbf{X})$  act on Pauli operators by conjugation, and verify that  $\mathbf{H}$  and  $\mathbf{P}$  are in  $\mathcal{C}_1$  and that  $\Lambda(\mathbf{X})$  is in  $\mathcal{C}_2$ .
- b) Show that  $\mathbf{H}$  and  $\mathbf{P}$  generate  $\mathcal{C}_1$ . [**Hint:** Note that the elements of the one-qubit Clifford group are the permutations of  $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ , with minus signs appropriately chosen so that the product  $\mathbf{XYZ} = i\mathbf{I}$  remains invariant.]
- c) Let  $\Lambda(\sigma)$  denote the two-qubit gate that applies  $\sigma$  to the target qubit if the control qubit is  $|1\rangle$ , and acts trivially if the control qubit is  $|0\rangle$ . Let  $\sigma_j$  denote  $\sigma$  acting on qubit  $j$ . Show that  $\Lambda(\mathbf{Z})$  and  $\Lambda(\mathbf{Y})$  can be constructed from  $\Lambda(\mathbf{X})$ ,  $\mathbf{H}$ , and  $\mathbf{P}$ . Show that

$$\Lambda(\sigma)\mathbf{Z}_1\Lambda(\sigma) = \mathbf{Z}_1, \quad \Lambda(\sigma)\mathbf{X}_1\Lambda(\sigma) = \mathbf{X}_1\sigma_2, \quad (5)$$

where qubit 1 is the control of the  $\Lambda(\sigma)$  and qubit 2 is its target. Here  $\sigma$  is one of  $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ , so that in particular  $\sigma^2 = I$ .

We will prove that  $\mathbf{H}$ ,  $\mathbf{P}$ , and  $\Lambda(\mathbf{X})$  generate  $\mathcal{C}_n$  by induction. We have already shown (b). Now assume, as an inductive hypothesis, that  $\mathbf{H}$ ,  $\mathbf{P}$ , and  $\Lambda(\mathbf{X})$  generate  $\mathcal{C}_n$ . We need to show that they generate  $\mathcal{C}_{n+1}$ .

- d) Suppose that  $\mathbf{U}$  is an element of  $\mathcal{C}_{n+1}$ . Show that there is a  $\mathbf{W}$  generated by  $\mathbf{H}$ ,  $\mathbf{P}$ , and  $\Lambda(\mathbf{X})$  such that the action of  $\mathbf{WU}$  by conjugation is

$$\begin{aligned} \mathbf{WU} : \mathbf{X}_1 &\rightarrow \mathbf{X}_1\mathbf{M} \\ \mathbf{Z}_1 &\rightarrow \mathbf{Z}_1\mathbf{N} . \end{aligned} \quad (6)$$

where each of  $\mathbf{M}, \mathbf{N}$  is a tensor product of Pauli matrices acting on qubits 2 through  $n+1$ .

- e) Now consider

$$\mathbf{V} \equiv \Lambda(\mathbf{M})\mathbf{H}_1\Lambda(\mathbf{N})\mathbf{H}_1\mathbf{WU} , \quad (7)$$

where  $\Lambda(\mathbf{M})$  denotes the transformation controlled by the first qubit that applies  $\mathbf{M}$  to the other  $n$  qubits, and similarly for  $\Lambda(\mathbf{N})$ . Note that  $\Lambda(\mathbf{M})$  and  $\Lambda(\mathbf{N})$  can be constructed from  $\mathbf{H}$ ,  $\mathbf{P}$ , and  $\Lambda(\mathbf{X})$ . It follows from (c) that the action of  $\Lambda(\mathbf{M})$  by conjugation is

$$\begin{aligned} \Lambda(\mathbf{M}) : \mathbf{X}_1 &\rightarrow \mathbf{X}_1\mathbf{M} \\ \mathbf{Z}_1 &\rightarrow \mathbf{Z}_1 . \end{aligned} \quad (8)$$

Show that the action of  $V$  by conjugation is

$$\begin{aligned} V : \mathbf{X}_1 &\rightarrow \mathbf{X}_1 \\ \mathbf{Z}_1 &\rightarrow \mathbf{Z}_1 . \end{aligned} \quad (9)$$

f) Show that  $V$  is in  $\mathcal{C}_n$ , and therefore can be constructed from  $\mathbf{H}$ ,  $\mathbf{P}$ , and  $\Lambda(\mathbf{X})$ . Show that  $U$  can also be constructed from  $\mathbf{H}$ ,  $\mathbf{P}$ , and  $\Lambda(\mathbf{X})$ . This completes the inductive step, and the proof.

## 7.2 Correcting a shift

Operators acting on a  $d$ -level quantum system (or *qudit*) can be expanded in terms of the  $d^2$  “Pauli operators”

$$X^a Z^b, \quad a, b = 0, 1, 2, \dots, d-1 . \quad (10)$$

Here  $X$  and  $Z$  are generalizations of the Pauli matrices  $\sigma_x$  and  $\sigma_z$ , which act in a particular basis  $\{|j\rangle, j = 0, 1, 2, \dots, d-1\}$  according to

$$\begin{aligned} X &: |j\rangle \rightarrow |j+1 \pmod{d}\rangle , \\ Z &: |j\rangle \rightarrow \omega^j |j\rangle , \end{aligned} \quad (11)$$

where  $\omega = \exp(2\pi i/d)$ . Note that it follows that

$$ZX = \omega XZ . \quad (12)$$

An “error” acting on a qudit can be expanded in this basis.

Suppose that errors with  $|a|, |b|$  small compared to  $d$  are common, but errors with large  $|a|$  and  $|b|$  are rare. We wish to design a quantum error-correcting code that corrects these small “shifts” in the amplitude or phase of the qudit.

For  $d = nr_1 r_2$  (where  $n, r_1$  and  $r_2$  are positive integers), consider the stabilizer generators

$$M_X = X^{nr_1}, \quad M_Z = Z^{nr_2} . \quad (13)$$

- a) Verify that  $M_X$  and  $M_Z$  commute.
- b) Find the commutation relations of  $M_X$  with  $X^a Z^b$  and of  $M_Z$  with  $X^a Z^b$ .

- c) Find two generators of the normalizer group of the code (the group of Pauli operators that commute with the stabilizer). What commutation relations are satisfied by these normalizer generators? What is the dimension of the code subspace?
- d) How large an amplitude shift  $|a|$  and phase shift  $|b|$  can be corrected by this code?

### 7.3 Polynomial CSS codes

Consider a *pit* that takes the  $p$  possible values  $\{0, 1, 2, \dots, p-1\}$ , where  $p$  is prime. The set  $\{0, 1, 2, \dots, p-1\}$  can be regarded as a finite field  $\mathbb{F}_p$  with addition and multiplication defined modulo  $p$ ;  $\mathbb{F}_p$  is a field because each nonzero element has a multiplicative inverse.

In this exercise, you will study the properties of a family of quantum codes for *qubits* ( $p$ -level quantum systems). These quantum codes are related to linear classical codes that are vector spaces over the field  $\mathbb{F}_p$ . We will refer to the quantum codes as *polynomial CSS codes*.

Let  $x_0, x_1, \dots, x_{n-1}$  (where  $n \leq p$ ) be specified distinct elements of  $\mathbb{F}_p$ , and consider a classical code  $C_1$  that contains all strings of  $n$  elements of  $\mathbb{F}_p$  of the form

$$\left( f(x_{n-1}), f(x_{n-2}) \dots, f(x_2), f(x_1), f(x_0) \right), \quad (14)$$

where  $f(x)$  is a polynomial of degree at most  $m$  with coefficients in  $\mathbb{F}_p$ . (The code depends on how the elements  $x_0, x_1, \dots, x_{n-1}$  of  $\mathbb{F}_p$  are chosen. Different codewords within the code are obtained by varying the polynomial  $f$ .)

- a) Show that  $C_1$  is a vector space over  $\mathbb{F}_p$ .
- b) The *weight* of a vector in  $\mathbb{F}_p^n$  is the number of nonzero components of the vector, and the *distance* of a linear code is the minimum weight of a nonzero vector in the code. Show that the distance  $d_1$  of  $C_1$  satisfies

$$d_1 \geq n - m. \quad (15)$$

**[Hint:** A nonzero polynomial  $f(x)$  of degree  $m$  has at most  $m$  zeros over the field  $\mathbb{F}_p$ .]

Now let  $C_2$  be the subcode of  $C_1$  such that  $f(x)$  has degree at most  $m - 1$ .

- c) Show that  $C_2$  is a vector space over  $\mathbb{F}_p$ , and a subspace of  $C_1$ .
- d) Suppose that  $\{z_1, z_2, \dots, z_m\}$  are distinct elements of  $\mathbb{F}_p$ , and that  $\{y_1, y_2, \dots, y_m\}$  are arbitrary elements of  $\mathbb{F}_p$  (not necessarily distinct). Show that there is a polynomial  $f(x)$  of degree less than  $m$  such that

$$\begin{aligned} f(z_1) &= y_1, \\ f(z_2) &= y_2, \\ &\cdot \\ &\cdot \\ f(z_m) &= y_m. \end{aligned} \tag{16}$$

[**Hint:** It is easy to construct such a polynomial  $f(x)$  explicitly.]

- e) The code  $C_2^\perp$  dual to  $C_2$  contains all vectors in  $\mathbb{F}_p^n$  that are orthogonal to all vectors in  $C_2$ . Show that the distance  $d_2$  of  $C_2^\perp$  satisfies

$$d_2 \geq m + 1. \tag{17}$$

[**Hint:** Choose any  $m$  components of the  $n$ -component  $C_2$  codewords, and consider the corresponding projection of  $C_2$  into  $\mathbb{F}_p^m$ . Using the result of (d), show that the image of  $C_2$  under this projection is all of  $\mathbb{F}_p^m$ . Conclude that any vector orthogonal to all vectors in  $C_2$  must have weight at least  $m + 1$ .]

- f) Now consider a quantum error-correcting code of the CSS type, based on the codes  $C_1$  and  $C_2 \subset C_1$ . We can choose a basis for the code space such that each element of the basis is a uniform superposition of all  $C_1$  codewords that belong to the same  $C_2$  coset. What is the number of encoded qubits? (How many distinct  $C_2$  cosets are contained in  $C_1$ ?)
- g) A CSS code can correct  $t$  errors if  $d_1 \geq 2t + 1$  and  $d_2 \geq 2t + 1$ . Explain how to construct polynomial CSS codes that encode one qubit, correct  $t$  errors, and have block size  $4t + 1$ . For what values of  $p$  can such codes be constructed?

## 7.4 Good CSS codes

In class we derived the *quantum Gilbert-Varshamov bound*:

$$|\mathcal{E}^{(2)}| - 1 < \frac{2^{2n} - 1}{2^{n+k} - 2^{n-k}}. \tag{18}$$

This is a sufficient condition for the existence of a (possibly degenerate) binary stabilizer code that can correct all Pauli operators in a set  $\mathcal{E}$ ; here  $|\mathcal{E}^{(2)}|$  denotes the number of distinct Pauli operators of the form  $E_a^\dagger E_b$ , where  $E_a, E_b \in \mathcal{E}$ . One consequence of this bound is that there exist “good”  $[[n, k, d = 2t + 1]]$  stabilizer codes that achieve an asymptotic rate  $R = k/n = 1 - H_2(2t/n) - (2t/n) \log_2 3$ .

The purpose of this exercise is to show that good Calderbank-Shor-Steane (CSS) codes exist.

- a) Derive a quantum Gilbert-Varshamov bound for CSS codes. Denote by  $\mathcal{E}^X$  the set of  $X$ -type errors that the code can correct (those that can be expressed as a tensor product of  $X$ 's and  $I$ 's) and denote by  $\mathcal{E}^Z$  the set of  $Z$ -type errors that the code can correct (those that can be expressed as a tensor product of  $Z$ 's and  $I$ 's). Denote by  $\mathcal{E}^{X(2)}$  the set of  $\{E_a^\dagger E_b\}$  where  $E_a, E_b \in \mathcal{E}^X$ , and similarly for  $\mathcal{E}^{Z(2)}$ . The quantum Gilbert-Varshamov bound for CSS codes is a sufficient condition for the existence of a CSS code with  $n_X$  stabilizer generators of the  $X$  type and  $n_Z$  stabilizer generators of the  $Z$  type that can correct all errors in  $\mathcal{E}^X$  and  $\mathcal{E}^Z$ , expressed as an inequality involving  $n_X$ ,  $n_Z$ ,  $|\mathcal{E}^{X(2)}|$  and  $|\mathcal{E}^{Z(2)}|$ .
- b) Use the quantum Gilbert-Varshamov bound for CSS codes to show the existence of CSS codes that achieve the asymptotic rate  $R = k/n = 1 - H_2(2t_X/n) - H_2(2t_Z/n)$ , where the code can correct  $t_X$   $X$  errors and  $t_Z$   $Z$  errors.