Obligatory Exercises MAT3420

Linus Ekstrøm

March 2021

Contents

1 Project Text

Theorem 1 *For quantum computers with at least 3 qubits, the following gates form a universal gate set*

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \quad K = \begin{bmatrix}1 & 0\\ 0 & i\end{bmatrix} \quad \textit{and}$$
$$K^* = K^{-1} = \begin{bmatrix}1 & 0\\ 0 & -i\end{bmatrix} \quad CNOT = C(X) \quad C^2(X)$$

Task 1)

With X, Y, Z denoting the Pauli matrices check that

$$Y = KXK^{-1}, \quad Z = HXH$$

Answer: This one is fairly simple,

$$\begin{aligned} Y &= \begin{bmatrix}1 & 0\\ 0 & i\end{bmatrix}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & -i\end{bmatrix} \\ &= \begin{bmatrix}0 & 1\\ i & 0\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & -i\end{bmatrix} = \begin{bmatrix}0 & -i\\ i & 0\end{bmatrix} = Y \end{aligned}$$

Where we have used that since K is unitary its inverse equal to its own conjugate transpose, in this case K is diagonal and thus its inverse is equal to its conjugate. Next we check the Z expression

$$\begin{aligned} Z &= \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \\ &= \frac{1}{2}\begin{bmatrix}1 & 1\\ -1 & 1\end{bmatrix}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} = \frac{1}{2}\begin{bmatrix}2 & 0\\ 0 & -2\end{bmatrix} = Z \end{aligned}$$

We see that both identities indeed are correct. Next we are to conclude that $C^2(i)$, *where* i *denotes the scalar operator of multiplication by* i, can be written as

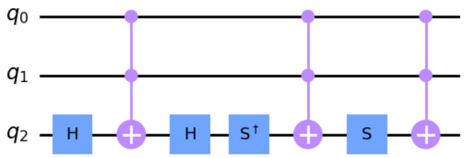


Figure 1: Image produced using Qiskit's draw function. Note that in Qiskit they name the K-gate and its inverse S and $S^\dagger$ respectively.

This was solved in three/four different ways depending on how you look at it. Initially, I just experimented to see what would happen if I ignored the tensor products. The reason I did this was I thought that most of the matrices would be similar to the identity matrix for the two top qubits. (Note, I realize now this is not the reason it works out like this), but this was my initial thoughts, so I calculated

$$\begin{aligned} M = \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}&\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \\ &\begin{bmatrix}1 & 0\\ 0 & -i\end{bmatrix}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & i\end{bmatrix}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} \\ &= \begin{bmatrix}i & 0\\ 0 & i\end{bmatrix} \end{aligned} \tag{1}$$

Which is exactly the scalar operator of multiplication by i on one qubit. However, looking at this I noticed this above calculation can be simplified to

$$\begin{aligned} M = ZY^{-1}X &= \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}\begin{bmatrix}0 & i\\ -i & 0\end{bmatrix}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} \\ &= \begin{bmatrix}i & 0\\ 0 & i\end{bmatrix} \end{aligned} \tag{2}$$

So, finding this was encouraging and showed I was on the right track. Next I implemented the circuit in Qiskit

QisKit implementation

```
import qiskit as qk
circ = qk.QuantumCircuit(3)
circ.h(2)
circ.ccx(0, 1, 2)
circ.h(2)
circ.sdg(2)
circ.ccx(0, 1, 2)
circ.s(2)
circ.ccx(0, 1, 2)
```

with this code you can analyse various features of the circuit. I mainly used the Operator class which can be imported from qiskit.quantum_info. This class lets you print out the for the defined circuit. However, it did not match with what I expected so I browsed around in the documentation and the qiskit textbook and discovered the reason was due to the qubit indexing that qiskit uses. I decided on looking at the Operator from the circuit because I was not 100% sure of which order I should have the tensor products i.e.

$$\begin{aligned} &\big(H\otimes I\otimes I\big)C^2(X)\big(H\otimes I\otimes I\big) \\ &\quad\big(K^*\otimes I\otimes I\big)C^2(X)\big(K\otimes I\otimes I\big)C^2(X) \\ &\qquad\qquad \textit{or} \\ &\big(I\otimes I\otimes H\big)C^2(X)\big(I\otimes I\otimes H\big) \\ &\quad\big(I\otimes I\otimes K^*\big)C^2(X)\big(I\otimes I\otimes K\big)C^2(X) \end{aligned} \tag{3}$$

If I have understood correctly this feels like it just depends on the choice of basis (not entirely sure if basis is the correct word to use here). So, in order to convince myself that it is indeed correct to use the second one I quickly coded the equation into python, because actually multiplying them out seemed tiresome (A). So I had convinced myself numerically that it was the case that the circuit did indeed implement the $C^2(i)$ unitary. But I still do not feel like my discussion up until this point has proved that the circuit implements the gate, however I came across the notation

$$\left(I \otimes I \otimes H\right) = \begin{bmatrix} H & 0 & H & 0 \\ 0 & H & 0 & H \\ H & 0 & H & 0 \\ 0 & H & 0 & H \end{bmatrix}$$

where 0 denotes the zero-matrix. And finally I realised if you write the whole circuit expression in this form it perfectly retrieves ((1), (2)). So I can conclude that my original intuition that since the Toffoli-gate is similar to the identity gate everywhere except the lower right two by two 'matrix' and we apply two Hadamards $HH = I$ and $K^{-1}K = I$ it is indeed correct to simplify the whole (3) to (1).

Task 2

a)

This task asks for a proof of

$$C^2(i) = C(K) \otimes I$$

Answer: Since I had already used my Python script to convince myself numerically that I understood the tensor products, I just implemented a test for this. Now I know this is not a proof but it was a good start. Using the notation discussed at the end of the last section it is easy to see it must be the case, and using a script for this (after I realized how to do it) seems overkill

$$C(K) \otimes I = \begin{bmatrix} 1I & 0 & 0 & 0 \\ 0 & 1I & 0 & 0 \\ 0 & 0 & 1I & 0 \\ 0 & 0 & 0 & iI \end{bmatrix} = C^2(i)$$

Not exactly sure if this counts as a mathematical proof but it does show the equality.

b)

Conclude that Theorem (1) follows from

Theorem 2 *For quantum computers with at least 2 qubits, the gates $H, C(K), C(K^*)$ form a universal gate set.*

To show this all we have to do is show that we can construct all the gates defined in Theorem (1) with the gates defined in Theorem (2). We can perform the single qubit operations K and K^*, pretty much by definition, with the $C(K)$ and $C(K^*)$ gates where the control qubit is set to $|1\rangle$. The remaining operations of $CNOT$ and $C^2(X)$ can be constructed from our gate set as follows

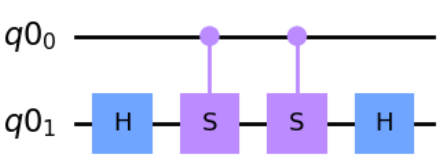


Figure 2: The circuit which implements the $C(X)$ gate from the basis defined in Theorem (2)

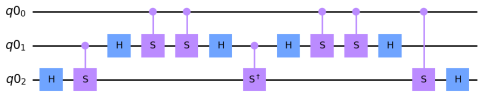


Figure 3: The circuit which implements the $C^2(X)$ gate from the basis defined in Theorem (2). I used the implementation from the above figure to build the required $C(X)$ gates for no other reason then to see if I could. The code used to produce both these circuit drawings in qiskit is listed in the appendix (A)

The matrix I got from my script which is from qiskit's Operator function which works on the circuit I set up is clearly the CNOT and the Toffoli gate respectively. Thus we can conclude that Theorem (1) follows from Theorem (2)

$$C(X) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$C^2(X) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Task 3

The following problem was given, *and can be assumed to be solved in two dimensions*. The goals is to prove the following generalization by reducing the main part to the already known case.

Assume $\mathcal{H}$ is a finite dimensional Hilbert space of dimension at least 2. Let $|\phi_1\rangle$ and $|\phi_2\rangle$ be two unit vectors in $\mathcal{H}$ not lying on the same line. Let $G_i(i = 1, 2)$ be the set of unitary operators U on $\mathcal{H}$ such that $U|\phi_i\rangle = |\phi_i\rangle$ and $\det U = 1$. Let G be the group generated by G_1 and G_2, that is, the set of unitary operators that can be written as finite products of G_1 and G_2.

a)

Show that for every unit vector $|\psi\rangle \in \mathcal{H}$ there is an element $U \in G$ such that $U|\psi\rangle = |\phi_1\rangle$.

hint: first show that there is an element V of G_1 or G_2 such that $V|\psi\rangle$ lies in the two-dimensional subspace spanned by $|\phi_1\rangle$ and $|\phi_2\rangle$

Answer: The project text says that $|\phi_1\rangle$ is an eigenvector of all unitaries $U \in G_1$ with eigenvalues $\lambda_1 = 1$, similarly $|\phi_2\rangle$ is an eigenvector of all unitaries $U \in G_2$ with eigenvalues $\lambda_2 = 1$. In addition it says that $|\phi_1\rangle$ and $|\phi\rangle_2$ are linearly independent.

First a digression about *unitary matrices.* Unitaries are the complex matrix analogy to orthogonal real matrices[1] Structurally, the represent rotations and reflections and as such it makes sense that they preserve norms[2]. Another way to say this is that they preserve the Hermitian inner product[18]. For this reason they are very important in quantum mechanics because preserving norms means preserving probability amplitudes. Said in another way, when acting on a quantum state with a unitary matrix you never end up with a situation where the total probability for 'all things happening' is larger than 1. A common definition for unitary matrices is

$$U^*U = UU^* = I = U^\dagger U = UU^\dagger$$

where U^* is the conjugate transpose of U and $U^\dagger$ is the Hermitian adjoint of U. Really just two 'words' for the exact same thing, although the latter is widely used in physics texts.

Moving onto the problem, we define

$$|\psi\rangle = \alpha|\phi_1\rangle + \beta|\phi_o\rangle$$

where we have constructed $|\phi_0\rangle$ from $|\phi_1\rangle$ and $|\phi_2\rangle$ using Gram-Schmidt. Following this, we can set the standard requirement on α and β, $|\alpha|^2 + |\beta|^2 = 1$. Next we let a unitary V be from either G_1 or G_2. Now since V is unitary with $\det V = 1$, meaning it does not squish to a lower space[3], $\beta V|\phi_o\rangle$ and $\alpha V|\phi_1\rangle$ will still lie in the same space. Another way of thinking of this is that they will, in some way, be rotated / reflected by the transformation which will leave them in the same space that they started in. Thus, if we have a $V \in G_1$ or G_2 the action on a general unit vector $|\psi\rangle$ will leave it in the span of $|\phi_1\rangle$ and $|\phi_o\rangle$. Next if we want to obtain a unitary $U \in G$, such that

$$U|\psi\rangle = |\phi_1\rangle$$

all we really need to to is find a unitary such that for every application of it we shrink the absolute value of β, and correspondingly increase the absolute value of α. Keeping in mind here that $|\alpha|^2 + |\beta|^2 = 1$, what this 'translates' to is that we find a sequence of unitaries $U = V_nV_{n-1}...V_0$ so that we for each one rotate $\beta|\phi_o\rangle$ closer and closer to $\alpha|\phi_1\rangle$[4]. Basically, we find rotations that walk $|\phi_o\rangle$ over to $|\phi_1\rangle$, this works since all the V's keep $|\psi\rangle$ in the Span($|\phi_1\rangle$, $|\phi_o\rangle$).

b

Show that G coincides with the entire special unitary group $SU(\mathcal{H})$, that is, the set of unitary operators with determinant one.

Answer: The *special unitary group $SU(\mathcal{H})$* is the group of matrices, defined as[5][7]

$$\begin{aligned} SU(n) &= \{P \in U(n) |\det P = 1\} \\ &= \{P \in GL_n(\mathbb{C}) | P^\dagger P = I, \det P = 1\} \end{aligned}$$

for an $n \times n$ matrix P and where $GL_n(\mathbb{C})$ is the general linear group. Further, every $P \in SU(n)$ can be shown to be written as

$$P = \begin{bmatrix} a & b \\ -\bar{b} & \bar{a} \end{bmatrix}$$

where $a, b \in \mathbb{C}$ and $|a|^2 + |b|^2 = 1$. Now looking at what we are supposed to show, it does intuitively feel as though $G = SU(n)$ where n is the dimension of $\mathcal{H}$, but we are asked to show it.

After some discussions with fellow students I am not so sure whether this is what coincides means, this is how I understood it, but there's this notion that even if you can make all elements in one set by combinations of things from the other set they're still not the same. I'm not entirely sure, *because this is the first time dealing with groups like this and equalities of them*, but I suspect that here it is more accurate to say that we can exactly make all unitaries of $SU(\mathcal{H})$ because we can rotate any vector in $\mathcal{H}$ onto every other vector also in $\mathcal{H}$. Also if, we like in task a), find a unitary that rotates $|\psi\rangle$ onto $|\phi\rangle$ then the inverse unitary will transfer $|\phi\rangle$ to $|\psi\rangle$, and this inverse will necessarily also be in G and thus we can say $SU(\mathcal{H}) = G$.

After more thinking I have come to the realization that in the above paragraph I need to try to be more

[1] at least how I understand it.

[2] Actually it seems that the additional requirement of $\det U = 1$ is necessary for this to be strictly true.

[3] Taking some liberty and just using some intuition here.

[4] Unsure if I here should keep the α and β when discussing this.

[5] Everywhere else I see it defined $SU(n)$ so I'll assume what is meant by $SU(\mathcal{H})$ is the special unitary group with dimension matching the Hilbert space we are talking about. This is probably implicitly understood but just to be sure I'm mentioning it here.

specific. Trying to write it out a bit

$$U \in G \;\; s.t \;\; U\left|\psi\right\rangle = \left|\phi_1\right\rangle \;\; \text{by a)}$$

Since U by definition is unitary with determinant equal 1 then we also have $U \in SU(\mathcal{H})$. Now I think if we make some other arbitrary unitary $V \in SU(\mathcal{H})$ and show that this is also in G then we can say $G = SU(\mathcal{H})$. Now the V we pick is such that $V\left|\psi\right\rangle = \left|\phi_1\right\rangle$. So far we have

$$\begin{aligned} &U \in G \;\; s.t. \;\; U\left|\psi\right\rangle = \left|\phi_1\right\rangle \\ &V \in SU(\mathcal{H}) \;\; s.t. \;\; V\left|\psi\right\rangle = \left|\phi_1\right\rangle \end{aligned}$$

So we can also write and substitute in $\left|\psi\right\rangle = U^*\left|\phi_1\right\rangle = U^{-1}\left|\phi_1\right\rangle$ and therefore we can write

$$VU^*\left|\phi_1\right\rangle = \left|\phi_1\right\rangle$$

which means that $VU^* \in G$, because it returns $\left|\phi_1\right\rangle$ to $\left|\phi_1\right\rangle$ so we can make V from VU^* and U which are two elements in G

$$V = (VU^*)U$$

and this is why we can say $G = SU(\mathcal{H})$.

Task 4

Show that Problem 3. implies the following. Assume $\mathcal{H}$ is a finite dimensional Hilbert space of dimension at least 2. Let $\left|\phi\right\rangle$ be a unit vector and G be the set of unitary operators U on $\mathcal{H}$ such that $U\left|\phi\right\rangle = \left|\phi\right\rangle$ and $\det U = 1$. Let V be a unitary operator such that $\left|\phi\right\rangle$ is not an eigenvector of V. Then $SU(\mathcal{H})$ is generated by G and VGV^*, that is every element of $SU(\mathcal{H})$ can be written as a finite product of elements of G and VGV^*

Answer: Here I found it helps to use geometric intuition for my argument. So what we have here is one set of operators G which when acting on $\left|\phi\right\rangle$ just returns itself, while our other set $VGV^* = VGV^{-1}$ can be thought of as a rotation of $\left|\phi\right\rangle$ onto some other unit vector $\left|\eta\right\rangle$ which we then rotate back into *the ϕ-part of* $\mathcal{H}$ (honestly feel like my terminology is lacking here..) and then we perform the inverse of our first rotation. So what have we done here, we have started at a vector $\left|\phi\right\rangle$ which is defined to be the unit vector such that it is an eigenvector for any $U \in G$ and we have rotated it onto some other unit vector $\left|\eta\right\rangle$. Now if we think of this operation in inverse, we have the exact same situation as discussed in the previous task. We can start with an arbitrary unit vector $\left|\eta\right\rangle$ and, using the inverse operations from the previous point, rotate it onto $\left|\phi\right\rangle$. This also naturally ties into the discussion surrounding theorem (3)[6] and equation (6), in which we stated that we can approximate to arbitrary accuracy any rotation $R_{\hat{n}}(\alpha)$ by $R_{\hat{n}}(\theta)^n$. So to summarize, *and try to write it mathematically*, we have

$$\begin{aligned} \left|\phi\right\rangle \;\; s.t \;\; U \in G \to U\left|\phi\right\rangle = \left|\phi\right\rangle \\ and \;\; V \;\; s.t \;\; V\left|\phi\right\rangle \neq \left|\phi\right\rangle \\ next \;\; V\left|\phi\right\rangle = \left|\psi\right\rangle \\ then\; for \;\; U \in G, \;\; VUV^*\left|\psi\right\rangle = \left|\psi\right\rangle \end{aligned}$$

where in the last line we have used that $VUV^*\left|\psi\right\rangle = VU\left|\phi\right\rangle = V\left|\phi\right\rangle = \left|\psi\right\rangle$ because $V\left|\phi\right\rangle = \left|\psi\right\rangle \implies V^*\left|\psi\right\rangle = \left|\phi\right\rangle$. The situation we are in here feels reminiscent of where we were at the end of the last problem. Now we introduce the set G' of unitaries U' with determinant 1 such that

$$U'\left|\psi\right\rangle = \left|\psi\right\rangle$$

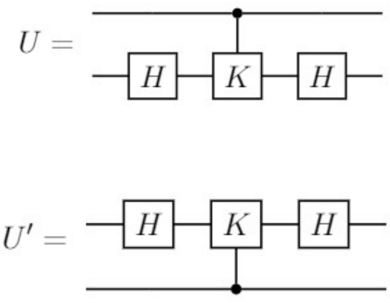
Further we have

$$VUV^*\left|\psi\right\rangle = U'\left|\psi\right\rangle \tag{4}$$

In other words VGV^* generates G' because for each $U \in G$ we have a $U' \in G'$. Now we can use what we did in the problem 3), we see that we have the same situation if we think of VGV^* as G_1 and G as G_2. It then follows that G and VGV^* generate $SU(\mathcal{H})$

Task 5

Consider the gate $U = \big(I \otimes H\big)C(K)(I \otimes H)$. Denote by U' the gate $\Sigma U \Sigma$, where Σ swaps the qubits. In other words, $U' = \big(H \otimes I\big)C(K)(H \otimes I)$



a)

Show that the operators U and U' leave the vectors $\left|00\right\rangle$ and $\left|\eta\right\rangle = \left|01\right\rangle + \left|10\right\rangle + \left|11\right\rangle$ invariant.

Answer: We start by writing the operators in matrix form (*admittedly this took me way to long, was very unsure how to write the C(K) when swapping the qubits, turns out regardless of which qubit is the control it has the same representation. Also regardless of which qubit is defined as the MSB or LSB* [15]). In this case I actually did the multiplication by hand, should probably

[6] A bit messy with referencing something from the task below I know but

have used Python here as well

$$U = \begin{bmatrix} \frac{1}{\sqrt{2}}I & \frac{1}{\sqrt{2}}I \\ \frac{1}{\sqrt{2}}I & -\frac{1}{\sqrt{2}}I \end{bmatrix} C(K) \begin{bmatrix} \frac{1}{\sqrt{2}}I & \frac{1}{\sqrt{2}}I \\ \frac{1}{\sqrt{2}}I & -\frac{1}{\sqrt{2}}I \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

$$\times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2}+\frac{i}{2} & 0 & \frac{1}{2}-\frac{i}{2} \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{2}-\frac{i}{2} & 0 & \frac{1}{2}+\frac{i}{2} \end{bmatrix}$$

Same procedure for U', we have

$$U' = \begin{bmatrix} 1H & 0 \\ 0 & 1H \end{bmatrix} C(K) \begin{bmatrix} 1H & 0 \\ 0 & 1H \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

$$\times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2}+\frac{i}{2} & \frac{1}{2}-\frac{i}{2} \\ 0 & 0 & \frac{1}{2}-\frac{i}{2} & \frac{1}{2}+\frac{i}{2} \end{bmatrix}$$

With both of these it is very easy to check that both $|00\rangle$ and $|\eta\rangle$ are eigenvectors of both U and U' with eigenvalues 1.

$$U\,|00\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2}+\frac{i}{2} & 0 & \frac{1}{2}-\frac{i}{2} \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{2}-\frac{i}{2} & 0 & \frac{1}{2}+\frac{i}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$U\,|\eta\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2}+\frac{i}{2} & 0 & \frac{1}{2}-\frac{i}{2} \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{2}-\frac{i}{2} & 0 & \frac{1}{2}+\frac{i}{2} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ \frac{1}{2}+\frac{1}{2}+\frac{i}{2}-\frac{i}{2} \\ 1 \\ \frac{1}{2}+\frac{1}{2}+\frac{i}{2}-\frac{i}{2} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Thus we have showed that both vectors are eigenvectors for U. Now all that remains is to show it for U'.

$$U'\,|00\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2}+\frac{i}{2} & \frac{1}{2}-\frac{i}{2} \\ 0 & 0 & \frac{1}{2}-\frac{i}{2} & \frac{1}{2}+\frac{i}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$U'\,|\eta\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2}+\frac{i}{2} & \frac{1}{2}-\frac{i}{2} \\ 0 & 0 & \frac{1}{2}-\frac{i}{2} & \frac{1}{2}+\frac{i}{2} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 1 \\ \frac{1}{2}+\frac{1}{2}+\frac{i}{2}-\frac{i}{2} \\ \frac{1}{2}+\frac{1}{2}+\frac{i}{2}-\frac{i}{2} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Now I have showed that, for both U and U', $|00\rangle$ and $|\eta\rangle$ are eigenvectors both with eigenvalue 1. I do feel however that this was of showing it is a bit overkill, but it also does help to actually work through the details to see why it is true. The first thing I thought when looking at this problem was that in some way it is 'obvious'[7] that there is some symmetry of the system here, and that it should not really affect the end result whether you use the above or below qubit for the calculation in this setup. The actual particles should not care whether one or the other is the control qubit. I'm not sure how to prove this mathematically, and here I am asking for feedback as to how it is done.

b)

Let $\mathcal{K}$ be the orthogonal complement of the span of $|00\rangle$ and $|\eta\rangle$. Consider the operators $V = U^*U'$ and $V' = U'U^*$. Show that the restrictions of V and V' to $\mathcal{K}$ do not commute and have eigenvalues

$$\lambda_\pm = \frac{1 \pm i\sqrt{15}}{4}$$

Answer: I think the orthogonal complement of $\mathcal{V} = \mathrm{Span}(|00\rangle, |\eta\rangle)$ is the space where all vectors dotted with either of those two equals the zero vector, so $\mathcal{K} = \mathcal{V}^\perp$. So since we have a basis for $\mathcal{V}$ then we can find $\mathcal{K}$ as follows

$$\langle x, |00\rangle\rangle = 0 \quad and \quad \langle x, |\eta\rangle\rangle = 0$$

This gives us a system of equations with the solution being the orthogonal complement. One basis for the orthogonal complement $\mathcal{K}$ is

$$|u_1\rangle = \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix} \quad and \quad |u_2\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}$$

[7] I hate when math and physics text say something is obvious and then don't show the details, if it is so obvious then it shouldn't take that long to explain why it is so!

Moving on we explicitly calculate V and V'

$$
\begin{aligned}
V &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2}-\frac{i}{2} & 0 & \frac{1}{2}+\frac{i}{2} \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{2}+\frac{i}{2} & 0 & \frac{1}{2}-\frac{i}{2} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2}+\frac{i}{2} & \frac{1}{2}-\frac{i}{2} \\ 0 & 0 & \frac{1}{2}-\frac{i}{2} & \frac{1}{2}+\frac{i}{2} \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2}-\frac{i}{2} & (\frac{1}{2}+\frac{i}{2})(\frac{1}{2}-\frac{i}{2}) & (\frac{1}{2}+\frac{i}{2})(\frac{1}{2}+\frac{i}{2}) \\ 0 & 0 & \frac{1}{2}+\frac{i}{2} & \frac{1}{2}-\frac{i}{2} \\ 0 & \frac{1}{2}+\frac{i}{2} & (\frac{1}{2}-\frac{i}{2})(\frac{1}{2}-\frac{i}{2}) & (\frac{1}{2}-\frac{i}{2})(\frac{1}{2}+\frac{i}{2}) \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2}-\frac{i}{2} & \frac{1}{2} & \frac{i}{2} \\ 0 & 0 & \frac{1}{2}+\frac{i}{2} & \frac{1}{2}-\frac{i}{2} \\ 0 & \frac{1}{2}+\frac{i}{2} & -\frac{i}{2} & \frac{1}{2} \end{bmatrix} = U^*U' = V
\end{aligned}
$$

and doing the same for V'

$$
\begin{aligned}
V' &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2}+\frac{i}{2} & \frac{1}{2}-\frac{i}{2} \\ 0 & 0 & \frac{1}{2}-\frac{i}{2} & \frac{1}{2}+\frac{i}{2} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2}-\frac{i}{2} & 0 & \frac{1}{2}+\frac{i}{2} \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{2}+\frac{i}{2} & 0 & \frac{1}{2}-\frac{i}{2} \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2}-\frac{i}{2} & 0 & \frac{1}{2}+\frac{i}{2} \\ 0 & (\frac{1}{2}-\frac{i}{2})(\frac{1}{2}+\frac{i}{2}) & \frac{1}{2}+\frac{i}{2} & (\frac{1}{2}-\frac{i}{2})(\frac{1}{2}-\frac{i}{2}) \\ 0 & (\frac{1}{2}+\frac{i}{2})(\frac{1}{2}+\frac{i}{2}) & \frac{1}{2}-\frac{i}{2} & (\frac{1}{2}+\frac{i}{2})(\frac{1}{2}-\frac{i}{2}) \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2}-\frac{i}{2} & 0 & \frac{1}{2}+\frac{i}{2} \\ 0 & \frac{1}{2} & \frac{1}{2}+\frac{i}{2} & -\frac{i}{2} \\ 0 & \frac{i}{2} & \frac{1}{2}-\frac{i}{2} & \frac{1}{2} \end{bmatrix} = U'U^* = V'
\end{aligned}
$$

Next to show that the restrictions of V and V' to $\mathcal{K}$ do not commute

$$
\begin{aligned}
[VV', V'V] &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\frac{i}{2} & \frac{1}{2}+\frac{i}{2} & \frac{1}{2} \\ 0 & \frac{1}{2}+\frac{i}{2} & 0 & \frac{1}{2}-\frac{i}{2} \\ 0 & \frac{1}{2}+\frac{i}{2} & \frac{1}{2}-\frac{i}{2} & \frac{i}{2} \end{bmatrix} \\
&- \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2}-\frac{i}{2} & \frac{1}{2}+\frac{i}{2} \\ 0 & \frac{1}{2}-\frac{i}{2} & \frac{i}{2} & \frac{1}{2} \\ 0 & \frac{1}{2}+\frac{i}{2} & \frac{1}{2} & -\frac{i}{2} \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -\frac{i}{2} & i & -\frac{i}{2} \\ 0 & i & -\frac{i}{2} & -\frac{i}{2} \\ 0 & -\frac{i}{2} & -\frac{i}{2} & i \end{bmatrix}
\end{aligned}
$$

If the restrictions of V and V' to $\mathcal{K}$ do not commute then for any complex number ξ the vector $|v\rangle = |u_1\rangle + \xi\,|u_2\rangle$ will not be zero when $[VV', V'V]$ acts on it

$$
[VV', V'V]\,|v\rangle = \begin{bmatrix} 0 \\ -\frac{3}{2}\xi \\ \frac{3i}{2}\xi + \frac{3i}{2} \\ -\frac{3i}{2} \end{bmatrix}
$$

It is clear that this is never zero for any choice of ξ, and thus the restrictions of V and V' to $\mathcal{K}$ do not commute.

Next we calculate the eigenvalues of V and V' to check they indeed are the ones given in the task. For most of the time spent on this task I interpreted this sentence "*Show that the restrictions ... and have eigenvalues*" from the project text to mean that we should check that the commutators had eigenvalues equal to $\lambda_\pm$. I've calculated what feels like a million different times the restricted matrices and their commutators, it was just a stroke of luck that I decided to run the script I made to help me calculate quicker A on the matrices V and V' and finally I saw I got the correct values. Checking the given eigenvalues by hand is not too much work either

$$
\begin{aligned}
\det(V' - tI) &= \begin{vmatrix} 1-t & 0 & 0 & 0 \\ 0 & \frac{1}{2}-\frac{i}{2}-t & 0 & \frac{1}{2}+\frac{i}{2} \\ 0 & \frac{1}{2} & \frac{1}{2}+\frac{i}{2}-t & -\frac{i}{2} \\ 0 & \frac{i}{2} & \frac{1}{2}-\frac{i}{2} & \frac{1}{2}-t \end{vmatrix} \\
&= (1-t)\Big((\frac{1}{2}-\frac{i}{2}-t)\Big((\frac{1}{2}-t) \\
&+ \frac{i}{2}(\frac{1}{2}-\frac{i}{2})\Big) + (\frac{1}{2}+\frac{i}{2})\Big(\frac{1}{2}(\frac{1}{2}-\frac{i}{2}) \\
&- \frac{i}{2}(\frac{1}{2}+\frac{i}{2}-t)\Big)\Big) \\
p(t) &= (1-t)\big(-t^3 + \frac{3t^2}{2} - \frac{3t}{2} + 1\big)
\end{aligned}
$$

Now we can input our given eigenvalues $\lambda_\pm$ and we do indeed get $p(\lambda_\pm) = 0$. The same routine could be used to show that we get the same eigenvalues for V, but as I have done the above in addition to providing a script which calculates both eigenvalues I feel this should be enough to show that V and V' indeed have $\lambda_\pm$ as eigenvalues.

c)

Show using properties of algebraic integers that the numbers $\lambda_\pm = (1 \pm i\sqrt{15})/4$ are not roots of unity. Conclude that V and V' generate a dense subset of the set G of unitary operators W such that $W\,|00\rangle = |00\rangle$, $W\,|\eta\rangle = |\eta\rangle$ and $\det W = 1$.

Answer: We start with the definition of a algebraic integer: *A number $z \in \mathbb{C}$ is called an algebraic interger if it is a root of a monic polynomial with integral coefficients. That is, there are $n \geq 1$ and $a_0, ..., a_{n-1} \in \mathbb{Z}$ such that*

$$z^n + a_{n-1}z^{n-1} + ... + a_1 z + a_0 = 0$$

Now if $\lambda_\pm$ are roots of unity then there exists a positive integer n such that

$$\lambda_\pm^n - 1 = 0 \tag{5}$$

Moving on, expressions of the form $x^n - 1$ can be factored into products of cyclotomic polynomials [16]

$$\prod_{d|n} \Phi_d(x) = x^n - 1$$

There are some properties of cyclotomic polynomials we care about here: *1. $\Phi_d(x) \in \mathbb{Z}[x]$ i.e. all of Φ's coefficients are integers, 2. Φ is monic and 3. Φ is irreducible over $\mathbb{Q}[x]$ i.e. it cannot be factored into two non-constant polynomials with rational coefficients.* Next we compute the monic polynomial $p(x) \in \mathbb{Q}[x]$ for $\lambda_\pm$. (Here I initially calculated it for λ_+ but quickly noticed that both of them share the same monic polynomial. I would here like to call it due to their symmetry, but I do not know the correct terms to say it. Said another way: because they are complex conjugates they both have the same monic polynomial.)

$$\begin{aligned} p(x)_{\lambda_\pm} &= \Big(z - (\frac{1+i\sqrt{15}}{4})\Big)\Big(z - (\frac{1-i\sqrt{15}}{4})\Big) \\ &= z^2 - \frac{z}{2} + 1 \end{aligned}$$

where we have used the complex conjugate theorem to make $p(x)_{\lambda_\pm}$. This polynomial is by definition the lowest-powered monic polynomial with $\lambda_\pm$ as the roots. Now since we have a non-integer coefficient, $\frac{1}{2}$, in our polynomial $p(x)_{\lambda_\pm} \neq \Phi_d(x)$ which in turn means that (5) does not hold for a positive integer n, which means that $\lambda_\pm$ are not roots of unity.

Now because, the eigenvalues of the commutators are not roots of unity, we can apply much the same argument as was shown in lecture 11 [11], which seems to be the exact argument from [1] (p.196) "*... Moreover, this θ can be shown to be an irrational multiple of 2π ... Next we show that repeated iteration of $R_{\hat{n}}(\theta)$ can be used to approximate to arbitrary accuracy any rotation $R_{\hat{n}}(\alpha)$*". Now an arbitrary single qubit unitary operator can be written in the form

$$U = \exp\{(i\alpha)\}R_{\hat{n}}(\theta) = \exp\{(i\alpha)\}\exp\Big\{(-i\frac{\theta}{2}(n\cdot\sigma))\Big\}$$

which can be shown by considering the adjoint of this U

$$\begin{aligned} U^\dagger &= (\exp\{(i\alpha)\})^\dagger \exp\Big\{(-i\frac{\theta}{2}(n\cdot\sigma))\Big\}^\dagger \\ &= \exp\{(i\alpha)\}\exp\Big\{(-i\frac{\theta}{2}(n\cdot\sigma))\Big\} = U \end{aligned}$$

This above statement is *exercise 4.8* in [1]. This exercise leads into the following theorem

Theorem 3 ***(Z-Y decomposition for a single qubit)*** *Suppose U is a unitary operation on a single qubit. Then there exists real numbers α, β, γ and δ such that*

$$U = \exp\{i\alpha\}R_z(\beta)R_y(\gamma)R_z(\delta)$$

Now we can follow along with the explanation in either [1] or [11] where we divide up the unit circle in arc segments and prove that we have

$$E(R_{\hat{n}}(\alpha), R_{\hat{n}}(\theta)^n) < \frac{\epsilon}{3} \tag{6}$$

in other words we can approximate to arbitrary accuracy any rotation $R_{\hat{n}}(\alpha)$ by repeated use of $R_{\hat{n}}(\theta)$ which is exactly what is meant by something generating a dense subset. This means we can say that V and V' generate a dense subset of the set G, because for any operator in G we can approximate them using V and V' precisely because their commutators are not roots of unity, i.e. we are not stuck in the same cycle around the unit circle when we increase n, so we will be able to generate all numbers around the unit circle by the arc argument along with the pigeonhole principle.

Task 6

Finish the proof of Theorem (2) by applying Problem 4 twice:

a)

Show that the set G' of unitary operators W such that $W|00\rangle = |00\rangle$ and $\det W = 1$ is generated by G and $C(K)GC(K)^*$

Answer: I started by looking at the analogues to the situation in problem 4.

- $C(K)$ and $C(K)^*$ are like V and V^*
- W is like U
- $|00\rangle$ is like $|\phi\rangle$
- $|\eta\rangle$ is like $|\psi\rangle$ (except we don't have $C(K)|\psi\rangle = |\phi\rangle$)

This is not exact just how I started thinking about it. Next we have

$$\begin{aligned} C(K)WC(K)^*|\eta\rangle &= C(K)W\big(|01\rangle + |10\rangle - i|11\rangle\big) \\ &= C(K)\big(W|01\rangle + W|10\rangle - i|11\rangle\big) \\ &= W|01\rangle + W|10\rangle + |11\rangle \\ &= W|\eta\rangle = |\eta\rangle \end{aligned}$$

where I used that $|11\rangle = W|11\rangle$, my reasoning for this is that $W|\eta\rangle = W(|01\rangle + |10\rangle + |11\rangle) = |\eta\rangle$. So for the case for $|\eta\rangle$ being analogous to $|\psi\rangle$ is still there.

$$C(K)WC(K)^*|00\rangle = |00\rangle$$

by definition. Now, $C(K)|\eta\rangle \neq |\eta\rangle$ and we let G' be the set of all unitaries W' such that $W'|00\rangle = |00\rangle$. I think trying to think of this in analogy has made this harder than it should, where we stand right now, we have everything we need to apply problem 4: *We have a vector $|\eta\rangle$ which is an eigenvector for unitaries $W \in G$, and we have a unitary operator $C(K)$ such that $|\eta\rangle$ is not an eigenvector*

$$W'|00\rangle = |00\rangle \rightarrow C(K)WC(K)^* = W' \in G'$$

by the same argument as in the problem 4 solution (4), $C(K)WC(K)^*$ generates G' because for each $W \in G$ we have a $W' \in G'$. Again, following the argument made in problem 4 we can say that if we think of $C(K)GC(K)^*$ as G_1 and G as G_2 then it follows from problem 3 that $C(K)GC(K)^*$ and G generate G'.

b)

Show that the special unitary group on the $2 - \mathit{qubit}$ Hilbert space is generated by G' and $\big(H \otimes I\big)G'\big(H \otimes I\big)$

Answer: For this task, we try to apply the same routine as we did in the a). Here we use that $(H \otimes I)\,|00\rangle \neq |00\rangle$. And we also have that $|00\rangle$ is an eigenvector for all operators $W' \in G'$. Not completely sure what more there is to show, we obviously have the same situation as both problem 4 and a) so as the problem text asks us to do we apply problem 4 the second time and the it follows that $(H \otimes I)G'(H \otimes I)$ and G' generate $SU(\mathcal{H})$ where $\mathcal{H}$ is the two qubit Hilbert space we are dealing with. Putting it explicitly in the notation of problem 4 we have $G' = G_1$ and $(H \otimes I)G'(H \otimes I) = G_2'$, where the equals sign is just saying they are like.

c)

Using that any gate decomposes into one-qubit and two-qubit gates, prove Theorem (2).

Answer: Since I already showed that we can easily construct the $C(X)$-gate from the gates defined in theorem 2 it should follow from other proofs that $C(X)$ and single qubit gates are universal. But this does not do what the problem asks, we define $U \in SU(\mathcal{H})$ where $\mathcal{H}$ is the two qubit Hilbert space, then by b) we can generate it as a finite element from G' and $(H \otimes I)G'(H \otimes I)$. Looking at the lecture notes where we deal with a similar situation: the end of [10] and start of [11] we can compare to our situation here.

Theorem 4 ***(Universal Gate Sets)****A collection A of quantum gates on $\mathcal{H}^{\otimes n}$ closed under inversion is called a universal gate set if it generates a dense subgroup of $PU(\mathcal{H}^{\otimes n})$, that is, if U is unitary on $\mathcal{H}^{\otimes n}$, then for all $\epsilon > 0$ we can find $\phi \in \mathbb{R}$ and $U_1, ..., U_m \in A$ such that*

$$||U - e^{i\phi}U_1...U_m|| < \epsilon$$

So during this assignment we have shown that we are able to generate $SU(\mathcal{H})$ using G' in which $C(K)$ and $C(K)^*$ are members, and using $(H \otimes I) = (H \otimes I)^\dagger$ so therefore the results says that given $U \in U(\mathcal{H})$ and $\epsilon > 0$ we can find $\phi \in \mathbb{R}$ such that the above theorem holds. This is with the corollary from [11] which is also proven in the reference guarantees that 2 is true.

A Python Scripts

Below is the script used to convince myself of the equality discussed in 1

```
h = np.array([[1/np.sqrt(2), 1/np.sqrt(2)],
              [1/np.sqrt(2), -1/np.sqrt(2)]])
i = np.eye(2)
ccx = np.array([[1,0,0,0,0,0,0,0],
                [0,1,0,0,0,0,0,0],
                [0,0,1,0,0,0,0,0],
                [0,0,0,1,0,0,0,0],
                [0,0,0,0,1,0,0,0],
                [0,0,0,0,0,1,0,0],
                [0,0,0,0,0,0,0,1],
                [0,0,0,0,0,0,1,0]])
k = np.array([[1, 0],
              [0, 1j]])
k_inv = np.array([[1, 0],
                  [0, -1j]])

#matrix and tensor products
ii = np.kron(i, i)
iih = np.kron(ii, h)
one = np.matmul(iih, ccx)
two = np.matmul(one, iih)
iik_inv = np.kron(ii, k_inv)
three = np.matmul(two, iik_inv)hadde no
four = np.matmul(three, ccx)
iik = np.kron(ii, k)
five = np.matmul(four, iik)
six = np.matmul(five, ccx)
ck = np.array([[1,0,0,0],
               [0,1,0,0],
               [0,0,1,0],
               [0,0,0,1j]])
test = np.kron(ck, i)
epsilon = 1e-3
print(np.abs(six - test) < 1e-3)
```

The following script is the one which I used to explicitly check the circuit equalities I used to conclude that Theorem(1) follows from Theorem (2).

```
import numpy as np
import qiskit as qk
import matplotlib.pyplot as plt
from qiskit.circuit.library.standard_gates\
    import SGate, HGate, SdgGate, CXGate, CCXGate
from qiskit.quantum_info import Operator
from qiskit.visualization import circuit_drawer



def cnot(circuit, register, control, target,
        print_operator=True,
        visualize=True,
        check_approx_equality=True,
        plot_circuit=False):
```

```
    """
    circuit (Qiskit circuit object)
    register (Qiskit quantum register object)
    control (int) index in the register for the control qubit
    target (int) index in the register for the target qubit
    """
    h_gate = HGate()
    control_s_gate = SGate().control(1)

    circuit.append(h_gate, [register[target]])
    circuit.append(control_s_gate, [register[control], register[target]])
    circuit.append(control_s_gate, [register[control], register[target]])
    circuit.append(h_gate, [register[target]])

    if print_operator:
        # using reverse bits here to recover conventional notation
        print(Operator(circuit.reverse_bits()).data)

    if visualize:
        if visualize=='mpl':
            circuit.draw('mpl')
            plt.show()

        else:
            print(circuit)

    if check_approx_equality:
        print(Operator(circuit) == Operator(CXGate()))



def toffoli(circuit, register, control1, control2, target,
        print_operator=True,
        visualize=True,
        check_approx_equality=True,
        plot_circuit=False):
    """
    circuit (Qiskit circuit object)
    register (Qiskit quantum register object)
    control1 (int) index in the register for the first control qubit
    control2 (int) index in the register for the second control qubit
    target (int) index in the register for the target qubit
    """
    h_gate = HGate()
    control_s_gate = SGate().control(1)
    control_sdg_gate = SdgGate().control(1)


    circuit.append(h_gate, [register[target]])
    circuit.append(control_s_gate, [register[control2], register[target]])
    cnot(circuit, register, control1, control2,
            print_operator=False,
            visualize=False,
            check_approx_equality=False)
    circuit.append(control_sdg_gate, [register[control2], register[target]])
    cnot(circuit, register, control1, control2,
            print_operator=False,
            visualize=False,
            check_approx_equality=False)
    circuit.append(control_s_gate, [register[control1], register[target]])
    circuit.append(h_gate, [register[target]])

    if print_operator:
        print(Operator(circuit.reverse_bits()).data)

    if visualize:
        if visualize=='mpl':
            circuit.draw('mpl')
            plt.show()

        else:
            print(circuit)

    if check_approx_equality:
        print(Operator(circuit) == Operator(CCXGate()))



if __name__=='__main__':
    #qr1 = qk.QuantumRegister(2)
    #qc1 = qk.QuantumCircuit(qr1)
    #cnot(qc1, qr1, 0, 1, visualize='mpl')


    qr2 = qk.QuantumRegister(3)
    qc2 = qk.QuantumCircuit(qr2)
    toffoli(qc2, qr2, 0, 1, 2, visualize='mpl')
```

The following script was intended as a sanity check for my work on 5b) however it mainly just served to make me doubt the method I found. Turned out to be a interpretation mistake + a few sign errors which were remedied by using symbolab.com to check my multiplications.

```
import numpy as np

e_1 = np.array([
    [0],
    [1/np.sqrt(2)],
    [-1/np.sqrt(2)],
    [0]
])

e_2 = np.array([
    [0],
    [1/np.sqrt(2)],
    [0],
    [-1/np.sqrt(2)]
])

V = np.array([
    [1, 0, 0, 0],
    [0, 1/2 - 1j/2, 1/2, 1j/2],
    [0, 0, 1/2+1j/2, 1/2-1j/2],
    [0, 1/2+1j/2, -1j/2, 1/2]
])

V_prime = np.array([
    [1, 0, 0, 0],
    [0, 1/2 - 1j/2, 0, 1/2+1j/2],
    [0, 1/2, 1/2+1j/2, -1j/2],
    [0, 1j/2, 1/2-1j/2, 1/2]
])


def calc_restriction(A, u_1, u_2):
    """
    Calculates the restriction of A to the space spanned by u_1 and u_2
    V|_k = [<u_1, Vu_1>, <u_1, Vu_2>]
           [<u_2, Vu_1>, <u_2, Vu_2>]
    """

    A_u1 = np.matmul(A, u_1)
    A_u2 = np.matmul(A, u_2)

    result = np.array([
        [np.vdot(A_u1, u_1), np.vdot(A_u2, u_1)],
        [np.vdot(A_u1, u_2), np.vdot(A_u2, u_2)]
    ])

    return result


restriction_V = calc_restriction(V, e_1, e_2)
restriction_V_prime = calc_restriction(V_prime, e_1, e_2)

commutator1 = np.matmul(restriction_V, restriction_V_prime) - np.matmul(restriction_V_prime, restriction_V
commutator2 = np.matmul(restriction_V_prime, restriction_V) - np.matmul(restriction_V, restriction_V_prime

#print(commutator1, commutator2)

eigenvalues1 = np.linalg.eig(V)
eigenvalues2 = np.linalg.eig(V_prime)

print(f"eig1: {eigenvalues1[0]}\n")
print(f"eig2: {eigenvalues2[0]}\n")
print(f"given: {(1 + 1j*np.sqrt(15))/4}")
print(f"given: {(1 - 1j*np.sqrt(15))/4}")
```

B Solovay-Kitaev

Everything under here was cut from my task 3b) answer. I initially thought it we were supposed to use the Solovay-Kitaev theorem because I thought that coincides meant that we could create all the elements of $SU(n)$ from finite elements. However, more thinking and reading has led me to cut it away. Seemed a shame to throw away everything I found out about it though, so I'll just leave it here if the lecturer or anyone else wants to look at it in the future. *Quick Digression to the Solovay-Kitaev theorem:*

I found a paper by Nielsen and Dawson [5] which I felt explained in a bit more detail than the textbook also by Nielsen [1]. The Solovay-Kitaev theorem and corresponding algorithm speaks on the approximation of an arbitrary quantum gate U by using a finite sequence of gates $g_1, ..., g_m$ drawn from some finite set G. We call the set G an *instruction set*, and the process of finding a good approximation *compilation*. Now before moving onto the actual theorem we set up what we define what we mean by instruction set.

Definition: *an **instruction set** G for a d-dimensional qudit*[8] *is a finite set of quantum gates satisfying:*

I. *All gates $g \in G$ are in $SU(d)$, that is, they are unitary and have determinant 1*

II. *For each $g \in G$ the inverse operation $g^\dagger$ is also in G*

III. *G is a universal set for $SU(d)$, i.e. the group generated by G is dense in $SU(d)$. This means that given any quantum gate $U \in SU(d)$ and any accuracy $\epsilon > 0$ there exists a product $S \equiv g_1 ... g_m$ of gates from G which is an ϵ- approximation to U.*

The above definition is lifted straight from [5]. This definition is supplemented that by approximation we mean approximation in operator norm. In other words, a sequence of instructions generation a unitary operation S is said to be an ϵ-approximation to a quantum gate U if $d(U,S) \equiv ||U - S|| \equiv \sup_{||\psi||=1} ||\big(U - S\big)\psi|| < \epsilon$

Theorem 5 ***(Solovay-Kitaev)*** *Let G be an instruction set for $SU(d)$, and let a desired accuracy $\epsilon > 0$ be given. There is a constant c such that for any $U \in SU(d)$ there exists a finite sequence S of gates of length $\mathcal{O}(\log^c(1/\epsilon))$ and such that $d(U,S) < \epsilon$.*

The first thing I thought was that the above task would be solved using the Solovay-Kitaev theorem, but looking at the above definition of the instruction set, we see that one of the assumptions *3.* of the Solovay-Kitaev theorem is that the set G is dense in $SU(n)$. Also the Solovay-Kitaev theorem talks of making approximations to unitaries with a limited number of unitaries, whereas here we have an infinite number of unitaries to generate all of $SU(\mathcal{H})$.

I went through a lot of papers and web sites to try to figure out exactly how to show that G was dense in $SU(n)$ so that I could use the Solovay-Kitaev Theorem (5), but everywhere I found it just sort of glossed over it. In [1] is where I found the closest thing to something I could use:

A subset S of $SU(2)$ is said to be dense in $SU(2)$ if for any element $U \in SU(2)$ and $\epsilon > 0$ there is an element $s \in S$ such that $D(s,U) < \epsilon$. Suppose S and W are subsets of $SU(2)$, then S is said to form an $\epsilon - net$ for W where $\epsilon > 0$, if every point in W is within a distance ϵ of some point in S (p.618).

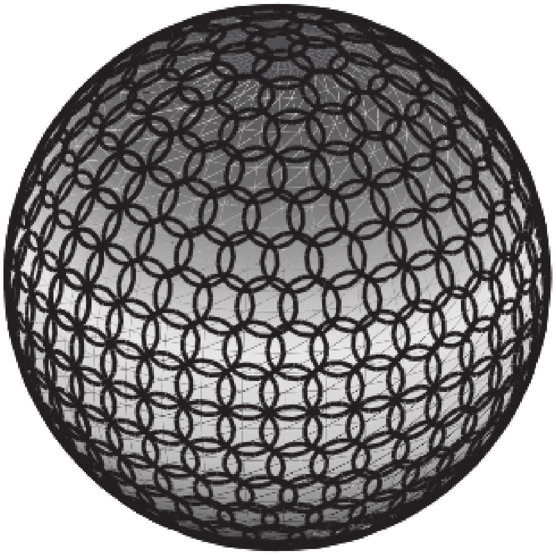


Figure 4: Visualization of an $\epsilon - net$. Gotten from page 199 in [1]

I did not really feel that this illuminated exactly how to show that G_i is dense in $SU(n)$. Other than saying that since we can pick a random unitary $V \in G_i$ and the action of V on ψ keeps ψ in the span of $|\phi_1\rangle, |\phi_2\rangle$. Further, since $|\phi_1\rangle, |\phi_2\rangle$ form a basis for $\mathcal{H}$ we should be able to write any $U \in SU(n)$ as a finite sequence, which we in turn can write as an element of G_i, and thus the above definition of density should hold. But I am not sure if this actually shows it or I am tricking myself somewhere along this explanation.

[8] Had not seen this word before I read this paper, from wikitionary: A *quantum dit*; the unit of quantum information described by a superposition of d states, where the number of states is an integer greater than two.

However, using statements from the 10-th [10], specifically the remark that any two randomly chosen 2×2 unitaries generate a dense subgroup of $PU(2)$ with probability 1 and the following lemma: *"Assume mutually orthogonal vectors $\vec{v}, \vec{w} \in S^2$, then every $U \in SU(2)$ can be written as $U = R_{\vec{v}}(\alpha)R_{\vec{w}}(\beta)R_{\vec{v}}(\gamma)$ for some $\alpha, \beta, \gamma \in \mathbb{R}$"*. Now in lecture 12 [12], we have the remark that *"In particular, A generates a dense subgroup of $SU(d)$"* which finally gets me convinced that our G coincides with $SU(\mathcal{H})$. Because for any $U \in SU(2)$ we can generate it as a finite sequence of elements in G by the Solovay-Kitaev Theorem (5). *End of digression.*

References

[1] Quantum Computation and Quantum Information 10th Anniversary Edition, Nielsen M.A., Chuang I.L, Cambridge University Press

[2] Qiskit Textbook `https://qiskit.org/textbook/preface.html`, Abraham Asfaw, Luciano Bello, Yael Ben-Haim, Mehdi Bozzo-Rey, Sergey Bravyi, Nicholas Bronn, Lauren Capelluto, Almudena Carrera Vazquez, Jack Ceroni, Richard Chen, Albert Frisch, Jay Gambetta, Shelly Garion, Leron Gil, Salvador De La Puente Gonzalez, Francis Harkins, Takashi Imamichi, Hwajung Kang, Amir h. Karamlou, Robert Loredo, David McKay, Antonio Mezzacapo, Zlatko Minev, Ramis Movassagh, Giacomo Nannicni, Paul Nation, Anna Phan, Marco Pistoia, Arthur Rattew, Joachim Schaefer, Javad Shabani, John Smolin, John Stenger, Kristan Temme, Madeleine Tod, Stephen Wood, James Wootton.

[3] Basic Concepts in Quantum Computation, February 1. 2008, Ekert A., Hayden P., Inamori H., Centre for Quantum Computation, University of Oxford, `https://arxiv.org/pdf/quant-ph/0011013.pdf`

[4] Equivalent Quantum Circuits, October 14. 2011, Garcia-Escartin J.C., Chamorro-Posada P., Department of Signal Theory and Telematics, Universidad de Valladolid, `https://arxiv.org/pdf/1110.2998.pdf`

[5] The Solovay-Kitaev Algorithm, February 1. 2008, Nielsen M.A., Dawson C.M., School of Physical Sciences, The University of Queensland, Brisbane, `https://arxiv.org/pdf/quant-ph/0505030.pdf`

[6] Universality of Single Quantum Gates, October 16. 2018, Bauer B., Levaillant C., Freedman M., Station Q Microsoft Research, Santa Barbara CA, Department of Mathematics, University of California Santa Barbara CA, `https://arxiv.org/pdf/1404.7822.pdf`

[7] The problem of Constructing Efficient Universal Sets of Quantum Gates, Liang Q., Thompson J., `https://lsa.umich.edu/content/dam/math-assets/math-document/reu-documents/REUsummer2015LiangThompsonDamelin.pdf`

[8] Efficient Decomposistion of Single-Qubit Gates into V Basis Circuits, 12 July. 2013 Bocharov A., Gurevich Y., Svore K.M., Quantum Architechtures and Computation Group, Microsoft Research, Research in Software Engineering Group Microsoft Research, Redmond Washington, `https://journals-aps-org.ezproxy.uio.no/pra/pdf/10.1103/PhysRevA.88.012313`

[9] Commuting and Non-Commuting Operators `http://web.mnstate.edu/marasing/CHEM460/Handouts/Chapters/9%20Commuting%20and%20Non%20Commuting%20Operators.pdf`

[10] Lecture 10 Mat3420, February 12. 2021, Neshveyev S. `https://www.uio.no/studier/emner/matnat/math/MAT3420/v21/filer/lecture10.pdf`

[11] Lecture 11 Mat3420, February 15. 2021, Neshveyev S. `https://www.uio.no/studier/emner/matnat/math/MAT3420/v21/filer/lecture11.pdf`

[12] Lecture 12 Mat3420, February 19. 2021, Neshveyev S. `https://www.uio.no/studier/emner/matnat/math/MAT3420/v21/filer/lecture12.pdf`

[13] Properties of the tensor product. `https://www.uio.no/studier/emner/matnat/math/nedlagte-emner/MAT-INF2360/v12/tensortheory.pdf`

[14] `https://www.sciencedirect.com/topics/mathematics/hadamard-gate`

[15] `https://qiskit.org/documentation/tutorials/circuits/3_summary_of_quantum_operations.html`

[16] https://en.wikipedia.org/wiki/Cyclotomic_polynomial

[17] Explanation of restriction of a matrix to a subspace, https://math.stackexchange.com/questions/2532488/matrices-restricted-to-a-subspace/2532899

[18] https://mathworld.wolfram.com/HermitianInnerProduct.html

[19] Appendix on SU(N) https://onlinelibrary.wiley.com/doi/pdf/10.1002/9783527648887.app8