5: Special Matrices

Motivation:

- what are quantum algorithms other than applying a unitary transformation (matrix) to a unit vector?
- This is why we need to talk about special families of matrices.
- These families are important and have applications other than quantum algorithms.

5.1 Hadamard Matrices

Aside: Hadamard Matrices were discovered by Joeseph Sylvester, not Jacques Hadamard. (1867)

Def: The Hadamard matrix H_N of order N is defined recursively by $H_2 = H$ & $N \geq 4$

$$H_N = H_{N/2} \otimes H = \frac{1}{\sqrt{2}} \begin{bmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & -H_{N/2} \end{bmatrix}$$

Notation: if it is based on n not N we denote $H^{\otimes n}$

$H_1 = [1]$

$$H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Ex: H_4?

$$H_4 = H_{4/2} \otimes H = H_2 \otimes H_2$$

$$= \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

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

Sometimes we want the direct way to solve, not a recursive definition.

Lemma: For any row r and column c,

$$H_N[r, c] = (-1)^{r \cdot c}$$

where $r \cdot c$ is inner product of r & c treated as Boolean strings

Relating to the example, we see the first -1 is in the [1,1] place.

$$H_2[1,1] = (-1)^{1 \cdot 1} = (-1)^{(0,1)\cdot(0,1)} = (-1)^{0+1} = (-1)^1$$

Proof: we prove inductively

Above example is our base case.
Next we do inductive hypothesis
"$n \to n+1$ or $N \Rightarrow 2N$

$$H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix}$$

The first digit in $x \cdot y$ is now just the one bit case so we only see change in sign in the [1,1] block and $1 \cdot 1 \oplus H_N$

$$(-1)^{1+H_N} \Rightarrow (-1)\ \mathrm{sign}(H_N)$$

I hope this next step (which sort of looks like pseudocode) makes sense for actually inputting into our algorithms.

Corollary: For any vector $\bar{a}$, the vector $\bar{b} = H_N \bar{a}$:=

$$\bar{b}(x) = \frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} (-1)^{x \cdot t} \bar{a}(t)$$

5.2 Fourier Matrices

Let $\omega = e^{2\pi i/N}$

Def: The Fourier Matrix F_N of order N

$$\frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & & \omega^{N-2} \\ 1 & \omega^3 & \omega^6 & \omega^9 & & \omega^{N-2} \\ \vdots & & & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{N-2} & \omega^{N-3} & \cdots & \omega \end{bmatrix}$$

That is $F_N[i,j] = \omega^{ij \bmod N}$ divided by $\sqrt{N}$

Corollary: For any vector $\bar{a}$, the vector $\bar{b} = F_N \bar{a}$:=

$$\bar{b}(x) = \frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} \omega^{xt} \bar{a}(t)$$

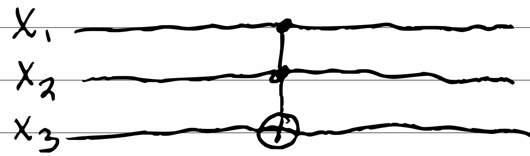
This should look familiar. The subtle distinctions between ± 1 vs ω and inner product vs. multiplication is the difference which separates Shor's and Simon's algorithm.

5.3 Reversible Computation & Permutation Matrics

Definition: The Toffoli Gate

It is a universal reversible Logic gate. which means any classical reversible circuit can be constructed by a Toffoli Gate

It has 3 inputs and is also called the controlled-control-not, or CCNOT



It might help to see the truth table and permutation matrix

Input			Output		
x_1	x_2	x_3	x_1	x_2	$x_3 \oplus (x_1 \wedge x_2)$
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

$$TOF = \begin{bmatrix} I_6 & 0 \\ 0 & \begin{matrix} 0 & 1 \\ 1 & 0 \end{matrix} \end{bmatrix}$$

$$TOF(x_1, x_2, x_3) = (x_1, x_2, x_3 \oplus (x_1 \wedge x_2))$$

$$Not(a) = TOF(1, 1, a)$$
$$AND(a, b) = TOF(a, b, 0)$$

<u>Theorem</u>: All classically feasible Boolean functions f have feasible quantum computations in the form of P_f.

Proof: Recall AND & NOT are universal logic gates. Let C be a circuit computing $f(x_1, \ldots, x_n)$ using r-many NOT and s-many ANDs. As you can see above we have a way to encode NOT in a 2×2 matrix.

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

We now turn our attention to s-many AND gates, using $AND(a,b) = TOF(a,b,0)$ but we might need duplicate copies on each wire coming out of the gate.

We add ancilla z for each wire w coming from C then using TOF, we get $z \oplus (a \wedge b)$ with $z = |0\rangle$. TOF gates have controls that if are the same don't change each other

So the overhead is bounded by w wires in circuit C, which is polynomial, and all added ancilla bits obey the convention of being initialized to 0.

$\blacksquare$

Take away: Permutation matrix is feasible if it is induced by classical feasible function on quantum coordinates.

5.4 Feasible Diagonal Matrices

Recall: Any diagonal matrix whose entries are ± 1 is unitary.

$$U = \begin{pmatrix} \pm 1 & & \\ & \ddots & \\ & & \pm 1 \end{pmatrix} \quad U^T = \begin{pmatrix} \pm 1 & & \\ & \ddots & \\ & & \pm 1 \end{pmatrix}$$

$$UU^T = I$$

But is U feasible?

If size of U is small $\Rightarrow$ basic $\Rightarrow$ feasible

$S \subseteq \{0,1\}^N$ for $N \times N$ matrix

$$U_S[x,x] = \begin{cases} -1 & x \in S \\ 1 & \text{otherwise} \end{cases}$$

But this is still doubly exponentially large, which means most aren't feasible

Def: When S is a set of arguments that $f(S)=1$ we write U_f which is called the <u>Grover Oracle for f.</u>

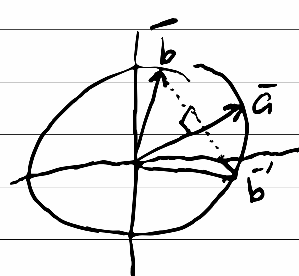
Theorem: If f is a feasible boolean function, then the Grover Oracle, U_f is feasible.

Proof idea: Apply Hadamard matrix to e_x for $F(x,y) = (x, (y \oplus f(x))$ then we have flipped everything, apply another H and NOT we can flip everything back

$$e_x \longmapsto (-1)^{f(x)} e_x$$

5.5 Reflections

Def. given any unit vector $\bar{a}$ we can create a unitary operator $Ref_{\bar{a}}$, which reflects another unit vector $\bar{b}$ around $\bar{a}$

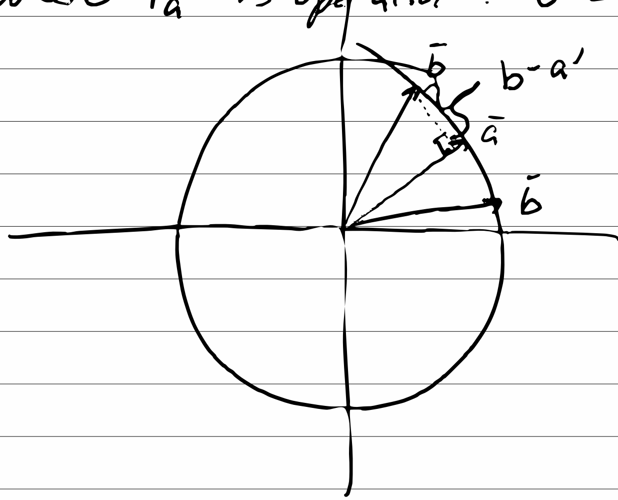


$b \to b'$ preserves the unit sphere and is its own inverse. $\Rightarrow$ Unitary

The point on $\bar{a}$ is the projection of $\bar{b}$ onto $\bar{a}$ which is $a' = a\langle a, b\rangle$

$$b' = b - 2(b - a\langle a,b\rangle)$$
$$= (2P_a - I)\, b$$

where P_a is operation: $\forall b \;\; P_a b = a\langle a, b\rangle$



Ex: Let $\bar{a}$ be unit vector with entries $1/\sqrt{N}$, we call j. Then Projector is matrix whose entries are all $1/N$, which we call J

$$\text{Ref} = V = 2J - I = \begin{bmatrix} \frac{2}{N}-1 & \frac{2}{N} & \cdots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N}-1 & \cdots & \frac{2}{N} \\ \vdots & \vdots & \ddots & \\ \frac{2}{N} & \frac{2}{N} & & \frac{2}{N}-1 \end{bmatrix}$$

This is feasible! But are there other feasible reflection operations?

Let $\bar{a}$ be a characteristic vector of nonempty set S,

$$\bar{a}(x) = \begin{cases} 1/\sqrt{|S|} & \text{if } x \in S \\ 0 & \text{otherwise} \end{cases}$$

Let's apply Ref_a to $\bar{b}$ that $e = \bar{b}(x)$ for $x \in S$ are equal. Let $k = |S|$

then $\langle a, b \rangle = k \, e/\sqrt{k} = e\sqrt{k}$ and

$a' = P_a \bar{b}$ we get

$$a'(x) = \begin{cases} e & \text{if } x \in S \\ 0 \end{cases}$$

$$b' = 2a' - b$$
$$= \begin{cases} b(x) & \text{if } x \in S \\ -b(x) & \text{else} \end{cases}$$

because $b'(x) = 2e - b(x) = 2e - e = e = b(x)$ and $x \notin S$ $b'(x) = -b(x)$

This is the Grover oracle of compliment of S which negation of feasible boolean function is feasible.

Theorem: $\forall$ feasible Boolean functions f, provided we restrict to linear subspace of argument vectors whose entries indexed by "true set" S_f of f are equal, reflection about the characteristic vector $\bar{S}_f$ is feasible quantum operation.

Proof idea: set of argument vectors form a linear subspace & contain $\bar{j}$ or the start vector. Reflecting by $\bar{a}$ or $\bar{b}$ applied to vectors in the subspace spaned by $\bar{a}, \bar{b}$ stay in the subspace.