Chapter 5

Quantum Fourier Transform

Quantum Fourier Transform is a *quantum* implementation of the discrete Fourier transform. You might be familiar with the discrete Fourier Transform or Fourier Analysis from the context of signal processing, linear algebra, or one of its many other applications. In short, Fourier Analysis is a tool to describe the internal frequencies of a function.

Here we will present a quantum algorithm for computing the discrete Fourier transform which is exponentially faster than the famous Fast Fourier Transform of classical computers. However, this algorithm provides an excellent example of the tension between exponentially faster quantum algorithms and the problems of measurement. While we can carry out the QFT algorithm to transform the n qubit state vector $|\alpha\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \cdots + \alpha_n |n\rangle$ to its Fourier transform $|\beta\rangle = \beta_0 |0\rangle + \beta_1 |1\rangle + \cdots + \beta_n |n\rangle$, a measurement on $|\beta\rangle$ will only return one of its n components, and we are not able to recover all the information of the Fourier transform. For this reason, we describe this algorithm as quantum Fourier *sampling*.

The Quantum Fourier Transform is a generalization of the Hadamard transform. It is very similar, with the exception that QFT introduces phase. The specific kinds of phases introduced are what we call primitive roots of unity, ω. Before defining the Fourier Transform, we will take a quick look at these primitive roots of unity.

Recall that in the complex plane, there exist n solutions to the equation $z^n = 1$. For example if $n = 2$, z could be 1 or -1. If $n = 4$, z could be $1, i, -1$, or $-i$. You can easily check that these roots can be written as powers of $\omega = e^{2\pi i/n}$. This number ω is called the primitive nth root of unity. In Figure 1, ω is drawn along with the other complex roots of unity for n=5.

In this figure, we see that ω lies on the unit circle so $\mid \omega \mid = 1$, and the line from the origin to ω makes the angle $\phi = 2\pi/M$ with the real line. If we square ω, we double the angle. Furthermore, if we raise ω to the jth power, ω^j has phase angle $\phi = 2j\pi/M$ and is still an Mth root of unity.

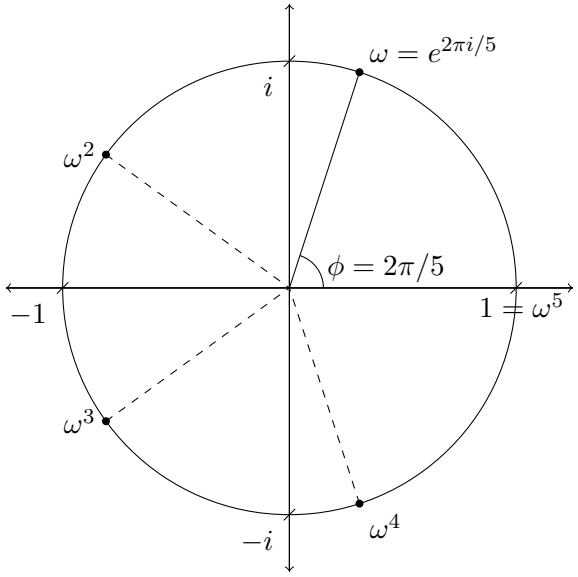


Figure 5.1: The 5 complex 5th roots of 1.

Now we can move in to the Fourier Transform itself. The discrete Fourier transform is defined by

$$QFT_M = \frac{1}{\sqrt{M}} \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{M-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2M-2} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3M-3} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{M-1} & \omega^{2M-2} & \omega^{3M-3} & \cdots & \omega^{(M-1)(M-1)} \end{pmatrix}$$

Another way of writing this is to say that the jkth entry of QFT_M is ω^{jk}. The transform takes the vector $\begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{n-1} \end{pmatrix}$ to its Fourier transform $\begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{n-1} \end{pmatrix}$ as specified by the above matrix.

Examples

Ex. 1

Lets take a look at QFT_2. Because $M = 2$, $\omega = e^{\pi i} = -1$ Therefore we have

$$QFT_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & \omega \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

As you can see, QFT_2 is simply equal to $H^{\otimes 2}$.

How about QFT_4? The primitive 4th root of unity is i, so that

$$QFT_4 = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}$$

Ex. 2

Find the quantum Fourier transform for $M = 4$ of the functions $|f\rangle = \frac{1}{2}(|0\rangle+|1\rangle+|2\rangle+|3\rangle) = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$;

$|g\rangle = |0\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, and $|h\rangle = |1\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$.

The corresponding Fourier transforms are given by:

1. $|f\rangle$:
$$\left|\hat{f}\right\rangle = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$
2. $|g\rangle$:
$$|\hat{g}\rangle = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$
3. $|h\rangle$:
$$\left|\hat{h}\right\rangle = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 \\ i \\ -1 \\ -i \end{pmatrix}$$

Lets do a bit of analysis of these examples. In example 1, you might notice that the columns of QFT_4 are orthogonal. For example, the inner product of the first column with the second column is $\frac{1}{2}[(1*1)+(1*-i)+(1*-1)+(1*i)] = \frac{1}{2}(1-i-1+i) = 0$. You should also notice that, by design, the columns of QFT_4 have magnitude 1. Thus QFT_4 is unitary.

In example 2 you should notice is that the vectors like $|f\rangle$ that had a lot of zeros (large spread) had Fourier transforms with few zeros (narrow spread), and vice-versa.

Finally, in example 2 notice how the only difference between the Fourier transforms of $|g\rangle$ and $|h\rangle$ is a difference of relative phase shifts.

We would like to be able to make some statements to solidify these ideas about Fourier transforms, so lets prove them.

Properties of QFT

Studying the properties of a mathematical object gives us insight into the way it works. These properties will not only be important to our use of the Fourier transform later on, but they also provide a foundation of how to understand the discrete Fourier transform.

1. QFT_M is unitary.

   It is well known that an operator is unitary if its columns are orthonormal. Denote the ith and jth columns of QFT_M as F_i and F_j. Then $F_i = \frac{1}{\sqrt{M}} \begin{pmatrix} 1 \\ \omega^{i*1} \\ \vdots \\ \omega^{i*(M-1)} \end{pmatrix}$ and $F_j = \frac{1}{\sqrt{M}} \begin{pmatrix} 1 \\ \omega^{i*j} \\ \vdots \\ \omega^{j*(M-1)} \end{pmatrix}$.

   Thus

   $$\langle F_i | F_j \rangle = \frac{1}{M} \sum_{n=0}^{M-1} \omega^{ni} \overline{\omega^{nj}} = \frac{1}{M} \sum_{n=0}^{M-1} (\omega^{i-j})^n$$

   From here it is easy to see that if $i = j$, $\langle F_i | F_j \rangle = 1$.

   For the case $i \neq j$, we will notice that $\frac{1}{M} \sum_{n=0}^{M-1} (\omega^{i-j})^n$ is a geometric series, and expand the sum. Thus

   $$\frac{1}{M} \sum_{n=0}^{M-1} (\omega^{i-j})^n = \frac{1}{M} \frac{\omega^{M(i-j)} - 1}{\omega^{i-j} - 1} = \frac{1}{M} \frac{1 - 1}{\omega^{i-j} - 1} = 0$$

   where $\omega^{M(i-j)} = 1$ because ω is an Mth root of unity.

   Because the Fourier transform is a unitary operator, we can implement it in a quantum circuit. Thus if $N = 2^n$, we can apply the Fourier transform QFT_N to a n-qubit system.

2. *Linear Shift*

   This, property noted in the above examples, states that linear shifts of state-vectors cause relative phase shifts of their Fourier transform. This is expressed mathematically by saying if $|f(x)\rangle$, $x \in \mathbf{Z}_M$, has Fourier transform $\left|\hat{f}(x)\right\rangle$, then $|f(x+j)\rangle$ has Fourier transform $\left|\hat{f}(x)\right\rangle \phi_j$, where ϕ_j is a phase shift and $\phi_j = e^{\frac{2\pi}{M} xj}$. Furthermore, because QFT_M is unitary and $QFT_M QFT_M^\dagger = \mathbb{I}$, the converse is true. A linear phase shift on $|f\rangle$ produces a linear shift in $\left|\hat{f}\right\rangle$.

So if $QFT_N \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{N-1} \end{pmatrix} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{N-1} \end{pmatrix}$, then $QFT_N \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_0 \end{pmatrix} = \begin{pmatrix} \beta_0 \\ \omega\beta_1 \\ \vdots \\ \omega^{N-1}\beta_{N-1} \end{pmatrix}$ and $QFT_N \begin{pmatrix} \alpha_0 \\ \omega\alpha_1 \\ \vdots \\ \omega^{N-1}\alpha_{N-1} \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_0 \end{pmatrix}$.

If you have never seen this property before, it should be shocking. We will not offer a proof of this in general here, but below is an example with $N = 4$.

Example:

Let $|\Theta\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix}$ and $|\Phi\rangle = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_0 \end{pmatrix}$. Then

$$\left|\hat{\Theta}\right\rangle = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}\begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} \alpha_0 + \alpha_1 + \alpha_2 + \alpha_3 \\ \alpha_0 + i\alpha_1 - \alpha_2 - i\alpha_3 \\ \alpha_0 - \alpha_1 + \alpha_2 - \alpha_3 \\ \alpha_0 - i\alpha_1 - \alpha_2 + i\alpha_3 \end{pmatrix} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix}$$

$$\left|\hat{\Phi}\right\rangle = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_0 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} \alpha_0 + \alpha_1 + \alpha_2 + \alpha_3 \\ -i\alpha_0 + \alpha_1 + i\alpha_2 - \alpha_3 \\ -\alpha_0 + \alpha_1 - \alpha_2 + \alpha_3 \\ i\alpha_0 + \alpha_1 - i\alpha_2 - \alpha_3 \end{pmatrix} = \begin{pmatrix} \beta_0 \\ -i\beta_1 \\ -\beta_2 \\ i\beta_3 \end{pmatrix}$$

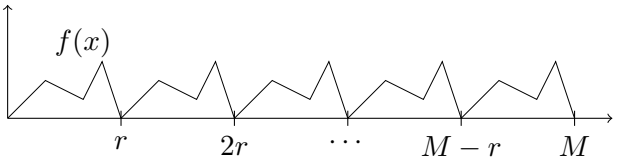
The important point here is that the only difference between $\left|\hat{\Theta}\right\rangle$ and $\left|\hat{\Phi}\right\rangle$ is a relative phase shift. But does this matter?

If we are going to measure a state then the phases don't matter at all, because if the phase is ϕ, then $\langle\phi|\phi\rangle = 1$. Therefore the phase of a given state does not effect the probability of measuring that state. However, there is a way we can gather information about the pahses.

We won't be able to tell by measuruing the difference between $\frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$ and $\frac{1}{2}\begin{pmatrix} 1 \\ i \\ -1 \\ -i \end{pmatrix}$ by making a measurment. However, if we apply QFT, we see that $QFT_4\frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$ and $QFT_4\frac{1}{2}\begin{pmatrix} 1 \\ i \\ -1 \\ -i \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$. Thus, **measuring the Fourier Transform of the states will reveal the relative phases**.

3. *Period/Wavelength Relationship*

Suppose f is periodic with period r, for example



Then $\hat{f}$ (the Fourier transform of f) is supported only on multiples of M/r. Thus, $\hat{f}(x) = 0$ unless $x = kM/r$.

If r is the period of f, we can think of M/r as the wavelength of f. If you already have intuition for Fourier transform this should come as no surprise. In general, the wider the range of a function, the sharper the range in the Fourier domain; and vise versa. In example, the fourier transform of a delta function is an even spread, while the transform of an even spread is a delta function.

We will prove this in a special case, where $f(j) = \begin{cases} \sqrt{\frac{r}{M}} & \text{if } j = 0 \pmod r \\ 0 & \text{otherwise.} \end{cases}$ While this is a very special case, it is actually the only case that we will need to develop Shor's algorithm. Furthermore this property *can* be proved in general.

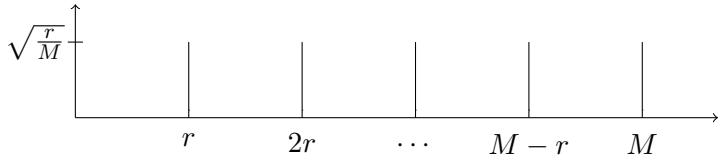


Figure 5.2: $f(j)$ in special case proof.

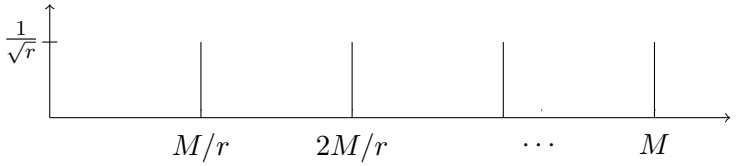


Figure 5.3: $\hat{f}(j)$

Because this function is relatively simple, we can prove the desired relationship by brute force. Suppose $|\alpha\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_N \end{pmatrix}$ has Fourier transform $|\beta\rangle = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_N \end{pmatrix}$, then the jth component of its Fourier transform is given by

$$\beta_j = \frac{1}{\sqrt{M}} \sum_{i=0}^{N-1} \alpha_i \omega^{ij} \tag{5.1}$$

This expression comes from matrix multiplication, you should take a look at the definition of F_M above to verify this if it looks unfamiliar. In our special case, $f(j) = \begin{cases} \sqrt{\frac{r}{M}} & \text{if } j = 0 \pmod r \\ 0 & \text{otherwise.} \end{cases}$. Using (**??**) we can calculate its Fourier transform

$$\hat{f}(j) = \frac{\sqrt{r}}{M} \sum_{i=0}^{\frac{M}{r}-1} \omega^{rij} \tag{5.2}$$

We have seen sums like this before. It is just a geometric series, and it is not too difficult to compute.

$$\sum_{i=0}^{\frac{M}{r}-1} \omega^{rij} = \frac{\omega^{Mj} - 1}{\omega^{rj} - 1}$$

But $\omega^{Mj} = 1$ (recall $\omega^M = 1$), so the numerator is always equal to 0. The only time the denominator can equal zero is when $rj = kM$ for some integer k. In this case, the numerator and the denominator are equivalently 0, so we must compute the limit using l'Hospitals rule.

$$\begin{aligned} \lim_{j \to k\frac{M}{r}} \frac{\omega^{Mj} - 1}{\omega^{rj} - 1} &= \lim_{j \to k\frac{M}{r}} \frac{M}{r} \frac{\omega^{Mj-1}}{\omega^{rj-1}} \\ &= \lim_{j \to k\frac{M}{r}} \frac{M}{r} \frac{\omega^{Mj}}{\omega^{rj}} \frac{\omega^{-1}}{\omega^{-1}} \\ &= \frac{M}{r} \end{aligned}$$

When we plug this result back into (**??**), the outcome is the desired result.

$$\hat{f}(j) = \begin{cases} \frac{1}{\sqrt{r}} & \text{if } j = 0 \pmod{\frac{M}{r}} \\ 0 & \text{otherwise.} \end{cases}$$

Notice that the normalization factor makes good sense.

To get a look at how we could prove this property in general, imagine periodic functions (in r) of this type as a basis for any periodic function. Allow the possibility of relative phase shifts, and you can prove this property in general.

Classical Fast Fourier Transform

The FFT was a major breakthrough for classical computers. Because the Fourier transform is an $M * M$ matrix, straightforward multiplication by F_M would take $O(M^2)$ steps to carry out, because multiplication of f on each row takes M multiplications. The FFT reduced this to $O(M \log M)$ steps. The FFT is incredibly important in signal processing that essentially all of your electronics rely on it. Without the FFT, modern electronics would have far fewer capabilities and would be much slower than they are today.

The FFT requires only that $M = 2^m$ for some integer m, but this is a relatively easy requirement because the computer can simply choose their domain.

The *fast* Fourier transform uses the symmetry of the Fourier transform to reduce the computation time. Simply put, we rewrite the Fourier transform of size M as two Fourier transforms of size $M/2$ - the odd and the even terms. We then repeat this over and over again to exponentially reduce the time. To see how this works in detail, we turn to the matrix of the Fourier transform. While we go through this, it might be helpful to have QFT_8 in front of you to take a look at. Note that the exponents have been written modulo 8, since $\omega^8 = 1$. Take a look at some of the symmetries and patterns.

$$\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\
1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\
1 & \omega^3 & \omega^6 & \omega & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\
1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 \\
1 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega & \omega^6 & \omega^3 \\
1 & \omega^6 & \omega^4 & \omega^2 & 1 & \omega^6 & \omega^4 & \omega^2 \\
1 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega^1
\end{pmatrix}$$

Notice how row j is very similar to row $j + 4$. Also, notice how column j is very similar to column $j + 4$. Motivated by this, we are going to split the Fourier transform up into its even and odd columns.

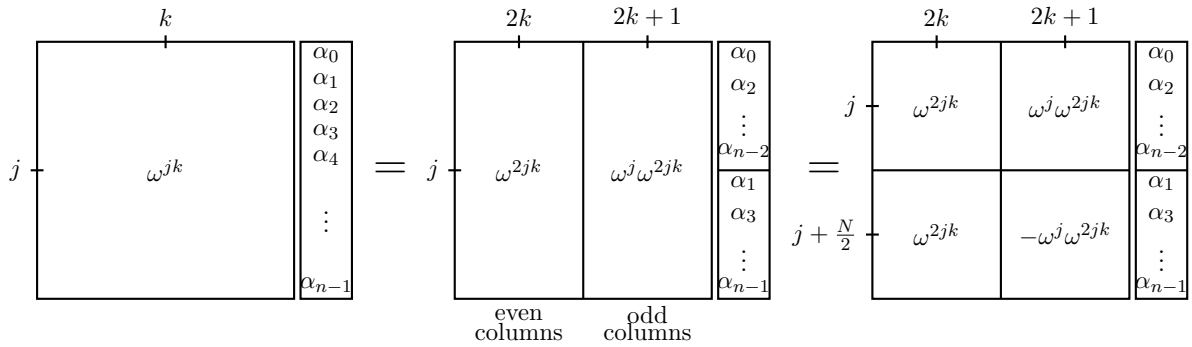


Figure 5.4: Breaking the Fourier transform up by symmetries.

In the first frame, we have represented the whole Fourier transform matrix by describing the jth row and kth column: ω^{jk}. In the next frame, we separate the odd and even columns, and similarly separate the vector that is to be transformed. You should convince yourself that the first equality really is an equality either by carrying out the matrix multiplication, or by noticing that all we did was re-organize the basis vectors. In the third frame, we add a little symmetry by noticing that $\omega^{j+N/2} = -\omega^j$ (since $\omega^{N/2} = -1$).

Notice that both the odd side and even side contain the term ω^{2jk}. But if ω is the primitive Nth root of unity, then ω^2 is the primitive $N/2$nd root of unity. Therefore, the matrices whose j,kth entry is ω^{2jk} are really just $QFT_{N/2}$! Now we can write QFT_N as in Figure 5.

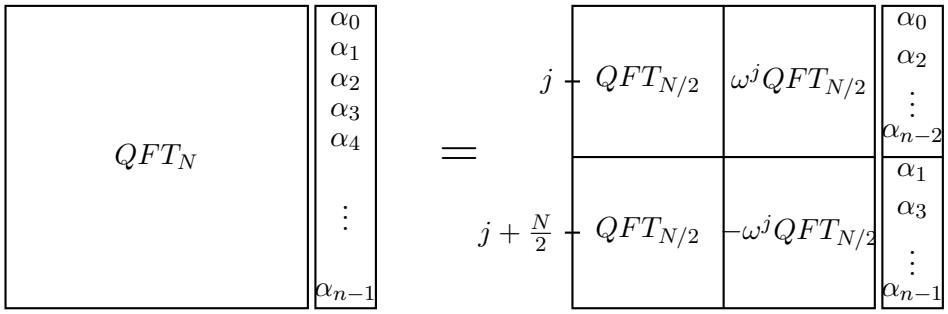


Figure 5.5: Simplified Fourier transform

Now suppose we are calculating the Fourier transform of the function $f(x)$. We can write the above manipulations as an equation that computes the jth term $\hat{f}(j)$.

$$\hat{f}(j) = \left(F_{M/2}\overrightarrow{f_{\text{even}}}\right)(j) + \omega^j \left(F_{M/2}\overrightarrow{f_{\text{odd}}}\right)(j)$$

This turns our calculation of QFT_N into two applications of $QFT_{N/2}$. We can turn this into four applications of $QFT_{N/4}$, and so forth. As long as $N = 2^n$ for some n, we can break down our calculation of QFT_N into N calculations of $QFT_1 = 1$. This greatly simplifies our calculation.

Quantum Fourier Transform w/ quantum gates

The strength of the the FFT is that we are able to use the symmetry of the discrete Fourier transform to our advantage. The circuit application of QFT uses the same principle, but because of the power of superposition QFT is even faster.

The QFT is motivated by the FFT so we will follow the same steps, but because this is a quantum algorithm the implementation of the steps will be different. That is, we first take the Fourier transform of the odd and even parts, then multiply the odd terms by the phase ω^j.

The first step is to separate the odd and even terms. But in a quantum algorithm, the odd and even terms are already together in superposition: the odd terms are those whose least significant bit is 1, and the even with 0. Therefore, we can apply $QFT_{M/2}$ to both the odd and even terms

together. We do this by applying $QFT_{M/2}$ to the m-1 most significant bits, then recombining the odd and even appropriately by applying the Hadamard to the least significant bit.

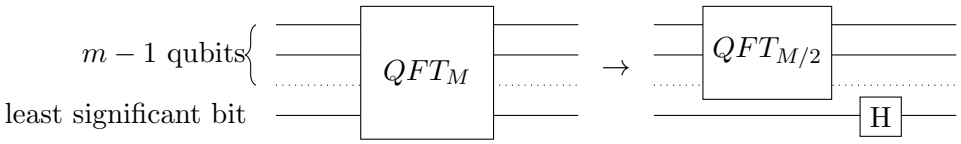


Figure 5.6: $QFT_{M/2}$ and a Hadamard gate correspond to $FFT_M/2$ on the odd and even terms

Now to carry out the phase multiplication, we need to multiply each odd term j by the phase ω^j. But remember, an odd number in binary ends with a 1 while an even ends with a 0. Thus we can use the *controlled phase shift*, where the least significant bit is the control, to multiply only the odd terms by the phase without doing anything to the even terms. Recall that the controlled phase shift is similar to the CNOT gate in that it only applies a phase to the target if the control bit is one.

The phase associated with each controlled phase shift should be equal to ω^j where j is associated to the kth bit by $j = 2^k$.

Thus, apply the controlled phase shift to each of the first $m-1$ qubits, with the least significant bit as the control. With the controlled phase shift and the Hadamard transform, QFT_M has been reduced to $QFT_M/2$.

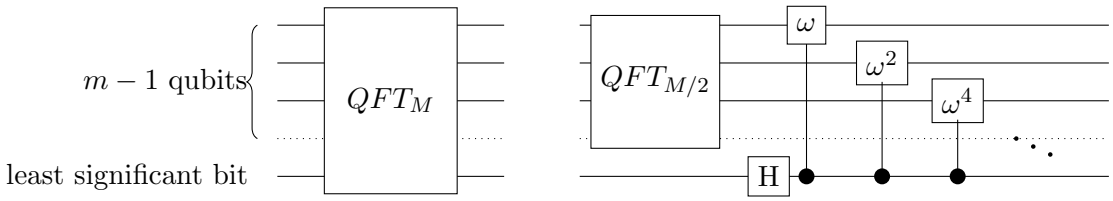


Figure 5.7: QFT_M is reduced to $QFT_{M/2}$ and M additional gates

Example

Lets construct QFT_3. Following the algorithm, we will trun QFT_3 into QFT_2 and a few quantum gates. Then continuing on this way we turn QFT_2 into QFT_1 (which is just a Hadamard gate) and another few gates. Controlled phase gates will be represented by R_ϕ.

then run through another iteration to get rid of QFT_2

You should now be able to visualize the circuit for QFT on more qubits easily. Furthermore, you can see that the number of gates necessary to carry out QFT_M it takes exactly $\sum_{i=1}^{\log M} i = \log M(\log M + 1)/2 = O(\log^2 M)$.

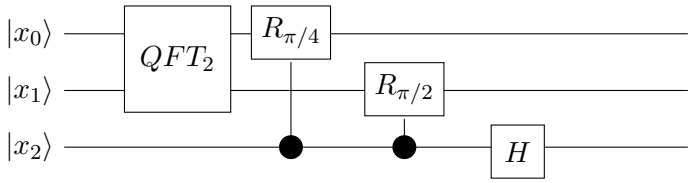


Figure 5.8: First Iteration

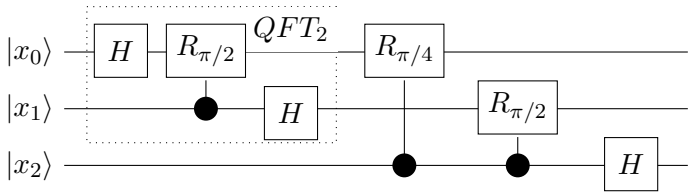


Figure 5.9: Second Iteration. Recall that $H = QFT_1$