Often it's useful to study functions on spaces by examining how they transform under various transformations of the spaces. This general idea gives rise to many interesting objects. For instance, if the space is the real line, we can ask how functions transform when the real line is translated. This gives rise to the Fourier transform. For functions on a sphere, we can ask about transformation under rotations. This is the origin of the spherical harmonics. On a finite cyclic group, we can ask how functions transform when the group is left- or right- multiplied by a group element. Thinking of the group elements as marking positions in space, we can think of this group action as a shift, and think about the resulting discrete Fourier transform as a discrete analogue of the standard one.

In this post, I'll discuss briefly a generalization of the discrete Fourier transform to arbitrary finite groups. The unifying thread in all the examples above is that the functions can be organized into irreducible representations of the group of transformations being applied to the space on which they're defined. Those transformations here will be left- and right-multiplication by group elements.

1. Definition of the QFT

Let G be a finite group of size $|G|$. Since G is finite, we don't have to be particularly careful about what we mean by "functions on G", but because it matches up with the infinite case and because we'll want an inner-product structure, we'll specify the space $\ell^2(G)$ of square-summable complex-valued functions. This space has a basis $\{\delta_g\}_{g\in G}$ of functions defined by $\delta_g(g) = 1$, $\delta_g(h) = 0$ for all $h \in G$, $h \neq g$. Following QI convention, I'll denote this $|g\rangle$.

Let Λ be a maximal set of unitarily-inequivalent unitary representations of G. We know that

$$\sum_{\lambda\in\Lambda} d_\lambda^2 = |G|$$

where d_λ is the dimension of the representation $\lambda \in \Lambda$. We can therefore choose another orthonormal basis for $\ell^2(G)$ with basis functions labeled by elements $|\lambda; ij\rangle$ for $\lambda \in \Lambda$ and $i, j = 1, 2, \ldots, d_\lambda$. Now we can define the Quantum Fourier Transform (QFT) and its inverse as transformations from the basis $|g\rangle$ to the basis $|\lambda; ij\rangle$ and back:

$$U_{QFT} = \sum_{\lambda\in\Lambda} \sum_{i,j=1}^{d_\lambda} \sum_{g\in G} \sqrt{\frac{d_\lambda}{|G|}} \left[R_\lambda(g)\right]_{ij} |\lambda; ij\rangle \langle g|$$

$$U_{QFT}^\dagger = \sum_{\lambda\in\Lambda} \sum_{i,j=1}^{d_\lambda} \sum_{g\in G} \sqrt{\frac{d_\lambda}{|G|}} \left[R_\lambda(g)\right]_{ij}^* |g\rangle \langle \lambda; ij|$$

There's nothing particularly quantum about this, other than the use of Dirac notation. However, an efficient implementation of the QFT is at the heart of Shor's factorization algorithm. Somewhat tragically, the algorithm only uses the QFT for cyclic groups, so its full generality isn't featured in that setting.

The QFT can be thought of as a map from complex functions on G to complex matrix-valued functions on the inequivalent unitary irreducible representations of G, where the value of the function at λ is a $d_\lambda \times d_\lambda$ matrix. A convenient way to visualize the Quantum Fourier Transform is via the following table. The row next to a group element g is the QFT of $|g\rangle$, and the complex conjugate of the column under the (i, j) matrix element of the representation λ is the inverse QFT of $|\lambda; ij\rangle$. In other words, the table is (transpose of) the matrix of the unitary transformation U_{QFT}. The fact that the columns form an orthonormal set is the Schur orthogonality relation. This can be used to prove that U_{QFT} is in fact unitary, so the rows do as well.

	λ_{trivial}	λ_1				λ_2				$\cdots$
	$(1,1)$	$(1,1)$	$(1,2)$	$\cdots$	$(d_{\lambda_1}, d_{\lambda_1})$	$(1,1)$	$(1,2)$	$\cdots$	$(d_{\lambda_2}, d_{\lambda_2})$	$\cdots$
e										
g_1						$\sqrt{d_{\lambda_2}/\lvert G\rvert} \cdot [R_{\lambda_2}(g_1)]_{1,2}$				
g_2										
g_3										
$\vdots$										

2. Transformation of Functions Under Group Action

The Fourier basis $|\lambda; ij\rangle$ defined above turns out to be a useful basis for seeing how functions transform under group actions. In the group element basis, we have

$U_{\text{QFT}}^\dagger |\tau; nm\rangle = \sum_g \sqrt{\frac{d_\tau}{|G|}} [R_\tau(g)]_{nm}^* |g\rangle$

For $h \in G$, define the operators L_h and R_h by $L_h |g\rangle = |hg\rangle$ and $R_h |g\rangle = \left|gh^{-1}\right\rangle$. As representations of G on $\ell^2(G)$, these are know as the right and left regular representations, respectively. (I made the mistake of using R for both the right regular representation of G on $\ell^2(G)$ and the matrices of the irreducible representations of G, but it should be clear from context what is meant.) Then:

$$\begin{aligned} L_h U_{\text{QFT}}^\dagger |\tau; nm\rangle &= \sum_g \sqrt{\frac{d_\tau}{|G|}} [R_\tau(g)]_{nm}^* |hg\rangle = \sum_g \sqrt{\frac{d_\tau}{|G|}} \left[R_\tau(h^{-1}g)\right]_{nm}^* |g\rangle = \sum_g \sqrt{\frac{d_\tau}{|G|}} \left[R_\tau(h^{-1})R_\tau(g)\right]_{nm}^* |g\rangle \\ &= \sum_k \left[R_\tau(h^{-1})\right]_{nk}^* \sum_g \sqrt{\frac{d_\tau}{|G|}} [R_\tau(g)]_{km}^* |g\rangle = \sum_k \left[R_\tau(h^{-1})\right]_{nk}^* U_{\text{QFT}}^\dagger |\tau; km\rangle \\ &= \sum_k [R_\tau(h)]_{kn} U_{\text{QFT}}^\dagger |\tau; km\rangle \end{aligned}$$

$$\begin{aligned} R_h U_{\text{QFT}}^\dagger |\tau; nm\rangle &= \sum_g \sqrt{\frac{d_\tau}{|G|}} [R_\tau(g)]_{nm}^* \left|gh^{-1}\right\rangle = \sum_g \sqrt{\frac{d_\tau}{|G|}} [R_\tau(gh)]_{nm}^* |g\rangle = \sum_g \sqrt{\frac{d_\tau}{|G|}} [R_\tau(g)R_\tau(h)]_{nm}^* |g\rangle \\ &= \sum_k [R_\tau(h)]_{km}^* \sum_g \sqrt{\frac{d_\tau}{|G|}} [R_\tau(g)]_{nk}^* |g\rangle = \sum_k \left[R_\tau(h^{-1})\right]_{mk} U_{\text{QFT}}^\dagger |\tau; nk\rangle \end{aligned}$$

where I've used the fact that these are *unitary* representations. Sticking a U_{QFT} in front of these two equations, we find

$$U_{\text{QFT}} L_h U_{\text{QFT}}^\dagger |\tau; nm\rangle = \sum_{k=1}^{d_\tau} [R_\tau(h)]_{kn} |\tau; km\rangle$$

$$U_{\text{QFT}} R_h U_{\text{QFT}}^\dagger |\tau; nm\rangle = \sum_{k=1}^{d_\tau} \left[R_\tau(h^{-1})\right]_{mk} |\tau; nk\rangle$$

so that

$$\boxed{\begin{aligned} &\langle \lambda; ij| U_{\text{QFT}} L_h U_{\text{QFT}}^\dagger |\tau; nm\rangle = [R_\tau(h)]_{in} \delta_{\lambda\tau}\delta_{jm} \\ &\langle \lambda; ij| U_{\text{QFT}} R_h U_{\text{QFT}}^\dagger |\tau; nm\rangle = \left[R_\tau(h^{-1})\right]_{mj} \delta_{\lambda\tau}\delta_{in}. \end{aligned}}$$

What we've done is to use the QFT to block diagonalize the left and right regular representations of the group G on $\ell^2(G)$. Notice that, unless G is Abelian and thus $d_\lambda = 1$ for all λ, it isn't possible to fully simultaneously diagonalize the operators L_g and R_g. Instead, we find a set of $|G| = \sum_\lambda d_\lambda^2$ basis functions on G grouped into sets of d_λ that transform under the representation λ, with multiplicity d_λ. We can think of these as being organized into a block diagonal matrix with blocks of size d_λ, one for each irreducible representation λ, with the left regular representation mixing functions in the same column and the right regular representation mixing functions in the same row. This is exactly how the $d_\lambda \times d_\lambda$ matrices transform under left and right multiplications by g or g^{-1}.

This analysis sheds some light on what the group Fourier transform/QFT is - a way to analyze functions on G based on how they transform under the left and right actions of G on itself - but it doesn't necessarily explain why we should care. One answer is simply that it provides a way of seeing these functions from a different perspective. I don't know of any "real world" examples of what looking at the Fourier transform of a function on a generic finite group tells us. For instance, Fourier transforms of time-series yield frequency-space functions, which can be used e.g. to identify periodically recurring phenomena. It would be nice to have a similarly data-processing-motivated intuition for the Fourier transform on arbitrary groups. Part of the problem is that there aren't many cases (that I can think of) of data taking the form of a function on any interesting group. Some googling suggests that one application is to functions on symmetric groups, where the Fourier transform can be used to study mixing times for randomly shuffled decks of cards.

In the last post, I discussed the Fourier Transform on finite groups G. Here I'll discuss in more detail the case $G = Z_N$, the finite cyclic groups. This is what is usually called the Discrete Fourier Transform.

The (Quantum) Fourier Transform is a unitary transformation on $\mathcal{H} = \ell^2(Z_N) = C^N$, which in the standard basis is expressed by the matrix

$$U_{\mathrm{QFT}} = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{pmatrix} \tag{1}$$

with $\omega = e^{2\pi i/N}$. It turns out that if $N = 2^n$ and we interpret the Hilbert space $\mathcal{H} = C^N$ as the state space of n qubits, U_{QFT} can be implemented efficiently, using only about n^2 elementary (1- and 2-qubit) gates.

Efficient Circuit for the QFT

Consider an n-qubit state

$$|\psi\rangle = \sum_{s=0}^{2^n-1} \alpha(s) \, |s\rangle \, . \tag{2}$$

The n-qubit QFT acts on this state as follows:

$$\begin{aligned}
\sqrt{2^n}U_{\mathrm{QFT}}\left|\psi\right\rangle &= \sum_{k=0}^{2^n-1}\sum_{s=0}^{2^n-1}\alpha(s)e^{2\pi isk/2^n}\left|k\right\rangle \\
&= \sum_{k=0}^{2^n-1}\left[\sum_{s=0,\mathrm{even}}^{2^n-1}\alpha(s)e^{2\pi isk/2^n}+\sum_{s=0,\mathrm{odd}}^{2^n-1}\alpha(s)e^{2\pi isk/2^n}\right]\left|k\right\rangle \\
&= \sum_{k=0}^{2^n-1}\left[\sum_{t=0}^{2^{n-1}-1}\alpha(2t)e^{2\pi itk/2^{n-1}}+e^{2\pi ik/2^n}\sum_{t=0}^{2^{n-1}-1}\alpha(2t+1)e^{2\pi itk/2^{n-1}}\right]\left|k\right\rangle \\
&= \sum_{k=0}^{2^{n-1}-1}\sum_{t=0}^{2^{n-1}-1}\alpha(2t)e^{2\pi itk/2^{n-1}}\left|k\right\rangle+\sum_{k=0}^{2^{n-1}-1}e^{2\pi ik/2^n}\sum_{t=0}^{2^{n-1}-1}\alpha(2t+1)e^{2\pi itk/2^{n-1}}\left|k\right\rangle \\
&\quad+\sum_{k=0}^{2^{n-1}-1}\sum_{t=0}^{2^{n-1}-1}\alpha(2t)e^{2\pi itk/2^{n-1}}\left|k+2^{n-1}\right\rangle-\sum_{k=0}^{2^{n-1}-1}e^{2\pi ik/2^n}\sum_{t=0}^{2^{n-1}-1}\alpha(2t+1)e^{2\pi itk/2^{n-1}}\left|k+2^{n-1}\right\rangle \\
&= \left(\sqrt{2^{n-1}}U_{\mathrm{QFT}}^{(n-1)}\otimes I\right)\left|\psi_0\right\rangle\left|0\right\rangle+\left(I_{n-1}\otimes X\right)\left(P^{(n-1)}\otimes I\right)\left(\sqrt{2^{n-1}}U_{\mathrm{QFT}}^{(n-1)}\otimes I\right)\left|\psi_1\right\rangle\left|1\right\rangle \\
&\quad+\left(I_{n-1}\otimes X\right)\left(\sqrt{2^{n-1}}U_{\mathrm{QFT}}^{(n-1)}\otimes I\right)\left|\psi_0\right\rangle\left|0\right\rangle-\left(P^{(n-1)}\otimes I\right)\left(\sqrt{2^{n-1}}U_{\mathrm{QFT}}^{(n-1)}\otimes I\right)\left|\psi_1\right\rangle\left|1\right\rangle \\
&= \left(I_{n-1}\otimes\left(I+X\right)\right)\left(\sqrt{2^{n-1}}U_{\mathrm{QFT}}^{(n-1)}\otimes I\right)\left|\psi_0\right\rangle\left|0\right\rangle \\
&\quad+\left(I_{n-1}\otimes\left(X-I\right)\right)\left(P^{(n-1)}\otimes I\right)\left(\sqrt{2^{n-1}}U_{\mathrm{QFT}}^{(n-1)}\otimes I\right)\left|\psi_1\right\rangle\left|1\right\rangle \\
&= \left(I_{n-1}\otimes\sqrt{2}H\right)cP^{(n-1)}\left(\sqrt{2^{n-1}}U_{\mathrm{QFT}}^{(n-1)}\otimes I\right)\left|\psi\right\rangle
\end{aligned} \tag{3}$$

where the P and cP gates are phase and controlled phase gates defined implicitly, and H is the Hadamard gate. Therefore, we have

$$U_{\mathrm{QFT}}^{(n)} = \left(I_{n-1}\otimes H\right)cP^{(n-1)}\left(U_{\mathrm{QFT}}^{n-1}\otimes I\right). \tag{4}$$

The controlled phase gate can be implemented with $n-1$ elementary (2-qubit) controlled gates, and the Hadamard gate H is a single qubit gate, so if we can implement the $(n-1)$-qubit QFT, we just need n additional elementary gates to implement the n-qubit QFT. The 1-qubit QFT is just a Hadamard gate, so we can implement the n-qubit QFT with $n+(n-1)+(n-2)+\ldots+1 = n(n+1)/2$ elementary 1- and 2-qubit gates. I won't reproduce the circuit here - nice drawings are available anywhere qubits are sold.

That's the good news. The bad news is that the QFT is not an algorithm for the Discrete Fourier Transform in the sense we would really like it to be, i.e. a black box that takes a vector of classical data and outputs, as classical (i.e., readable) data, the Fourier transform of that vector. Instead, the QFT is a unitary transformation that takes as input a quantum state and gives as output another quantum state. So the QFT is not an exponentially faster replacement for the classical FFT algorithm, which does precisely this in time $N\log N = n2^n$. In order to make use of the QFT as a limited replacement for the FFT, we need to figure out (1) what classical data can be efficiently encoded in the proper form and (2) what data can be efficiently extracted from the output state.

Efficiently Encodable Data

One example of data that can be efficiently encoded into a quantum state, given that it is already stored in a quantum accessible manner, is the following. Suppose that we have an oracle $\mathcal{D}$ that was capable of performing the transformation $\mathcal{D}:\left|s\right\rangle\mapsto\left|s\right\rangle\left|D(s)\right\rangle$ where D is a $\{0,1\}$-valued function. Such an oracle can be efficiently implemented with the QRAM protocol (see e.g. https://arxiv.org/pdf/0708.1879.pdf). Then applying $\mathcal{D}$ to the even superposition over all computational basis states (efficiently preparable from the zero state of n qubits via n Hadamard gates), measuring the second register in the computational basis, and

post-selecting for outcome 1, we obtain the state

$$\mathcal{N} \sum_{s:D(s)=1} |s\rangle \tag{5}$$

where $\mathcal{N}$ is the normalization factor. This is a state of the form

$$|\psi\rangle = \sum_{s=0}^{M-1} \alpha(s) |s\rangle \tag{6}$$

with $\alpha(s) \propto D(s)$, so now the QFT can be used to Fourier transform D.

If we assume that a constant fraction of the values of s have $D(s) = 1$, then we can prepare the state $|\psi\rangle$ with exponentially high probability in the number of tries.

Efficiently Extractable Data

Here I'll show that the QFT can be used in an efficient procedure for estimating the period of a quantum state. Since this application is at the heart of Shor's algorithm, it's probably the most famous use of the QFT.

Suppose that the vector $|\psi\rangle = \sum_{s=0}^{N-1} \alpha(s) |s\rangle$ is periodic with period p, i.e. $\alpha(s) = \alpha(s+p)$ for all s. Suppose also that p divides M. We have that

$$T^p |\psi\rangle = |\psi\rangle \tag{7}$$

where $T |s\rangle = |s+1\rangle$ is the shift operator. Using the unitarity of the QFT, we then have

$$\left(U_{\mathrm{QFT}} T U_{\mathrm{QFT}}^\dagger\right)^p U_{\mathrm{QFT}} |\psi\rangle = U_{\mathrm{QFT}} |\psi\rangle . \tag{8}$$

Expanding $U_{\mathrm{QFT}} |\psi\rangle$ in the Fourier basis, this gives

$$\sum_{k=0}^{M-1} \hat{\alpha}(k) \hat{T}^p |k\rangle = \sum_{k=0}^{M-1} \hat{\alpha}(k) |k\rangle \tag{9}$$

where the hat denotes change of basis into the Fourier basis. It is simple to verify that $\hat{T} |k\rangle = e^{2\pi i k/M}$, so that in order for this equation to hold, $\hat{\alpha}(k)$ must be zero unless kp is a multiple of M or, equivalently, k is a multiple of M/p (which we assumed was an integer). There are p such values of k.

This analysis shows that if we apply the QFT to $|\psi\rangle$ and measure in the computational basis, we will obtain a value of k such that $k = nM/p$. Suppose we prepare J copies of $|\psi\rangle$ and apply the QFT to each one, then measure in the computational basis. We obtain values $k_1, \ldots, k_J$, with $k_i = n_i M/p$. If the gcd of the n_i is one, then the gcd of the k_i is M/p, and since M is known, we obtain p. Of course, it's possible that the $\hat{\alpha}(k)$ are exponentially small except for some k a multiple of M/p, in which case we'll underestimate p with high probability unless J is exponentially large, in which case the quantum speedup is lost. However, in this case, $|\psi\rangle$ is very close to a state with the period we estimated, so it's not unreasonable to be satisfied with this performance.

On the other hand, suppose that all of the p non-zero $\hat{\alpha}(k)$ have roughly the same magnitude. Then the probability that with J samples we correctly find p is the same as the probability that the gcd of J integers drawn uniformly at random from $0, 1, 2, \ldots, p-1$ is one. It is not too hard to see that this probability is bounded exponentially (in J) close to one. If we didn't do the classical post-processing of the outcomes and just looked for the smallest value of k that appeared, we would not get this performance.

Failed Example

Combining these three parts (state preparation/data entry, transformation/data processing, and measurement/data extraction), we can try to build a quantum algorithm that gives with high probability a speedup over a classical computer for a simple problem. Suppose that we're given a string of 2^n bits, stored either in a classical RAM or in a QRAM, and would like to determine the period of the data. Assume that a constant fraction of the bits are 1. Then with a constant expected number of calls to the QRAM, we can prepare the n-qubit state whose amplitudes encode the unknown bitstring. With $\mathcal{O}(n^2)$ elementary gates, we perform the QFT on this state. Then we make a single n-qubit measurement to obtain a value of k present with non-zero amplitude in it's Fourier transform. We need to repeat this procedure several times in order to give an estimate of the period. But how many?

Suppose that a malicious user designs a family of classical inputs A_n and B_n of lengths 2^{2n}, where the A_n have period 2^n and the B_n are identical to the A_n except that the first 1 in the second half of the string is changed to a 0, and the first 0 to a 1, so that they have period 2^{2n}. Then the corresponding quantum states $|A_n\rangle$ and $|B_n\rangle$ are exponentially close in n, meaning that the error probability for distinguishing $|A_n\rangle$ and $|B_n\rangle$ with a single measurement is exponentially close to $1/2$. Therefore, it is exponentially unlikely that, given state $|B_n\rangle$, we measure an odd value of k after performing the QFT, as if it were more likely, the discrimination procedure that reports A if an even k is measured and B if an odd k is measured would have a lower error probability than is possible given the distance between the states. This means that given the inputs B_n, we need to repeat the basic QFT procedure above exponentially many times in order to have a large probability of obtaining an odd value of k and accurately determining the period.

This counterexample tells us that, even for this very contrived problem, the QFT does not give a speedup over the classical algorithm, at least for the worst case. However, it's worth pointing out that for some applications, it might be a feature that the quantum algorithm with sub-exponentially many runs of the QFT is insensitive to small deviations from perfect periodicity.