Order Finding by Phase Estimation

The QFT is a useful tool, and not just in constructing quantum algorithms. We can also use it to design efficient devices for measuring parameters appearing in physical Hamiltonians, like the strength of gravitational waves, for instance. A simpler example is estimating the strength of a magnetic field by using spin-1/2 systems. If we have n systems, we could prepare them in the spin-up state along an axis perpendicular to the field direction; this way they will precess around that axis when placed in the field. If we arrange things so that each successive spin spends twice as long in the field, then we end up with the state

$$\left(\bigotimes_{j=1}^{n} e^{-i 2^j b \sigma_z t} \right) |+\rangle^{\otimes n}$$

where the field $\vec{B} = 2b\hat{z}$ and the Hamiltonian is $H = -\vec{\mu}\cdot\vec{B}$ for $\vec{\mu} = \frac{\hbar}{2}\vec{\sigma}$; t is the fixed time interval, which we take to be 2π

$$= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{-2\pi i b k} |k\rangle$$

This state is insensitive to the parts of b larger than 1 (which is set by $t = 2\pi$), so assume $b = 0.b_1 b_2 \ldots$

By employing the QFT we can get a reliable estimate of b.

In the context of quantum algorithms we can use the QFT and phase estimation to solve the order finding problem, which in turn can be used to find the prime factors of a given integer.

Working in $\mathbb{Z}_d$, we can define the set $\mathbb{Z}_d^* = \{a \in \mathbb{Z}_d : \gcd(a,d)=1\}$ the elements relatively prime to d. For $\mathbb{Z}_6$, $\mathbb{Z}_6^* = \{1,5\}$; for $\mathbb{Z}_9$, $\mathbb{Z}_9^* = \{1,2,4,5,7,8\}$. When endowed with multiplication mod d, the sets $\mathbb{Z}_d^*$ become groups since if $a,b \perp d$, then $ab \perp d$ and since $ab \bmod d$ is distinct from $a \in \mathbb{Z}_d^*$ for all $b \in \mathbb{Z}_d^*$ not equal to 1 ($ab \bmod d = ab' \bmod d \Rightarrow a(b-b') \bmod d = 0 \Rightarrow b-b' \bmod d = 0$ since $a \in \mathbb{Z}_d^*$) and there are only a finite number of elements in $\mathbb{Z}_d^*$, then $\exists\ b \in \mathbb{Z}_d^*$ such that $ab = 1 \bmod d$.

In particular, this argument also implies that $\exists\ r \in \mathbb{Z}$ such that $a^r = 1 \bmod d$. r is called the order of a. For example, the order of 4 in $\mathbb{Z}_9^*$ is 3, since $4^3 = 64 = 9 \cdot 7 + 1 = 1 \bmod 9$.

The order-finding problem is thus: given d and $a \in \mathbb{Z}_d^*$, find r.

Classically the problem is not known to be efficient (in P). The brute force approach of computing powers of a will take time linear in d, but exponential in $\log d$, which is what counts (input is of size $\log d$).

How can we solve this problem using the phase estimation scheme? First we need a unitary: Let $n = \log \lfloor (d-1) \rfloor + 1$ so that $0, 1, \ldots, d-1$ can be represented by n bits. Then define

$U_a |x\rangle = |ax \bmod d\rangle$ (for $d \le x < 2^n$ set $U_a|x\rangle = |x\rangle$)

Can U_a and U_a^k be efficiently implemented? U_a itself is just multiplication. If a has the binary expansion $a = (a_1 a_2 \cdots a_n)_2$ $= 2^{n-1} a_1 + 2^{n-2} a_2 + \cdots + 2^0 a_1$ and $x = (x_1 x_2 \cdots x_n)_2$, then multiplying a and x requires n^2 binary operations to determine each term and another $n-1$ additions, so it can be done in $O(n^2)$ steps.

Similary, $U_a^k |x\rangle = |a^k x \bmod d\rangle$, so we just need a^k.

For $k = (k_1 \cdots k_n)_2$ we have $a^k = (a^{2^{n-1}})^{k_1} (a^{2^{n-2}})^{k_2} \cdots (a)^{k_n}$ meaning that we <u>really</u> only need the powers a^{2^m}. But these are easily obtained by repeated squaring. Altogether we need n powers, which means $n-1$ multiplications (each one taking $O(n^2)$ steps. Then we have to multiply the terms together, n more multiplications of $O(n^2)$ steps. In sum, we only need $O(n^3)$ steps, so the implementation is efficient.

For the phase estimation algorithm we also need an eigenvector of U_a. What do these look like and how can we construct them?

Using the order r, it's clear that $|\psi_0\rangle = \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} |a^j\rangle$ is an eigenvector with eigenvalue 1. We can also use roots of unity as coefficients: $|\psi_k\rangle = \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} \xi_r^{-jk} |a^j\rangle$ has eigenvalue ξ_r^j, where $\xi_r = e^{2\pi i/r}$. The $|\psi_k\rangle$ only span a space of dimension r (which we don't know), but they are orthogonal and they do have the property that

$$\frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |\psi_k\rangle = \frac{1}{r} \sum_{j,k} \xi_r^{-jk} |a^j\rangle = |a^0\rangle = |1\rangle,$$

which we'll use later. For now let's just assume that we have $|\psi_1\rangle$ and see how we can use phase estimation to find r.

Since the phase of $|\psi_1\rangle$ is $1/r$, we need to estimate the phase reliably enough to distinguish $1/r$ from $1/r \pm 1$. If the estimate is within $1/2r(r+1)$ of the correct value, which is $\frac{1}{2}\left(\frac{1}{r} - \frac{1}{r+1}\right)$, then we'll have enough precision to determine r. Of course, we don't know what $1/2r(r+1)$ is, since we don't know r, but $r < d$, so then $\frac{1}{2r(r+1)}$ becomes $\sim \frac{1}{2d^2}$. Suppose the phase estimate is r' and $\left|\frac{1}{r'} - \frac{1}{r}\right| \leq \frac{1}{2d^2}$. The estimate of r itself, which we get by inverting r' and rounding to the nearest integer, will

be good if $|r'-r| < \frac{1}{2}$. Is a precision of $\frac{1}{2d^2}$ enough? yes:

$$|r'-r| = \left| \frac{1}{\frac{1}{r} - \frac{1}{2d^2}} - r \right| = \left| \frac{2d^2 r}{2d^2 - r} - r \right| = \left| \frac{r^2}{2d^2 - r} \right| \leq \frac{(d-1)^2}{2d^2 - d} \leq \frac{1}{2}$$

How many bits do we need for a precision of $\frac{1}{2d^2}$? $d \sim 2^n$, so $m = 2n+1$ bits are enough.

What if we had used the eigenvector $|\psi_k\rangle$? It's phase is k/r, and if we know k, then we follow the same procedure as above after dividing by k. Even if we don't know k we can find r by using the continued fraction algorithm: Call the phase estimate φ. It can be appoximated by the continued fractions

$$\varphi = \cfrac{1}{\varphi_1 + \cfrac{1}{\varphi_2 + \ddots \varphi_m}} = [0; \varphi_1, \varphi_2 \ldots \varphi_m] \quad \text{for increasing } m$$

Each approximation (level m) is closer to $\frac{k}{r}$ than the previous (level $m-1$) and it can be shown that for $\left|\varphi - \frac{k}{r}\right| \leq \frac{1}{2d^2}$, $\frac{k}{r} = [0; \varphi_1, \ldots, \varphi_\ell]$ for some ℓ (such that the denominator of $[0; \varphi_1 \ldots \varphi_\ell] < d$).

So how do we get the φ_i? Easy. For φ_1, invert φ and take the integer part. Call the rest (the noninteger part) x_1. Invert this; the integer part of the result is φ_2. In general, $\varphi_k = \left\lfloor \frac{1}{x_{k-1}} \right\rfloor$ and $x_{k+1} = x_k - \varphi_{k+1}$ with $\varphi_0 = \lfloor \varphi \rfloor = 0$ (in this case) and $x_0 = \frac{1}{\varphi}$.

Suppose we're looking for the order of 4 in $\mathbb{Z}_9^*$ (it's 3).

We use phase estimation on a random $|\psi_k\rangle$, which happens to be $|\psi_2\rangle$. In order to ensure sufficient precision, the phase estimation scheme uses $9 = 2 \cdot \lceil \log 9 \rceil + 1$ qubits, and suppose that we get the (binary) outcome $0.101010011_2 = 339/512$ (which satisfies $\left|\frac{339}{512} - \frac{2}{3}\right| < \frac{1}{2 \cdot 9^2}$).

Working out the continued fraction expansion, we get

$$\varphi_1 = 1,\ \varphi_2 = 1,\ \varphi_3 = 1,\ \varphi_4 = 23,\ \varphi_5 = 1,\ \varphi_6 = 2,\ \varphi_7 = 2$$

which gives successive approximations

$$1,\ \frac{1}{2},\ \frac{2}{3},\ \frac{47}{71},\ \frac{49}{74},\ \frac{145}{219},\ \frac{339}{512}$$

The only one with denominator less than $d = 9$ is $\frac{2}{3}$, so we conclude that the order of 4 is 3. Similarly, the most likely outcome, 0.101010101_2 (which is $2/3$ up to 9 bits of precision) has a continued fraction expansion $(0; 1, 1, 1, 170)$, giving approximations $1, \frac{1}{2}, \frac{2}{3}, \frac{341}{512}$, so again we get 3 as the order.

If k divides r (e.g. $k = 2$, $r = 6$), then we only get a factor of r.

This is all well and good, but without an eigenvector it won't work, even an unknown eigenvector. But here's where the fact that $|1\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |\psi_k\rangle$ comes into play, along with the fact that the eigenvector is not changed by the algorithm.

Suppose we feed $|0\rangle^{\otimes m}|\psi_k\rangle$ into the phase estimation circuit. Then the output is $|\alpha_k\rangle|\psi_k\rangle$ for some state $|\alpha_k\rangle$. If we feed $|0\rangle^{\otimes m}|1\rangle$ in instead, we just get $\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|\alpha_k\rangle|\psi_k\rangle$, which means that the state of the control bits is $\frac{1}{r}\sum_{k=0}^{r-1}|\alpha_k\rangle\langle\alpha_k|$, i.e. a uniform mixture of what we would have gotten for a given k. In other words, we get an estimate of k/r for a random $0 < k < r$.

Therefore, repeating the phase estimation several times we get a sequence of k_j/r estimates, compute r (or a factor of it) from each, and take the least common multiple of the results as the output. The entire procedure is efficient and delivers the correct order with high probability.