

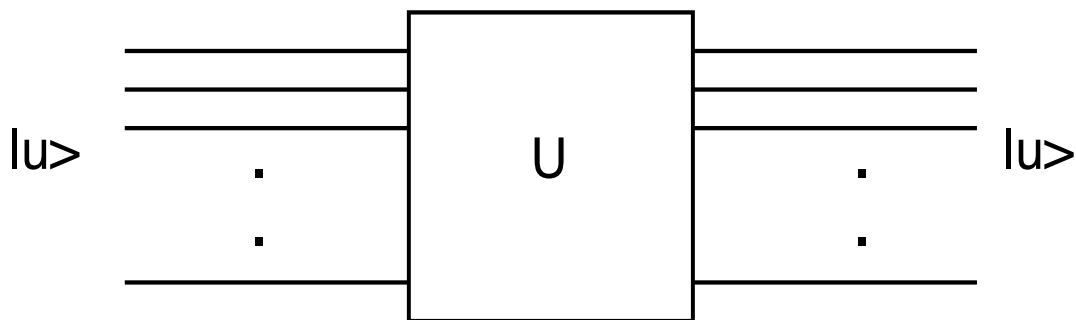
Phase estimation

Last time we saw how the quantum Fourier transform made it possible to find the period of a function by repeated measurements and the greatest common divisor (GCD) algorithm. We will now look at this same problem again, but making use of the Fourier transform in a more sophisticated way: by Kitaev's *phase estimation* algorithm.

By itself, the phase estimation algorithm is a solution to a rather artificial problem. But this solution turns out to be useful as a piece of several other algorithms, to solve much more natural and important problems.

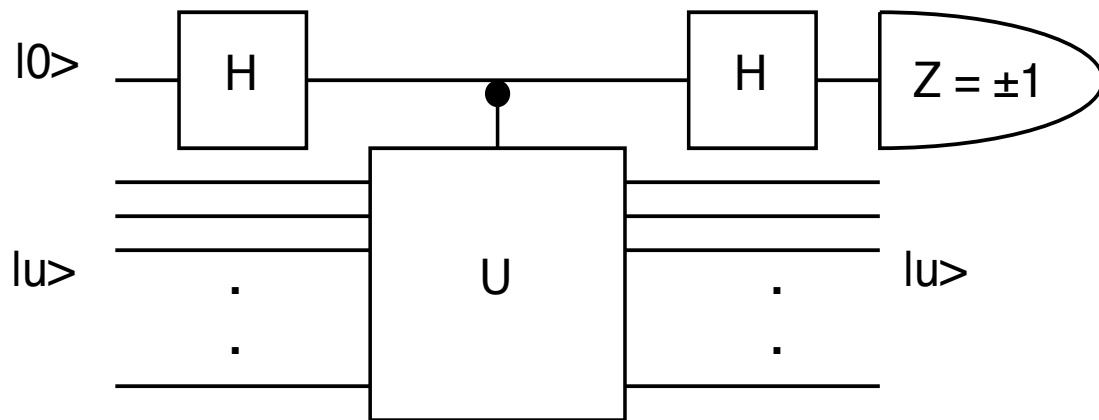
Suppose we are given a unitary operator \hat{U} on n q-bits, which has a known eigenstate $|u\rangle$ with an unknown eigenvalue $\exp(2\pi i\phi)$. We wish to find the phase ϕ with some precision (say, m bits).

The first thing we might try is to prepare n q-bits in the state $|u\rangle$, and carry out the unitary transformation \hat{U} on them:



Is there some measurement on the bits which will give us information about the phase ϕ ? The answer, of course, is no: the unitary operator just produces an overall phase on the state, with no observable consequences.

Here is a more sophisticated approach to the problem. Suppose we perform the following circuit:



Instead of performing the operation \hat{U} , we do controlled- \hat{U} , and then measure the control bit. It is not difficult to see that the expectation of \hat{Z} on the control bit is

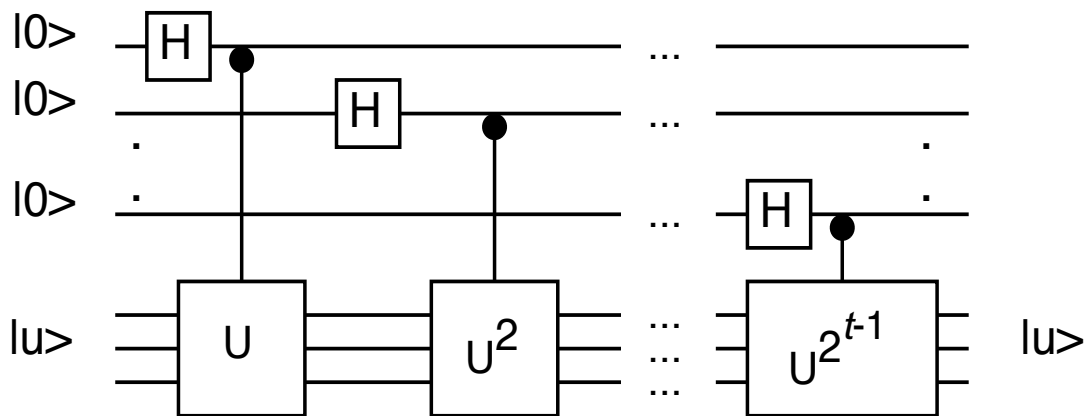
$$\begin{aligned} \langle \hat{Z} \rangle &= \langle u | (\hat{U} + \hat{U}^\dagger) | u \rangle / 2 \\ &= \frac{1}{2} (e^{2\pi i \phi} + e^{-2\pi i \phi}) = \cos(2\pi \phi). \end{aligned}$$

The n bits are left in state $|u\rangle$, so we can perform this circuit N times with different control bits and get an estimate of $\cos(2\pi\phi)$ with an accuracy of roughly $1/\sqrt{N}$. By starting the control bit in state $|1\rangle$, we can get a similar estimate of $\sin(2\pi\phi)$.

Unfortunately, the convergence of this algorithm is very slow. To estimate ϕ with m bits of accuracy requires $1/\sqrt{N} \sim 2^{-m}$, so the circuit would have to be repeated $O(2^{2m})$ times.

In fact, for some problems we can't do much better than that. But with a few extra assumptions, we can improve performance enormously.

The main assumption we will make is that we can not only do controlled- \hat{U} efficiently, we can also do controlled- \hat{U}^2 , controlled- \hat{U}^4 , and so on, up to controlled- $\hat{U}^{2^{t-1}}$. We might imagine that we have t different oracles which perform these operations. We then use them to carry out this circuit:



After this circuit, the n target bits are still left in the state $|u\rangle$, and the t control bits are left in the state

$$2^{-t/2} (|0\rangle + e^{2\pi i\phi}|1\rangle) \otimes (|0\rangle + e^{4\pi i\phi}|1\rangle) \\ \otimes \cdots \otimes (|0\rangle + e^{2^t \pi i\phi}|1\rangle).$$

Now that we have prepared this state, what do we do with it? Let us suppose that there is an exact t -bit expression for the phase, $\phi = 0.\phi_1\phi_2 \dots \phi_t$. Then,

$$\begin{aligned}
 e^{2\pi i\phi} &= e^{2\pi i0.\phi_1\dots\phi_t} . \\
 e^{4\pi i\phi} &= e^{2\pi i\phi_1.\phi_2\dots\phi_t} \\
 &= e^{2\pi i\phi_1+2\pi i0.\phi_2\dots\phi_t} \\
 &= e^{2\pi i0.\phi_2\dots\phi_t} . \\
 e^{2^j\pi i\phi} &= e^{2\pi i0.\phi_j\dots\phi_t} .
 \end{aligned}$$

The state of the control bits can be written

$$\begin{aligned}
 &2^{-t/2} (|0\rangle + e^{2\pi i0.\phi_1\dots\phi_t} |1\rangle) \\
 &\otimes \dots \otimes (|0\rangle + e^{2\pi i0.\phi_t} |1\rangle) ,
 \end{aligned}$$

which is the Fourier transform of the basis state $|\phi_1 \dots \phi_t\rangle = |2^t\phi\rangle!$

Rewriting our expression for the state of the t control bits, we get

$$\frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} e^{2\pi i \phi j} |j\rangle.$$

This has the form of a Fourier-transformed state. Applying the *inverse* Fourier transform, we get

$$|\Psi\rangle \approx |2^t \phi\rangle.$$

If ϕ has an exact t -bit expression, then this estimate will also be exact. If not, it will be quite close. How close is “quite close?”

We want the probability that one will measure a number significantly different from $2^t\phi$ to be small. Let $2^t\phi$ be the correct number, and $2^t\phi'$ the number that is actually measured. Define a to be our desired accuracy bound (i.e., we want $2^t|\phi - \phi'| \leq a$). It is not difficult to show that the probability of being outside this bound is

$$p(2^t|\phi - \phi'| > a) \leq \frac{1}{2(a - 1)}.$$

This implies that if we want an accuracy of at least m bits with probability at least $(1 - \epsilon)$ we need to use t control bits with

$$t = m + \log \left(1 + \frac{1}{2\epsilon} \right),$$

where we round this down to the next lowest integer.

We now see the complete phase estimation procedure:

1. Prepare the t -bit *control* register in state $|0\rangle$, and the n -bit *target* register in state $|u\rangle$.
2. Perform Hadamards on the control bits.
3. From each of the control bits in succession, do a controlled- \hat{U}^{2^j} from the j th control bit on the n target bits.
4. Do an inverse Fourier transform on the t control bits, and measure them in the computational basis. The measured bit values ϕ_1, \dots, ϕ_t give an estimate of the phase $\phi \approx 0.\phi_1 \dots \phi_t$.

Costs and assumptions

Assuming that the controlled- \hat{U}^{2^j} unitaries are given by oracles (and hence free), the complexity of this algorithm is basically that of the inverse Fourier transform, $O(t^2)$. If, however, we have to perform circuits for the controlled- \hat{U} unitaries, they will tend to dominate the complexity. Even if we have an efficient circuit for controlled- \hat{U} , we need efficient circuits for all the controlled- \hat{U}^{2^j} gates as well; just repeating the controlled- \hat{U} 2^j times will make the complexity exponential.

There is another somewhat artificial assumption as well. It is assumed that we don't know the eigenvalue $e^{2\pi i\phi}$, but that we can prepare the eigenvector $|u\rangle$. While this may sometimes be true, in most cases it will not be.

In spite of all this, phase estimation is at the heart of many other quantum algorithms.

Period-finding: Take 2

Now let's look again at our problem from last time: finding the period of a periodic function. Recall that the first steps of this algorithm involved preparing an input and output register in the state

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle |f(x)\rangle.$$

Last time, we applied a Fourier transform to the input register, and then measured it; we will now look at an almost identical procedure in a somewhat different way.

First, we define the inverse Fourier transform state

$$|\tilde{f}(\ell)\rangle = \frac{1}{\sqrt{r}} \sum_{x=0}^{r-1} e^{-2\pi i \ell x / r} |f(x)\rangle,$$

where r is the (unknown) period of $f(x)$.

We can invert this expression to get

$$|f(x)\rangle = \frac{1}{\sqrt{r}} \sum_{\ell=0}^{r-1} e^{2\pi i \ell x / r} |\tilde{f}(\ell)\rangle.$$

Substituting this, we get

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle |f(x)\rangle \approx$$

$$\frac{1}{\sqrt{r 2^n}} \sum_{\ell=0}^{r-1} \sum_{x=0}^{2^n-1} e^{2\pi i \ell x / r} |x\rangle |\tilde{f}(\ell)\rangle,$$

where the approximation is necessary when 2^n is not an exact multiple of r (which it generally won't be). However, the same bounds on accuracy that we deduced for phase-estimation work here, as well.

The first register has the form of a Fourier-transformed state; we apply the inverse Fourier transform to get

$$\frac{1}{\sqrt{r}} \sum_{\ell=0}^{r-1} |2^n \ell / r\rangle |\tilde{f}(\ell)\rangle.$$

If we measure the first register, we will get $2^n \ell / r$ for some value of ℓ , chosen at random from $0, \dots, r - 1$.

Last time we considered the use of Euclid's algorithm to find the period r , but didn't examine the problem of accuracy in the case when $2^n / r$ was not a whole number. We will look at another, more direct way of extracting the period in a moment. First, let's see the connection between period-finding and phase estimation.

Let \hat{U} be the unitary transformation that translates the argument of $f(x)$ by one:

$$\hat{U}|f(x)\rangle = |f(x+1)\rangle.$$

We can rewrite the transformation $|x\rangle|0\rangle \rightarrow |x\rangle|f(x)\rangle$ in terms of controlled- \hat{U} operations.

1. Shift the output register to $|f(0)\rangle$.
2. Apply a controlled- \hat{U} operation from the least significant bit of the input register to the output register:

$$|x\rangle|f(0)\rangle \rightarrow |x\rangle|f(x_0)\rangle.$$

3. Apply a controlled- \hat{U}^2 operation from the second bit:

$$|x\rangle|f(x_0)\rangle \rightarrow |x\rangle|f(2x_1 + x_0)\rangle.$$

4. Successively apply controlled- \hat{U}^{2^j} from bits $j = 2, \dots, n-1$. The final state is

$$|x\rangle|f(2^{n-1}x_{n-1} + \dots + x_0)\rangle \equiv |x\rangle|f(x)\rangle.$$

This has the form of the phase-estimation circuit, except that in that case the “output register” is prepared in an eigenstate $|u\rangle$. What are the eigenstates of \hat{U} ? It is easy to check that they are in fact the states $|\tilde{f}(\ell)\rangle$ that we introduced before, with eigenvalues

$$\hat{U}|\tilde{f}(\ell)\rangle = e^{2\pi i\ell/r}|\tilde{f}(\ell)\rangle.$$

Since we don't know the value r , we can't actually prepare the second register in the state $|\tilde{f}(\ell)\rangle$. But in fact, this doesn't matter for the purposes of period finding. By preparing the register in state $|0\rangle$ (or $|f(0)\rangle$), we are actually preparing a *superposition* of all the eigenstates $|\tilde{f}(\ell)\rangle$. When we make the final measurement, one of these states is picked out at random; but our ability to solve the problem doesn't depend on which ℓ is selected.

Note also that for this problem, the ability to efficiently perform $|x\rangle|0\rangle \rightarrow |x\rangle|f(x)\rangle$ implies the ability to efficiently perform the controlled- \hat{U}^{2^j} transformation for any j .

The continued fraction algorithm

We will now see a different way of extracting the period r from an approximate expression for $2^n \ell/r$. This gives an n -bit approximation to the fraction ℓ/r .

If we had an expression for ℓ/r as a fraction in most reduced form, then the denominator of the fraction would be r , except in the improbable event that ℓ and r shared a common factor. (It is easy to check that the denominator really is the period just by computing some values of $f(x + kr)$ for various x and k . If it is not, we run the program to get a new value of ℓ and check again.)

What we would like is to find the fraction which is closest to our n -bit expression, and check if its denominator is the period. We do this using *continued fractions*.

Continued fractions are expressions of the form

$$[a_0, \dots, a_M] \equiv a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\dots + \frac{1}{a_M}}}},$$

where a_0, \dots, a_M are integers. Any rational number can be put in this form with some finite set of a_j . Irrational numbers can also be expressed in this form; however, their expansion is *infinite*.

Suppose we truncate the series $[a_0, \dots, a_M]$ at the point $m < M$. This smaller continued fraction $[a_0, \dots, a_m]$ is an approximation to the full fraction, called the *mth convergent* of the fraction. The convergents of a continued fraction provide the best fractional approximations to any number.

The utility of the continued fraction approach comes from the following theorem. Suppose that ϕ is the n -bit approximate result from our measurement, and ℓ/r is the “true” fraction we are trying to find. If

$$\left| \frac{\ell}{r} - \phi \right| \leq \frac{1}{2r^2},$$

then ℓ/r is a convergent of the continued fraction for ϕ .

We construct the continued fraction in a series of steps. Let $\phi = 0.\phi_1 \dots \phi_n$. Then we start with

$$\phi = \frac{1}{\frac{1}{0.\phi_1 \dots \phi_n}} = \frac{1}{1/\phi}.$$

We let a_1 be the integer part of $1/\phi$, and f_1 be the remaining fractional part: $1/\phi = a_1 + f_1$.

Then

$$\phi = \frac{1}{a_1 + f_1} = \frac{1}{a_1 + \frac{1}{1/f_1}}.$$

We similarly define a_2 to be the integer part of $1/f_1$ and f_2 to be the remaining fractional part, and get the fraction

$$\phi = \frac{1}{a_1 + \frac{1}{a_2 + f_2}} = \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{1/f_2}}},$$

and continue on in the same manner. We get the m th convergent by setting f_m to zero and reducing the fraction to its simplest form. As we go through this procedure, we check the denominator of each convergent to see if it is the period of the function.

In fact, Euclid's algorithm is closely related to the continued fraction approach. At each step in the procedure, we divide one number by another and get the remainder. The successive quotients form a continued fraction expansion for the original fraction p/q .

Next time: the factoring algorithm.