1 Readings

Benenti et al., Ch. 3.12 - 3.14

Stolze and Suter, Quantum Computing, Ch. 8.3

Nielsen and Chuang, Quantum Computation and Quantum Information, Ch. 5.2 - 5.3, 5.4.1 (NC use phase estimation for this, which we present in the next lecture)

literature: Ekert and Jozsa, Rev. Mod. Phys. **68**, 733 (1996)

2 Introduction

With a fast algorithm for the Quantum Fourier Transform in hand, it is clear that many useful applications should be possible. Fourier transforms are typically used to extract the periodic components in functions, so this is an immediate one. One very important example is finding the period of a modular exponential function, which is also known as order-finding. This is a key element of Shor's algorithm to factor large integers N. In Shor's algorithm, the quantum algorithm for order-finding is combined with a series of efficient classical computational steps to make an algorithm that is overall polynomial in the input size $n = log_2 N$, scaling as $O(n^2 lognloglogn)$. This is better than the best known classical algorithm, the number field sieve, which scales superpolynomially in n, i.e., as $exp(O(n^{1/3}(logn)^{2/3}))$. In this lecture we shall first present the quantum algorithm for order-finding and then summarize how this is used together with tools from number theory to efficiently factor large numbers.

3 Shor's order-finding algorithm

3.1 modular exponentiation

Recall the exponential function a^x. The modular exponential function is obtained by taking this function and calculating the remainder on division by N, i.e., $F_N(x) = a^x mod\ N$. The order of the modular exponential, referred to as the order of $a\ mod\ N$ or $\mathrm{ord}(a)$, is the smallest positive integer r such that

$$a^r mod\ N = 1$$

Equivalently, we can say that r is the period of this function, since from the above equation we have

$$\begin{aligned} a^r &= k \cdot N + 1 \\ a^{r+1} &= k \cdot N \cdot a + a \\ a^{r+1} mod\ N &= a\ mod\ N, \end{aligned}$$

where k is some integer. So $F_N(x+r) = F(N_x)$, i.e., r is the period of $F_N(x)$. Note that $r \leq N$.

Three cases arise:

1. r is odd

2. r is even and $a^{r/2} mod\ N = -1$

3. r is even and $a^{r/2} mod\ N \neq -1$.

Cases 1) and 2) are not relevant to factorization of N, but in case 3) at least one of the two numbers $gcd(N, a^{r/2} \pm 1)$ is a non-trivial factor of N where $gcd(x, y)$ is the greatest common denominator of x and y (see Section 4 below).

How do we find $\text{ord}(a) = r$? The strategy is to calculation the function $F_N(x)$ for many values of x in parallel and to use Fourier techniques to detect the period in the sequence of function values. In the next subsection we show Shor's quantum algorithm does this efficiently using the quantum fourier transform.

3.2 Period finding

The algorithm uses two registers:

- register 1 (source) has K qubits and stores a number $Q = 2^K$, with $N^2 \leq Q \leq 2N^2$, or equivalently a number mod Q
- register 2 (target) has at least $n = log_2 N$ qubits, so can store N or more basis states, or equivalently, a number mod N.

Note that the total number of qubits required is then given by the sum of $K \leq 1 + 2log_2 N$ and $n \leq log_2 N$.

The algorithm can be decomposed into 6 steps.

1. Both registers are initialized in the state $|0\rangle \otimes |0\rangle$.
2. The source register is transformed to an equal superposition over all Q basis states. This can be done either by applying the K qubit Hadamard transform (see homework 3)

$$\begin{aligned} H^{\otimes K}|x\rangle &= \frac{1}{\sqrt{2^K}} \sum_y (-1)^{xy} |y\rangle \\ \Rightarrow H^{\otimes K}|0\rangle &= \frac{1}{\sqrt{2^K}} \sum_y |y\rangle \end{aligned}$$

or by applying the Fourier Transform

$$\begin{aligned} |q\rangle &\mapsto \frac{1}{\sqrt{Q}} \sum_{q'=0}^{Q-1} exp\left(2\pi i \frac{qq'}{Q}\right) |q'\rangle \\ \Rightarrow |0\rangle &\mapsto \frac{1}{\sqrt{Q}} \sum_{q'=0}^{Q-1} |q'\rangle . \end{aligned}$$

in both cases (what does this tell you about the relation of Hadamard to Fourier transform?) we get the full quantum state (of source and register)

$$\frac{1}{\sqrt{Q}} \sum_{q=0}^{Q-1} |q\rangle \otimes |0\rangle$$

3. Now we apply a gate U_a that implements the modular exponentiation $q \mapsto f(q) = a^q mod\, N$. This is a function that is easy to compute classically (it can be computed in $\log q$ multiplications using repeated squaring, $a^2 = a \times a$, $a^4 = a^2 \times a^2$, $a^8 = a^4 \times a^4$, ... see Nielsen and Chuang, p. 228 for a detailed analysis). As described above, $f(q)$ has r as its smallest period. Note that f is distinct on $[0, r-1]$ (i.e., all values are different) since otherwise it would have a smaller period.

   Applying the function f to the contents of source register 1 and storing the result in target register 2, i.e., we get

$$\frac{1}{\sqrt{Q}} \sum_{q=0}^{Q-1} |q\rangle \, |a^q mod\, N\rangle \, .$$

   Here $Q > N^2$ values of the function $f(q)$ are computed in parallel. Since $r < N$, the period r must manifest itself in the resulting sequence of function values now stored in the second register. So there can only be r different function values.

4. Now we measure the second register. When we measure, we must get some value which has to be one of the r distinct values of $f(q)$. Suppose it is $f(q_0)$. Then all superposed states of the first register inconsistent with this measured value must disappear. For simplicity, we shall restrict ourselves first to the case where $Q = mr$, i.e., there are m different values of q which have the same value of $f(q)$.

   Then exactly Q/r states of register 1 will contribute to the measured state of register 2, and after this measurement the combined state of the two registers must be given by

$$\frac{1}{\sqrt{Q/r}} \sum_{j=0}^{Q/r-1} |jr + q_0\rangle |f(q_0)\rangle$$

5. We now have a periodic superposition of state in register 1, with period r. From now on the second register is irrelevant and we can drop it from discussion. The first register has a periodic superposition whose period is the value that we wanted to compute in the first place. How do we get that period ?

   Can we get anything simply by measuring the first register ?No, since all we will get is a random point, with no correlation across independent trials (because q_0 is random). Instead, we first make a quantum Fourier transform on register 1.

   Applying the Fourier transform modulo Q to state

$$|\phi_{q_0}\rangle = \frac{1}{\sqrt{\frac{Q}{r}}} \sum_{j=0}^{\frac{Q}{r}-1} |jr + q_0\rangle$$

   gives us

$$\frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \omega^{kq_0} |k\frac{Q}{r}\rangle$$

   where ω is a primitive rth root of unity,

$$\omega = e^{\frac{2\pi i}{r}} \, .$$

You may be wondering how the sum got changed from Q/r terms to just r terms. This was the result of destructive interference in the QFT of the state $\left|\phi_{q_0}\right\rangle = \left|jr+q_0\right\rangle$. Here's how it happened. First rewrite $\left|\phi_{q_0}\right\rangle$ as a sum over all Q states:

$$\left|\phi_{q_0}\right\rangle = \sum_{a=0}^{Q-1} g(a)\left|a\right\rangle$$

where $g(a) = \sqrt{r/Q}$ if $a-q_0$ is a multiple of r and $g(a)=0$ otherwise. Then Fourier transforming this modulo Q (this just means the Fourier transform base K or with $Q=2^K$ basis states), gives

$$\begin{aligned} & \sum_c \frac{\sqrt{r}}{Q} \sum_{j=0}^{Q-1} exp\left(\frac{2\pi i(jr+q_0)c}{Q}\right)\left|c\right\rangle \\ = \; & \sum_c \frac{\sqrt{r}}{Q} \left[\sum_{j=0}^{Q-1} exp\left(\frac{2\pi i(jr)c}{Q}\right)\right] exp\left(\frac{2\pi i q_0 c}{Q}\right)\left|c\right\rangle. \end{aligned}$$

Now looking at the right hand side, you can see that when rc/Q is an integer, i.e., c is a multiple of Q/r, each term in the sum inside the square brackets will be equal to one. The square bracket term is then equal to Q and we obtain $exp\left(2\pi i q_0 c/Q\right)/\sqrt{r}$ for the coefficient of basis state $\left|c\right\rangle$. On the other hand, when rc/Q is not an integer, the sum in the square brackets cancels to zero (see Benenti p. 163 for an example). So the only states in the sum over c that survive are those for which c is a multiple of Q/r. Thus the Fourier transformed state has period Q/r, and furthermore it has non-zero values only at values of c that are multiples of this period. Writing $c = kQ/r$, we get then the QFT state

$$FT_Q\left|\phi_{q_0}\right\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} exp\left(\frac{2\pi i q_0 k}{r}\right)\left|k\frac{Q}{r}\right\rangle$$

which is what was given above. Note that the Fourier transform has moved the shift value q_0 in the index of the original state to a phase factor in the fourier transformed state.

6. Now we measure register 1. The measurement gives us a value $C = k\frac{Q}{r}$, where k is a random number between 0 and r-1. Now we have Q, C, and hence also the ratio $C/Q = k/r$. Now if $gcd(k,r)=1$, i.e., if k and r have no common divisors, we can reduce the ratio C/Q to an irreducible fraction, e.g., $1/r$. See Benenti p. 163 for an example. Since k is chosen at random in the measurement, then the probability that $gcd(k,r)=1$ is greater than $1/logr$ for larger values of r (see Appendix A.3 in Ekert and Jozsa, RMP 68, 733 (1996)). So one can repeat the is easy to see that with big probability $gcd(k,\frac{Q}{r})=1$.

   Then by repeating the calculation $O(logr) < O(log\mathbf{N})$ times, one can amplify the success probability to as close to one as desired. So we have an efficient determination of the order r.

In the general case, when $Q \neq nr$, one has a slightly modified analysis that results in the order being determined to a high probability. Note, in class we looked also at the slightly different procedure followed in Suter's book, where at step 4 one makes a Fourier Transform on register 1 and then measures this. See Suter Ch. 8.3.3.

4 Using order-finding to factor large numbers N efficiently

Once we have the order r of $a^x mod\ N$, we first check if r is even and $a^{r/2} mod\ N \neq -1$ (case 3) above). If so, then lets proceed with $y = a^{r/2}$. Since $y^2 mod\ N = 1$, then $y^2 - 1 = (y+1)(y-1)$ is divisible by N. So N has a common factor with either $y+1$ or $y-1$. The common factor must be one of greatest common divisors $gcd(N, y \pm 1)$. These can be efficiently computed with Euclid's algorithm (classical - what else!).

4.1 Euclid's algorithm for $gcd(x, y)$

Let x, y be 2 integers, $x > y$ and $z = gcd(z, y)$. Then both x and y and the numbers $x - y$, $x - 2y$, ... are multiples of z. Therefore the remainder $r = x - ky < y$ is also a multiple of z. Now if $r = 0$, then $z = y$ and the problem is solved. So we only have to figure out how to get to zero remainder from the starting integers x and y. This is easy. We simply repeatedly take the remainder:

$$z = gcd(x, y) = gcd(y, r_1) = gcd(r_1, r_2) = gcd(r_2, r_3) = ... = gcd(r_n, r_n),$$

where $r_1, r_2, ...$ are the successive remainders, $r_i = r_{i-1} - k_i y$. The last non-zero remainder r_n is z.

4.2 Shor's factoring algorithm

The overall quantum factoring algorithm is as follows:

1. If N even, return the factor 2
2. Determine whether $N = a^b$ for integers $a \geq 1$ and $b \geq 2$: if yes, return the factor a
3. Randomly choose y between 1 and $N - 1$. If $z = gcd(y, N) > 1$, return the factor z.
4. Use the order-finding algorithm to find the order r of $y\ mod\ N$, i.e., r such that $y^r mod\ N = 1$.
5. If r is even and $y^{r/2} mod\ N \neq -1$, then evaluate $gcd(y^{r/2} \pm 1, N)$. If one of these is a non-trivial factor (i.e., other than 1), return that value as a factor. If not, go back to step 3 and repeat.

The success rate of the last three steps must be reasonably high since this is a probabilistic algorithm. See discussions in the texts and in the paper of Ekert and Jozsa.