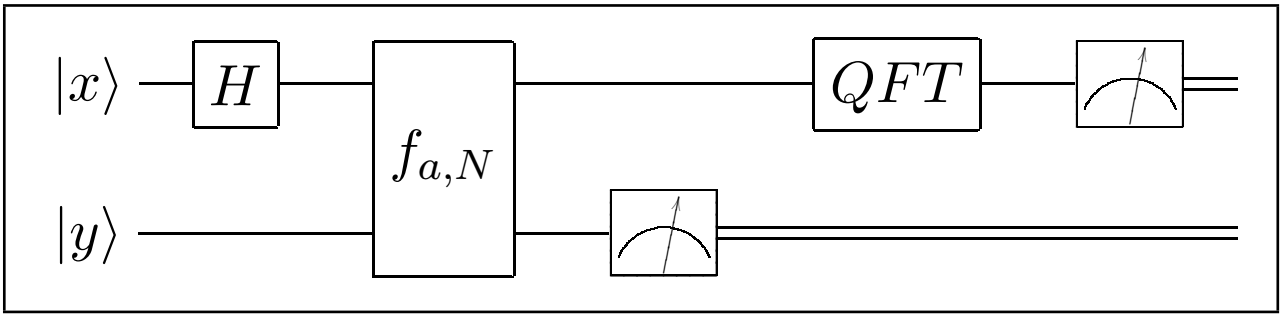
Shor's Algorighm

Denver Physics Study Group

msg2mw@gmail.com



Contents

1 Overview **3**

2 Notation **4**

3 Examples **4**

3.1 Y&M Example 6.5.6 4

3.2 Moorhouse Example 4

4 The Period Finding Circuit **5**

4.1 Step 1 - Input 5

4.2 Step 2 - Hadamard 5

4.3 Step 3 - The Modular Exponentiation Gate 6

4.4 Step 4 - Y Measurement 6

4.5 Step 5 - The QFT 7

4.6 Step 6 - Continued fractions and Convergents 9

5 The $f_{a,N}$ gate **10**

6 The QFT gate **11**

6.1 Constructing the QFT 11

6.2 What does the QFT operator do? 12

6.3 Comparing the QFT to the Hadamard 12

1 Overview

We have a large (odd) number N. We want to find a factor of N.

1. Randomly choose a number a such that $1 < a < N$.
2. Use Euclid's algorithm to determine GCD(a, N). If the GCD is not 1, then it's a factor of N and we're done.
3. Find r, the period of $f_{a,N}(x) = a^x \bmod N$. In other words find an $r > 1$ such that $a^r mod N = 1$.
   - *The Quantum Part*
   - Create a superposition of all the possible x values.
   - Use a quantum gate to set the y values to $f_{a,N}(x)$.
   - Measure the y value, leaving only one y with all the x values that produce it.
   - Put the x bits through a QFT gate.
   - Measure the x value.
   - *The Classical Part*
   - Run the continued fraction algorithm on the measured x value.
   - The last convergent denominator greater than N, or one of its multiples is, the desired period r.
4. If r is odd then go back to step 1 and try a new a.
5. Compute GCD$(a^{\frac{r}{2}} + 1, N)$ and GCD$(a^{\frac{r}{2}} - 1, N)$. Either one (or both) of these numbers is a factor of N, or else go back to step 1 and try a new a.

Note: There is apparently an efficient algorithm you can run first to determine if N is a power of a prime number. But I don't know what it is. Anybody?

2 Notation

I'm going to *attempt* to use the following variables consistensly. Feel free to point out when I forget to do that.

N - a number that we want to factor.
n - the number of bits required to represent N
n_x - the number of bits used for x (especially if it's not $2n$).
a - the number chosen at random between 1 and N.
r - the period that we are looking for.
D - the dimension of a problem, matrix, or set of bits. (eg. $D = 2^{2n}$ for the x bits.)
y_m - the measured value of the y bits (after $f_{a,N}$ in the quantum circuit).
x_m - the measured value of the x bits (the last step of the quantum circuit).

3 Examples

I'll be using (at least) these two examples in the notes that follow:

3.1 Y&M Example 6.5.6

This is Example 6.5.6 from the Yanofsky and Mannucci book, starting on page 210.

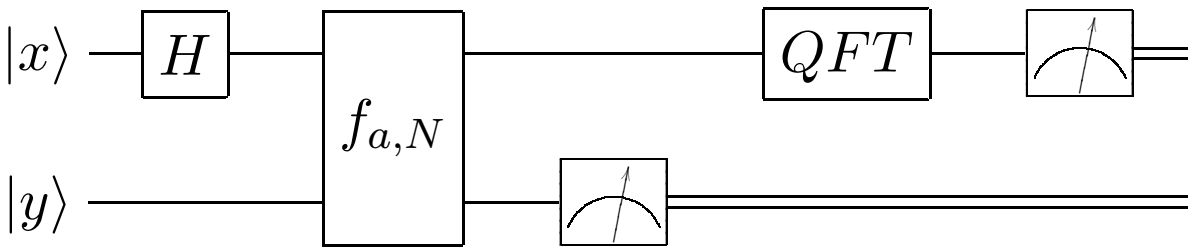
$N = 15$
$n = 4,\ n_x = 8$
$a = 13$
$r = 4$
$y_m = 7,\ x_m$ (not used?)

3.2 Moorhouse Example

This is an example used in a set of slides by G. Eric Moorhouse. They can be obtained from: http://ericmoorhouse.org/slides/talk2.pdf

$N = 55$
$n = 6,\ n_x = 13$ (instead of 12)
$a = 13$
$r = 20$
$y_m = 28,\ x_m = 4915$ (and we may also use 410)

4 The Period Finding Circuit



Quantum Circuit for Period Finding

We have a number N that we want to factor. We have chosen a number a as described above, and we want to find the period of a^x mod N. The cirucit above does not quite find that period, but it gives us a number from which we can find the period by using continued fractions.

4.1 Step 1 - Input

Suppose that n is the number of bits required to represent N. Then we need to use $2n$ bits for $|x\rangle$ and n bits for $|y\rangle$. Both $|x\rangle$ and $|y\rangle$ are set to all zeros.

So the total input state (the tensor product of $|x\rangle$ and $|y\rangle$ is simply $|0,0\rangle$.

4.2 Step 2 - Hadamard

We apply a Hadamard to the $|x\rangle$ bits. This puts $|x\rangle$ into a superposition of all the possible numbers that can be contained in $2n$ bits, all with equal probabilities. So $|x\rangle$ now looks like this:

$$\frac{1}{\sqrt{2^{2n}}}\big(\,|0\rangle + |1\rangle + |2\rangle + |3\rangle + ... + |2^{2n}-1\rangle\,\big).$$

And the total state at this point (including the $|y\rangle$ bits) is simply:

$$\frac{1}{\sqrt{2^{2n}}}\big(\,|0,0\rangle + |1,0\rangle + |2,0\rangle + |3,0\rangle + ... + |2^{2n}-1,0\rangle\,\big).$$

4.3 Step 3 - The Modular Exponentiation Gate

Now we apply the gate the does the modular exponentiation function: $a^x \bmod N$. In the circuit this is noted as $f_{a,N}(x)$. Or simply $f_{a,N}$.

What we're trying to do is to find the period of f which means that we need to find an $x > 0$ which gives a result of one. Classically we would have to try each x individually until we found the one we want.

But in this circuit we have all the possible values of x in a superposition in the $|x\rangle$ bits. So we only have to run the $f_{a,N}$ gate once in order to try them all.

Literally what the $f_{a,N}$ gate does is to exclusive-or the result of $a^x \bmod N$ with the $|y\rangle$ bits. But since we start out with $|y\rangle = 0$ we effectively simply set the $|y\rangle$ bits to $f(x)$. Note that the $f_{a,N}$ gate does not change the $|x\rangle$ bits.

After going through the $f_{a,N}$ gate, the total state is entangled. We no longer have separable $|x\rangle$ and $|y\rangle$ bits. Given that $f(x)$ is the result of running the gate on input x, then the total state after the gate is:

$$\frac{1}{\sqrt{2^{2n}}}\left(\,|0,f(0)\rangle + |1,f(1)\rangle + |2,f(2)\rangle + |3,f(3)\rangle + ... + |2^{2n}-1, f\left(2^{2n}-1\right)\rangle\,\right).$$

For the Y&M example we have:

$$\frac{1}{\sqrt{256}}\left(\,|0,1\rangle + |1,13\rangle + |2,4\rangle + |3,7\rangle + |4,1\rangle + |5,13\rangle + |6,4\rangle + ... + |255,7\rangle\,\right).$$

4.4 Step 4 - Y Measurement

Next we measure the $|y\rangle$ bits. Since the bits are entangled, once we do the y measurement the remaining $|x\rangle$ bits will be determined by the value we get.

In Y&M example, suppose we get a y value of 7. Then the resulting superposition will be:

$$\frac{1}{\sqrt{64}}\left(\,|3,7\rangle + |7,7\rangle + |11,7\rangle + ... + |255,7\rangle\,\right).$$

Note that, since only one out of four values survives, the (equal) probability of each part of the superposition has been divided by four.

4.5 Step 5 - The QFT

A Fourier transform can be thought of as moving from the time domain to the frequency domain. We don't exactly have a time axis here, but the values of x which have non-zero probabilities (the ones that match the measured value of y in the last step) are periodic.

Say for example that the period of the function is 4. Then we'll have a non-zero probability every four x values (where we match the measured y value). This means that there will be a total of (maximum-x-value / 4) spikes. So what we have is essentially a periodic function with a "frequency" of $r/2^{2n}$ (where r is the period).

The period is the inverse of the frequency. Also the transform detects not only the fundamental frequency but also "harmonics" (multiples of the fundamental frequency).

The bottom line is that, after the QFT gate, the $|x\rangle$ bits will have spikes at:

$$\frac{2^{2n}}{r} * m, \quad m = 0, 1, 2, ...$$

For example, in the Moorhouse problem we are using 13 $|x\rangle$ bits and $r = 20$. So we expect non-zero x values at muliples of:

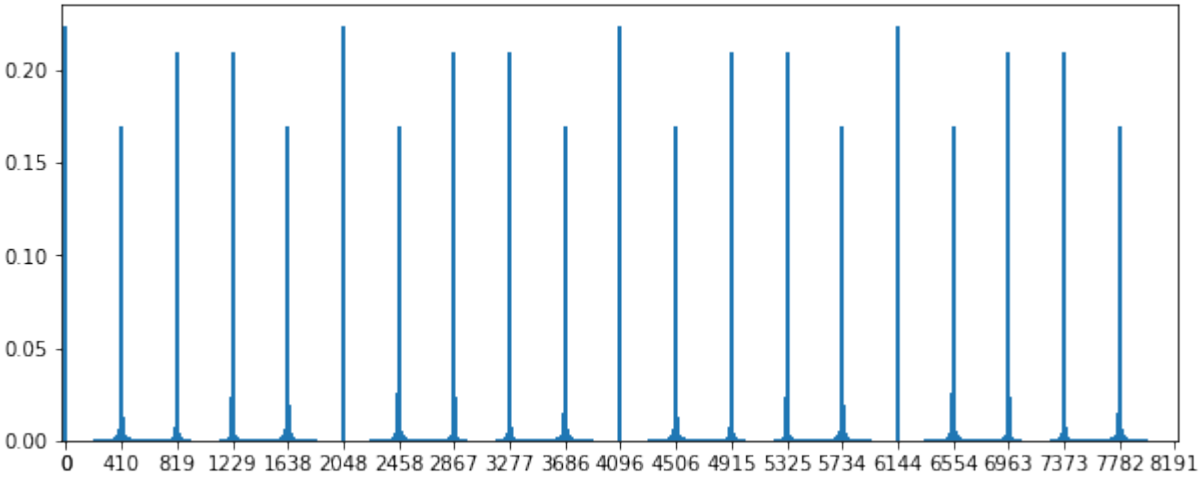
$$\frac{2^{2n}}{r} = \frac{2^{13}}{r} = \frac{8192}{20} = 409.6$$

But of course we don't have an x value of 409.6. So we are going to see spikes (roughly) at multiples of 410.

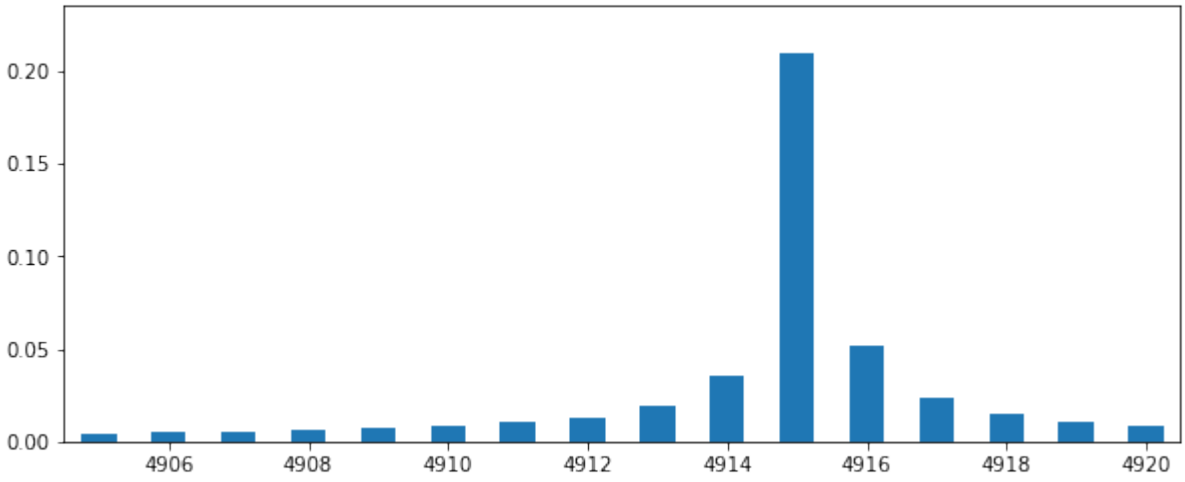
If we have a problem where r neatly divides the dimension of x then we may see neatly spaced non-zero x probabilities at exactly the expected locations. But in general we are only going to see an approximation of this, with *varying non-zero probabilities clustered around the expected values.*

The following two plots of the Moorhouse problem should help visualize this ...

Moorhouse Example: After the QFT gate



Close-Up of 4915



4.6 Step 6 - Continued fractions and Convergents

We have completed the quantum part of the calculation. The quantum circuit has finished and we are now doing a classical manipulation of that result.

We are factoring N. We've run the x bits through the QFT and then measured them. Given that xval is the value measured and xn is the number of x bits:

1. Use the measured x value as the numerator and 2 to the power of the number of x bits as the denominator. So we have the fraction $xval/2^{xn}$.
2. Carry out the continued fraction algorithm on that fraction.
3. Calculate the continued fractions convergents.
4. Take the last convergent that has a denominator smaller than N (the original number you were trying to factor).
5. The desired period r should be that denominator or some multiple of it.

Here's the Moorhouse problem with a measured x value of 20.

$$\frac{410}{8192} = 0 + \frac{1}{19 + \frac{1}{1+\frac{1}{50+\frac{1}{4}}}}$$

Convergents: $\left(0, \frac{1}{19}, \frac{1}{20}, \frac{51}{1019}, \frac{205}{4096}\right)$

The largest denominator $< N$ is 20 (which is the period we're looking for).

Note: To calculate the convergents what you do is to "cut off" the fraction at each + sign. So in the calculation above:

- The first convergent is just 0.
- The second one is: $0 + \frac{1}{19} = \frac{1}{19}$.
- The third is: $\frac{1}{19 + \frac{1}{1}} = \frac{1}{20}$.
- The fourth is: $\frac{1}{19 + \frac{1}{1+\frac{1}{50}}} = \frac{1}{19 + \frac{1}{\frac{51}{50}}} = \frac{1}{19 + \frac{50}{51}} = \frac{1}{\frac{969}{51} + \frac{50}{51}} = \frac{1}{\frac{1019}{51}} = \frac{51}{1019}$.
- And the last convergent always, well, converges to the original fraction.

5 The $f_{a,N}$ gate

(TBD)

6 The QFT gate

Note: In this section I'm using D for the dimension now, instead of N. This is to avoid confusion with N which represents the number we're factoring.

6.1 Constructing the QFT

This section shows how to construct a unitary matrix which does what's called "the quantum Fourier transform." Remember that when we have a unitary matrix we can use it to do a well-defined, invertible operation on a quantum state. In other words, we have (mathematically at least) a quantum gate. After we've constructed the operator then we'll talk about what the operation actually does and how we might use it.

(1) Let D be equal to the dimension of the desired matrix. In other words, $D = 2^n$ where n is the number of bits in play.

(2) Define $\omega_{_D}$ as:

$$\omega_{_D} = e^{2\pi i/D}$$

If you like you can think of ω_D as the Dth root of unity ...

(3) Set row j column k of the matrix to:

$$\text{QFT}_{jk} = \omega_{_D}^{jk}$$

We are using zero-based indexing here. So the first row is $j = 0$ and the first column is $k = 0$.

(4) Multiply the whole matrix by $\dfrac{1}{\sqrt{D}}$.

Example: For a QFT matrix that operates on 2 qbits:

$$(1)\ D = 4,\ (2)\ \omega_{_4} = e^{2\pi i/4},\ (3)\ \text{QFT} = \begin{bmatrix} \omega_{_4}^{(0)(0)} & \omega_{_4}^{(0)(1)} & \omega_{_4}^{(0)(2)} & \omega_{_4}^{(0)(3)} \\ \omega_{_4}^{(1)(0)} & \omega_{_4}^{(1)(1)} & \omega_{_4}^{(1)(2)} & \omega_{_4}^{(1)(3)} \\ \omega_{_4}^{(2)(0)} & \omega_{_4}^{(2)(1)} & \omega_{_4}^{(2)(2)} & \omega_{_4}^{(2)(3)} \\ \omega_{_4}^{(3)(0)} & \omega_{_4}^{(3)(1)} & \omega_{_4}^{(3)(2)} & \omega_{_4}^{(3)(3)} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}$$

$$(4)\ \text{QFT} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}$$

6.2 What does the QFT operator do?

If you take a state vector $|\psi\rangle$ and index its elements by j then operate on it with a QFT matrix, the jth element of the resulting vector will be:

$$|\phi\rangle = QFT|\psi\rangle$$

$$\phi_j = \frac{1}{\sqrt{D}} \sum_k e^{2\pi i j k/D} \psi_k$$

This is a discrete Fourier transform, from which the QFT gets its name. When we use the QFT in the period finding circuit we are essentially doing a DFT on the data in order to obtain a frequency, from which we can utlimately derive the period of the function.

6.3 Comparing the QFT to the Hadamard

As an additional note, it's interesting to compare the QFT to a Hadamard operator. The QFT is basically the same thing as a Hadamard except for the phases. This statement refers to both the QFT and Hadamard operators as well as the results of the operations. For example, both operators take a zero or one state vector and put it into an equal superpostion of all the possible values. Whereas the Hadamard only produces phases or + and -, the QFT produces phases cycling around the complex unit circle.

For example, comparing the 2-bit Hadamard to the 2-bit QFT we produced above:

$$HAD = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \quad QFT = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}$$

$$HAD\,|00\rangle = \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle \quad HAD\,|11\rangle = \frac{1}{2}|00\rangle - \frac{1}{2}|01\rangle - \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle$$

$$QFT\,|00\rangle = \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle \quad QFT\,|11\rangle = \frac{1}{2}|00\rangle - \frac{i}{2}|01\rangle - \frac{1}{2}|10\rangle + \frac{i}{2}|11\rangle$$

The QFT will actually produce the same result as the Hadamard when we only feed it zero bits. The difference only comes out when we operate on ones.

Here's a QFT on three bits:

$$QFT\,|0\rangle = \frac{1}{\sqrt{8}}|0\rangle + \frac{1}{\sqrt{8}}|1\rangle + \frac{1}{\sqrt{8}}|2\rangle + \frac{1}{\sqrt{8}}|3\rangle + \frac{1}{\sqrt{8}}|4\rangle + \frac{1}{\sqrt{8}}|5\rangle + \frac{1}{\sqrt{8}}|6\rangle + \frac{1}{\sqrt{8}}|7\rangle$$

$$QFT\,|1\rangle = \frac{1}{\sqrt{8}}|0\rangle + \frac{e^{7\pi i/4}}{\sqrt{8}}|1\rangle - \frac{i}{\sqrt{8}}|2\rangle + \frac{e^{21\pi i/4}}{\sqrt{8}}|3\rangle - \frac{1}{\sqrt{8}}|4\rangle + \frac{e^{35\pi i/4}}{\sqrt{8}}|5\rangle + \frac{i}{\sqrt{8}}|6\rangle + \frac{e^{49\pi i/4}}{\sqrt{8}}|7\rangle$$