Shor algorithm

QFT → "Phase estimation"

In the algorithm we only worked with real numbers. The QFT adds a phase to the problem. One can formulate the problem as follows.

You are given a state on n qubits as follows.

$$\frac{1}{\sqrt{2}^n}\sum_{k=0}^{2^n-1} e^{\frac{2i\pi}{2^n}kx}|k\rangle = \frac{1}{\sqrt{2}^n}\sum_{k=0}^{2^n-1}\left(e^{\frac{2i\pi}{2^n}x}\right)^k|k\rangle \qquad \left(\omega = e^{\frac{2i\pi}{2^n}x}\right)$$

$$= \frac{1}{\sqrt{2}^n}\sum_{k=0}^{2^n-1}\omega^k|k\rangle$$

Can you get the value of x, seen as a number between 0 and 2^n-1 ?

The answer is yes: we need a circuit computing

$$\frac{1}{\sqrt{2}^n}\sum_{k=0}^{2^n-1} e^{\frac{2i\pi}{2^n}kx}|k\rangle \mapsto |x\rangle$$ (beware: there will be re-indexing of the register x)

Let us try with $n=1$ and $x=0$ ou 1. Then

$$\sum_{k=0}^{1} e^{\frac{2i\pi}{2}kx}|k\rangle = |0\rangle + (-1)^x|1\rangle$$

and we want to get back $|x\rangle$... Hadamard is enough.

Let us try with $n=2$ and $x = 0,\ 1,\ 2$ or 3 : $\lfloor x\rfloor_2 = x_1x_2$, i.e. $x = 2x_1+x_2$

We have

$$\sum_{k=0}^{2^n-1} e^{\frac{2i\pi}{2^n}kx}|k\rangle = \sum_{k=0}^{3} e^{2i\pi\left(\frac{x_1}{2}+\frac{x_2}{4}\right)k}|k\rangle =$$

$$\sum_{k=0}^{3} e^{2i\pi\frac{x_1}{2}k}e^{2i\pi\frac{x_2}{4}k}|k\rangle = \sum_{k=0}^{3} e^{i\pi x_1 k}e^{i\pi\frac{x_2}{2}k}|k\rangle$$

$$=$$

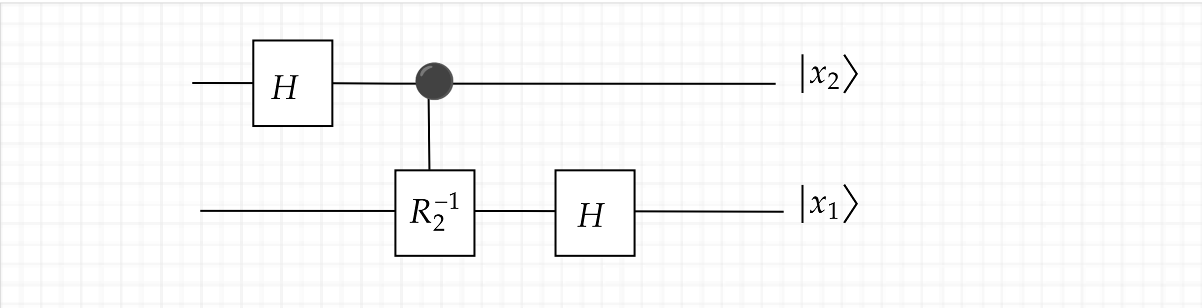
$$\frac{1}{2}(|00\rangle + e^{i\pi x_1}e^{i\pi\frac{x_2}{2}}|01\rangle + e^{2i\pi x_1}e^{i\pi x_2}|10\rangle +$$

$$e^{3i\pi x_1}e^{3i\pi\frac{x_2}{2}}|11\rangle)$$

$$=$$

$$\frac{1}{2}\left(|00\rangle + e^{i\pi x_1} e^{i\pi\frac{x_2}{2}}|01\rangle + e^{i\pi x_2}|10\rangle + e^{i\pi x_1} e^{3i\pi\frac{x_2}{2}}|11\rangle\right)$$
$$=$$
$$\frac{1}{2}\left(|0\rangle + e^{i\pi x_2}|1\rangle\right) \otimes \left(|0\rangle + e^{i\pi x_1} e^{i\pi\frac{x_2}{2}}|1\rangle\right)$$
$$=$$
$$\frac{1}{2}\left(|0\rangle + (-1)^{x_2}|1\rangle\right) \otimes \left(|0\rangle + (-1)^{x_1} e^{i\pi\frac{x_2}{2}}|1\rangle\right)$$

so to get back $|x_2 x_1\rangle$ (!!! BEWARE OF THE RE-INDEXING !!!)
the circuit is as follows



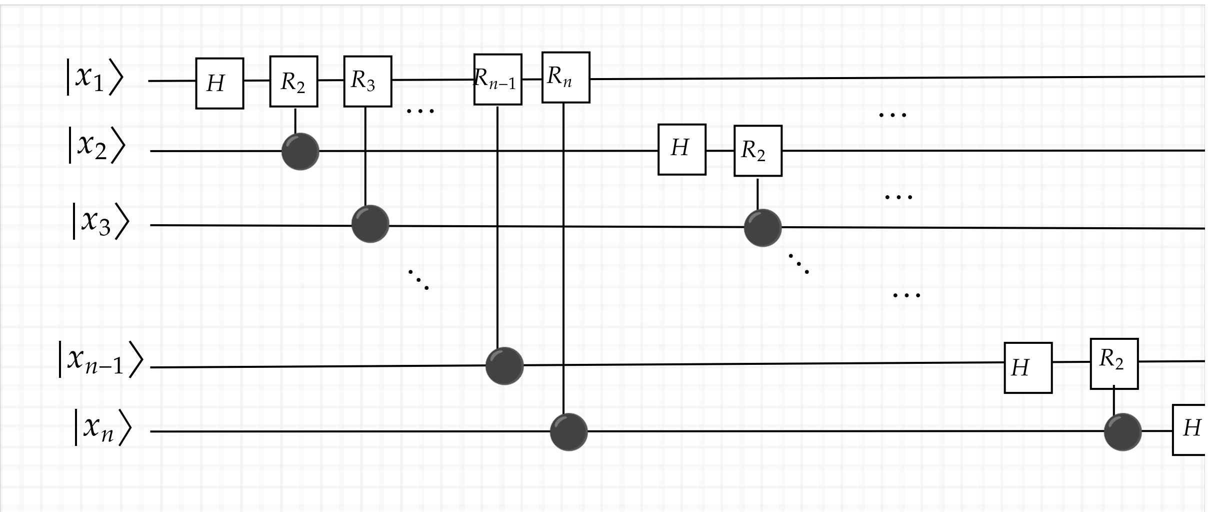
with $R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2i\pi/2^k} \end{pmatrix}$

This generalizes to n qubits, and the circuit is call "QFT inverse" (for "Quantum Fourier Transform"). The name comes from the fact that in the other direction, we compute

$|x\rangle \mapsto \sum_{k=0}^{2^n-1} e^{\frac{2i\pi}{2^n}kx}|k\rangle$ (modulo the x reindexing)

which is very close to a discrete Fourier transform.

The QFT circuit looks like this :



with

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2i\pi/2^k} \end{pmatrix}$$

Quantum Phase Estimation (QPE)

An very useful algorithm makes it possible to find the eigenvalues of a unitary map U. For such a map the eigenvalues are necessary of the form $e^{2i\pi\omega}$, and, without loss of generality, ω is a real number between 0 and 1.

Recall:
If $U|\psi\rangle = \lambda|\psi\rangle$ we say that $|\psi\rangle$ is an **eigenvector** of U and λ an **eigenvalue**
In the case where U is unitary, λ is of the form $e^{2i\pi\omega}$ because U preserves the norm... so $|\lambda| = 1$.

The algorithm QPE ("Quantum Phase Estimation") makes it possible to find ω.

To understand how this work, let us consider ω set to $0.x_1x_2$ in binary form.
So $\omega = \dfrac{x_1}{2} + \dfrac{x_2}{4}$.
Let $|\psi\rangle$ be the corresponding eigenvector.

We then have $U|\psi\rangle = e^{2i\pi\omega}|\psi\rangle$.

Computing C-U :

$$\frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big) \otimes |\psi\rangle$$
$$=$$
$$\frac{1}{\sqrt{2}}\big(|0\rangle \otimes |\psi\rangle + |1\rangle \otimes |\psi\rangle\big)$$

$$\xrightarrow{C-U}$$

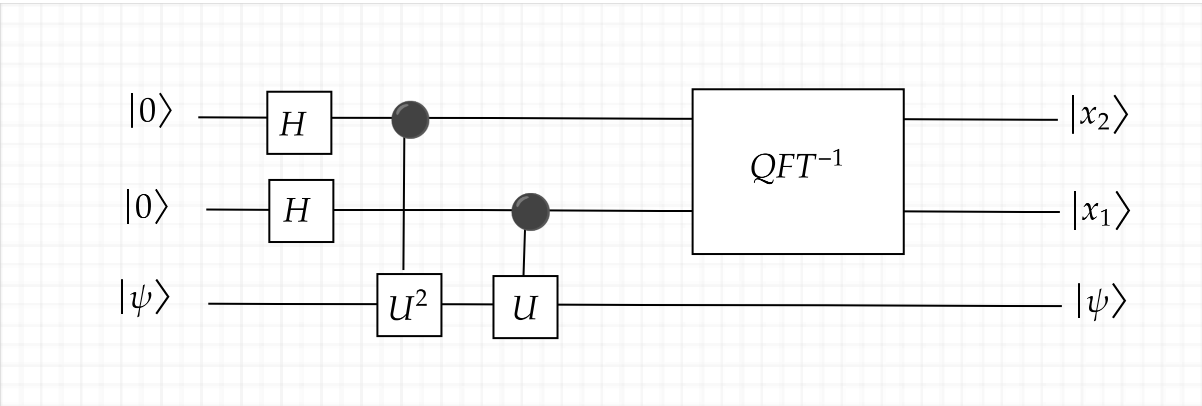
$$\frac{1}{\sqrt{2}}\Big(|0\rangle \otimes |\psi\rangle + |1\rangle \otimes \big(U|\psi\rangle\big)\Big)$$
$$=$$
$$\frac{1}{\sqrt{2}}\Big(|0\rangle \otimes |\psi\rangle + |1\rangle \otimes \big(e^{2i\pi\omega}|\psi\rangle\big)\Big)$$
$$=$$

$$\frac{1}{\sqrt{2}}\left(|0\rangle \otimes |\psi\rangle + e^{2i\pi\omega}|1\rangle \otimes |\psi\rangle\right)$$
$$=$$
$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2i\pi\omega}|1\rangle\right) \otimes |\psi\rangle$$

So

$$C-U : \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) \otimes |\psi\rangle$$
$$\mapsto \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2i\pi\left(\frac{x_1}{2}+\frac{x_2}{4}\right)}|1\rangle\right) \otimes |\psi\rangle$$
$$(C-U)^2 : \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) \otimes |\psi\rangle$$
$$\mapsto \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2i\pi\left(\frac{x_2}{2}\right)}|1\rangle\right) \otimes |\psi\rangle$$

By what we just saw with the inverse QFT:



(Note how the indices got swapped)

Once again, this generalizes. One can show that this is also working (albeit probabilistically) if ω is not writable on precisely n bits.

Typical use-case: Shor's algorithm

Problem: **FACTORIZATION**

Input : N a composite number: it admits a non-trivial factor.
Output : a divisor of N

Problem: **ORDER-FINDING**

Input : two integers N and a, co-primes
Output : The period r of a, i.e. the smallest $r > 0$ such that $a^r \equiv 1 \bmod N$

Problem: **PHASE-ESTIMATION**
Input : A unitary U and an eigenvector $|\phi\rangle$
Output : the corresponding eigenvalue

We can reduce FACTORIZATION to ORDER-FINDING, that can itself be reduced to PHASE-ESTIMATION...

1st step : reduction of FACTORIZATION to ORDER-FINDING.

Meaning: "If I know how to solve ORDER-FINDING I can easily factor a N"

It is purely a math problem.

So, suppose that I can solve ORDER-FINDING. I am given N to factor.

The idea is to realize that if N is the product of two co-prime integers greater than 2, one can derive from the chinese remainder theorem the existence of at least 4 numbers b such that $b^2 \equiv 1 \bmod N$. There is then at least one such b distinct from 1 and -1 (mod N) such that $b^2 - 1 \equiv 0 \bmod N$, i.e. such that N divides $(b-1)(b+1)$.
$\rightarrow b \not\equiv 1 \bmod N$ so $b - 1 \neq 0 \bmod N$ so N does not devide $b - 1$
$\rightarrow b \not\equiv -1 \bmod N$ so N does not devide $b + 1$
So $\gcd(N, b+1)$ and $\gcd(N, b-1)$ are non-trivial: we got factors of N (which are furthermore efficiently computable!)

An algorithm for FACTORIZATION then ultimately consists in efficiently finding such a b.
Shor's algorithm proceeds as follows.
1) We randomly select a number $1 < a < N$. If it is not co-prime with N, we are done: we have a non-trivial factor.
2) Otherwise, it is co-prime with N: we then invoke our algorithm for ORDER-FINDING: it outputs a a number r such that $a^r = 1$ mod N. Because of maths properties, the number r is odd or even with probability $\frac{1}{2}$. We want an even r: we start over to step 1 until we get one.
3) We now have an even r : $r = 2{*}r'$. So $\left(a^{r'}\right)^2 \equiv 1 \bmod N$. (Note : This is the number b we were looking for!)

So $\left(a^{r'}\right)^2 - 1 = 0$ mod N

So $\left(a^{r'} - 1\right)\left(a^{r'} + 1\right) = 0$ mod N and then N divides $\left(a^{r'} - 1\right)\left(a^{r'} + 1\right)$.

For density reasons, and invoking the chinese remainder theorem, we can assume

that $a^{r'}$ is neither 1 nor -1 modulo N.
4) A factor of N is then for instance $\gcd\left(N, a^{r'} - 1\right)$.

2nd step : solving ORDER-FINDING using PHASE-ESTIMATION (and then using a quantum co-processor !)

Note that if a and N are co-primes, then $x \mapsto a * x \bmod N$ is a reversible function (it is a permutation of $\{0... N-1\}$).

The unitary we can choose is simply $U_a : |x\rangle \mapsto |a * x \bmod N\rangle$ (multiplucation modulo N)

Beware: this does not say how to implement it efficiently... One possibility is to use the technique we saw with V_f's and U_f's, but there are better alternatives, see e.g. https://arxiv.org/abs/quant-ph/0205095.

What is an eigenvector for U_a ?

Let us try with successive approximations.
Start with $|1\rangle_n = |0...01\rangle$ (the binary encoding of 1 on n qubits)
If we apply U_a : we get $|a\rangle$
What about $\frac{1}{\sqrt{2}}\left(|1\rangle + |a \bmod N\rangle\right)$:

applying U_a, we get $\frac{1}{\sqrt{2}}\left(|a \bmod N\rangle + |a^2 \bmod N\rangle\right)$

What about $\frac{1}{\sqrt{3}}\left(|1\rangle + |a \bmod N\rangle + |a^2 \bmod N\rangle\right)$:

applying U_a, we get $\frac{1}{\sqrt{3}}\left(|a \bmod N\rangle + |a^2 \bmod N\rangle + |a^3 \bmod N\rangle\right)$

... We can continue like that. Eventually, the power of a will reach the order r, and $|a^r \bmod N\rangle$ is then $|1\rangle$.

So, summarizing:

$$\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|a^k \bmod N\rangle \xrightarrow{U_a} \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|a^{(k+1)} \bmod N\rangle = \frac{1}{\sqrt{r}}\sum_{k=1}^{r}|a^k \bmod N\rangle$$

$$= \frac{1}{\sqrt{r}}\left(|a^r \bmod N\rangle + \sum_{k=1}^{r-1}|a^k \bmod N\rangle\right)$$

$$= \frac{1}{\sqrt{r}}\left(|1 \bmod N\rangle + \sum_{k=1}^{r-1}|a^k \bmod N\rangle\right)$$

$$= \frac{1}{\sqrt{r}}\left(|a^0 \bmod N\rangle + \sum_{k=1}^{r-1}|a^k \bmod N\rangle\right)$$

$$\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|a^k \bmod N\rangle$$

And we have an eigenvector of U_a, with eigenvalue 1.

Can we find more of them ?

Using the same technique, we can define (indexed with s)

$$|\phi_s\rangle = \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{-2i\pi\frac{s*k}{r}}|a^k \bmod N\rangle$$

Let's compute:

$$U_a|\phi_s\rangle = \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{-2i\pi\frac{s*k}{r}} U_a|a^k \bmod N\rangle$$

$$=$$

$$\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{-2i\pi\frac{s*k}{r}}|a^{(k+1)} \bmod N\rangle$$

$$=$$

$$\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{-2i\pi\frac{s*(k-1)}{r}}|a^k \bmod N\rangle$$

$$=$$

$$=$$

$$\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{2i\pi\frac{s}{r}} e^{-2i\pi\frac{s*k}{r}}|a^k \bmod N\rangle$$

$$=$$

$$e^{2i\pi\frac{s}{r}}|\phi_s\rangle$$

This eigenvalue is nice, since it contains a phase parametrized by r... As QPE gives us the phase, with some luck we can retrieve r out.

U_a have r such eigenvectors, one for each value of s between 0 and $r-1$.

Is it over ? We would just have to use the fact that

$$|0...0\rangle \otimes |\phi_s\rangle \xrightarrow{QPE(U_a)} |x_1 ... x_n\rangle \otimes |\phi_s\rangle$$

where $x_1 ... x_n$ is a binary representation of the phase of the eigenvalue corresponding to $|\phi_s\rangle$, from which ---maybe--- with some processing one can infer r.

The problem is that one cannot directly use these eigenvectors, since we would need to know r.

However, what we can do is use the fact that the QPE is a linear map, so we can place them in superposition:

$$|0...0\rangle \otimes \left(\alpha|\phi_s\rangle + \beta|\phi_{s'}\rangle\right) \xrightarrow{QPE(U_a)}$$
$$\alpha\,|x_1 ... x_n\rangle \otimes |\phi_s\rangle \;+\; \beta\,|y_1 ... y_n\rangle \otimes |\phi_{s'}\rangle$$

(where $y_1 ... y_n$ is a binary representation of the phase of the eigenvalue corresponding to $|\phi_{s'}\rangle$)

The trick consists in realizing that if we place them all in (equal) superposition, as follows:

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|\phi_s\rangle = \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}e^{-2i\pi\frac{s*k}{r}}|a^k \bmod N\rangle$$
$$=$$
$$\frac{1}{r}\sum_{k=0}^{r-1}\sum_{s=0}^{r-1}e^{-2i\pi\frac{s*k}{r}}|a^k \bmod N\rangle$$
$$=$$
$$\frac{1}{r}\sum_{k=0}^{r-1}\left(\sum_{s=0}^{r-1}e^{-2i\pi\frac{s*k}{r}}\right)|a^k \bmod N\rangle \quad (***)$$

The inner (red) sum is equal to

- r if $k = 0$
- 0 otherwise (because it is a sum of all of the roots of unity.

Therefore, in the sum (***) all the terms are nul except when $k = 0$: we get

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|\phi_s\rangle \;=\; \frac{1}{r}\left(r|a^0 \bmod N\rangle\right) \;=\; |1\rangle_n$$ (the encoding of 1 on n qubits: $|0...01\rangle$)

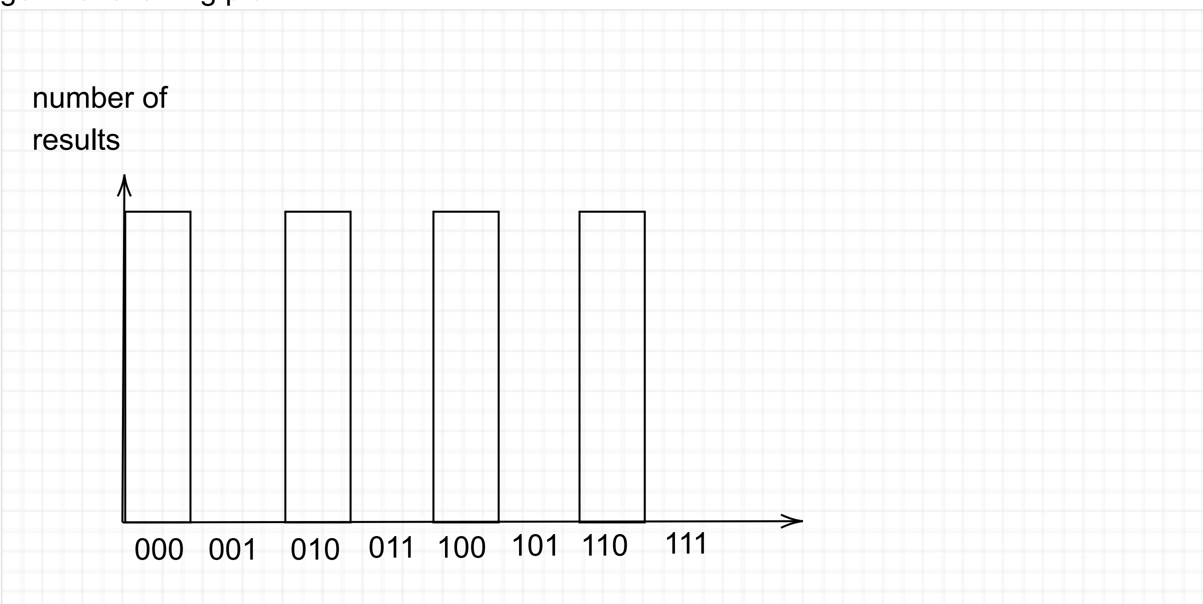
If we run QPE(U_a) on this input, we "compute" all of the phases at once. Consider two registers, the first one for retrieving the ω of the eigenvalue and the second one for the eigenvector. So

$QPE(U_a)\big(|0...0\rangle \otimes |\phi_s\rangle\big) = |s/r\rangle \otimes |\phi_s\rangle$
where by $|s/r\rangle$ we mean the approximation of s/r over the corresponding number of qubits.

Then

$$QPE(U_a)\left(|0...0\rangle \otimes \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|\phi_s\rangle\right) =$$
$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1} QPE(U_a)\big(|0...0\rangle \otimes |\phi_s\rangle\big) =$$
$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1} |s/r\rangle \otimes |\phi_s\rangle$$

→ measuring the first register, we get one of the $\frac{s}{r}$ (for the sake of the discussion, assume that the decomposition on n bits is exact)
For instance, if $r = 4$, there are exactly 4 elements in the sum: measuring, we get $0/4$, $1/4$, $2/4$ and $3/4$.
→ On 2 bits, this is 00, 01, 10 and 11.
→ On 3 bits, this is 000, 010, 100 and 110 (since $0.x_1x_2x_3 = \frac{x_1}{2} + \frac{x_2}{4} + \frac{x_3}{8}$)
If we were to perform many measurements (for 3 bits) and collecting the results, we would get the following plot



with equiprobable results 000, 010, 100 and 110.

If the decomposition were not exact (for instance when $r = 3$), we would instead get a less precise plot with 3 peaks but not as sharp. Possibly then 3 bits would not be enough to distinguish them, and we would need to get to 5 or 6 bits of precision.

In any case, when the precision is high enough, one can "read out" the period r of $a \bmod N$ from the plot.

→ this is what we shall do in the lab session !

However, to conclude, how to get out the r out of an estimate of s/r for some s ? This can be done with the algorithm of continued fractions. See e.g. https://en.wikipedia.org/wiki/Continued_fraction#Best_rational_approximations