Phase estimation, simulation and matrix inversion

September 23, 2018

Even though quantum algorithms have been an area of active and very intense research for more than twenty years, it is probably valid to say that most quantum algorithms can be considered as variations over a few central themes. One of these themes is the **quantum phase estimation**, a procedure introduced in the mid nineties of the last century, that is designed to discover eigenvalues and eigenstates of unitary operators. Since its first appearance in [7], this procedure has not only been employed in some groundbreaking algorithms like quantum matrix inversion and quantum machine learning, but has also turned out to be related to existing algorithms like Shor's famous algorithm for factoring large integers. In these notes, we describe the quantum phase estimation algorithm and explore some of its applications.

1 Quantum Phase estimation

In its simplest form, the **quantum phase estimation algorithm** is concerned with the following problem ([3]). We are given a unitary operator U acting on n qubits, and suppose that $|\psi\rangle$ is an eigenvector of U. As U is unitary, we know that the modulus of the eigenvalue is one, i.e. the eigenvalue is of the form $e^{i2\pi\Phi}$ with some $0 \leq \Phi < 1$. In other words, the eigenvalue is a pure phase factor. The objective of the algorithm is to estimate the phase, i.e. to find an approximation Φ.

To do this, we need to add m additional qubits to our quantum register, where m is a number chosen arbitrarily - we will see later that m determines the precision of the result. We will write the combined states as tensor products

$$|a\rangle|b\rangle$$

where $|a\rangle$ lives in the space spanned by the m ancillary qubits and $|b\rangle$ is the primary n-qubit quantum register. We then prepare the system in the state

$$\frac{1}{\sqrt{2}^m}\sum_{k=0}^{2^m-1} e^{2i\pi k\Phi}|k\rangle|\psi\rangle = \frac{1}{\sqrt{2}^m}\sum_{k=0}^{2^m-1}|k\rangle U^k|\psi\rangle \tag{1}$$

Before we proceed, let us see how this can be implemented as a quantum circuit. Consider the combination of gates shown in figure 1.

To see why this gate produces the state that we need, let us first analyse the first (least significant) qubit of the ancillary register. Ignoring all other qubits for a moment, the state after applying the Hadamard gate will be

$$\frac{1}{\sqrt{2}}\left[|0\rangle|\psi\rangle + |1\rangle|\psi\rangle\right]$$

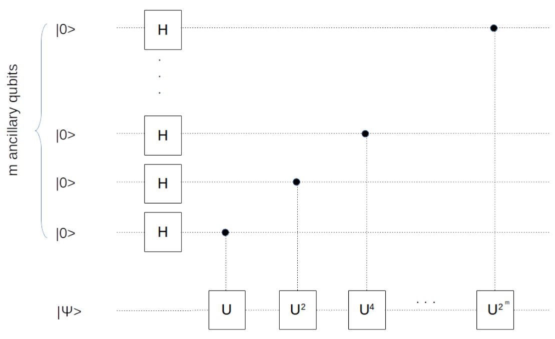

Figure 1: Quantum phase estimation circuit

Applying the controlled-U operation and observing that the state $|\psi\rangle$ is an eigenstate for U, we find that the state after the controlled-U gate is

$$\frac{1}{\sqrt{2}}\left[|0\rangle|\psi\rangle + e^{2i\pi\Phi}|1\rangle|\psi\rangle\right]$$

Doing the same exercise for the second ancillary qubit yields

$$\frac{1}{\sqrt{2}}\left[|0\rangle|\psi\rangle + e^{2i\pi 2^1\Phi}|1\rangle|\psi\rangle\right]$$

and so forth. Thus after applying the entire circuit, the overall state is

$$\frac{1}{\sqrt{2}^m}\prod_{j=0}^{m-1}(|0\rangle + e^{2i\pi 2^j\Phi}|1\rangle)|\psi\rangle$$

If we multiply this out, every possible m-bit string will appear exactly one, and we obtain the state (1).

Now let us examine this superposition in a bit more detail. To do this, let us first consider the special case that the number Φ is an exact multiple of 2^{-m}, i.e.

$$\Phi = \frac{t}{N}$$

where $N = 2^m$. Then our state can be written as

$$\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} e^{2i\pi\frac{kt}{N}}|k\rangle|\psi\rangle$$

But this looks familiar - it is just the inverse quantum Fourier transform applied to the state $|t\rangle|\psi\rangle$. This implies that we can recover the number t and with it the phase by applying the Fourier transform to our superposition. In other words, for the special case that Φ can be written exactly as a fraction with m bits, the following algorithm - the **quantum phase estimation algorithm** will work.

1. Start with the state $|0\rangle|\psi\rangle$ where $|\psi\rangle$ is an aigenstate of U
2. Apply the circuit shown in figure 1 to build the state (1)
3. Apply an quantum Fourier transform
4. Measure the first register and call the result t.
5. Return $e^{2\pi i \frac{t}{N}}$

Thus, in this special case, the entire transformation - applying the circuit in figure 1 followed by the quantum Fourier transform - performs the mapping

$$|0\rangle\psi\rangle \mapsto |N\Phi\rangle|\psi\rangle$$

and thus extracts the phase Φ.

Fortunately, it turns out that this is still approximateyl true if Φ is not an exact multiple of 2^{-m}. To see why this is the case, let us calculate, in the general case, the result of applying the inverse quantum Fourier transform to the state (1). Recall that the quantum Fourier transform acts as follows.

$$|k\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{s=0}^{N-1} e^{2\pi i \frac{-sk}{N}} |s\rangle$$

Therefore, after applying the quantum Fourier transform, our quantum registers will be in the following state

$$\frac{1}{2^m} \sum_{s,k=0}^{N-1} e^{2\pi i k(\Phi - \frac{s}{N})} |s\rangle|\psi\rangle$$

Now let t the the closest m-bit approximation to the actual phase Φ. In other words, let us write

$$\Phi = \frac{t}{N} + \delta$$

with $|\delta| < \frac{1}{2N}$. By passing from Φ to $1 - \Phi$ if needed (which describes the same eigenvalue), we can also assume that $0 \leq \delta$. With that, we can write our state as

$$\frac{1}{2^m} \sum_{s,k=0}^{N-1} e^{2\pi i \frac{k}{N}((t-s)+\delta N)} |s\rangle|\psi\rangle$$

The sum over s can actually be considered as a sum over the group $\mathbb{Z}/N\mathbb{Z}$, as the arguments are periodic with period N. Now any element of this group can be obtained as the residue of $t - l$ with exactly one $-\frac{N}{2} \leq l < \frac{N}{2}$. Therefore our state can as well be written as

$$\sum_{-\frac{N}{2} \leq l < \frac{N}{2}} \sum_{k=0}^{N-1} \frac{1}{2^m} e^{2\pi i \frac{k}{N}(l+\delta N)} |t - l \operatorname{mod} N\rangle|\psi\rangle = \sum_{-\frac{N}{2} \leq l < \frac{N}{2}} a_l |t - l \operatorname{mod} N\rangle|\psi\rangle$$

where the coefficient a_l is given by

$$a_l = \sum_{k=0}^{N-1} \frac{1}{2^m} e^{2\pi i \frac{k}{N}(l+\delta N)}$$

or, using the sum formula for a geometric series and the fact that the term $2\pi il$ drops out due to the periodicity

$$a_l = \frac{1}{N} \frac{1 - e^{2\pi i \delta N}}{1 - e^{2\pi i(\frac{l}{N}+\delta)}}$$

Intuitively, we expect that, as a function of l, this has a very sharp peak at $l = 0$, so that the probability to measure a value that differs significantly from t is very low. In fact, one can show (see [3] and appendix B) that given some $\epsilon > 0$ and some $\Delta > 0$, we can always make sure that the probability to measure a value that deviates by more than Δ from t is less than ϵ if we choose m sufficiently large. In this sense, the number m of ancillary qubits that we use is driving the precision of the algorithm.

Let us visualize the behavior of the amplitudes a_l for different values of N. We know (see appendix C) that we have the relation

$$|1 - e^{i\Phi}| = 2 \sin \frac{\Phi}{2}$$

which we can use to write

$$|a_l|^2 = \frac{1}{N^2} \frac{\sin^2 \pi\delta N}{\sin^2 \pi(\frac{l}{N} + \delta)}$$

which is of course only valid if $\delta \neq 0$, as otherwise the sum formula for the geometric series that we have used does not apply.

In diagram 2, we have plotted the amplitudes a_l for different values of N and $d = \frac{1}{2N}$ (which is the worst case as the actual value of Φ is exactly between two m-bit numbers in this case). The x-axis shows the absolute error that we make, i.e. $\frac{|s-t|}{N}$. The y-axis shows the probability for this error.

The upper image shows $N = 8$, corresponding to three ancillary qubits. We see a clear peak at $l = 0$, but $l = -1$ has the same probability to be measured, and there is still some substantial amplitude at -0.2. However, this changes quickly if we increase the number of ancillary qubits. The second diagram shows the results for the same choice $d = \frac{1}{2N}$, but this time we are using eight qubits and therefore $N = 256$. We see that the peak has become extremely sharp, and the probability to make an error of -0.2 is now virtually zero. Thus increasing the number of ancillary qubits leads very quickly to a high probability to obtain a very accurate result.

2 Finding eigenvectors

In the previous section, we have seen how the quantum phase estimation algorithm can be used to determine the eigenvalue of a given eigenstate $|\psi\rangle$. However, quite often we are not able to prepare the system in a pure eigenstate and are looking for a way to find both an eigenstate and the corresponding

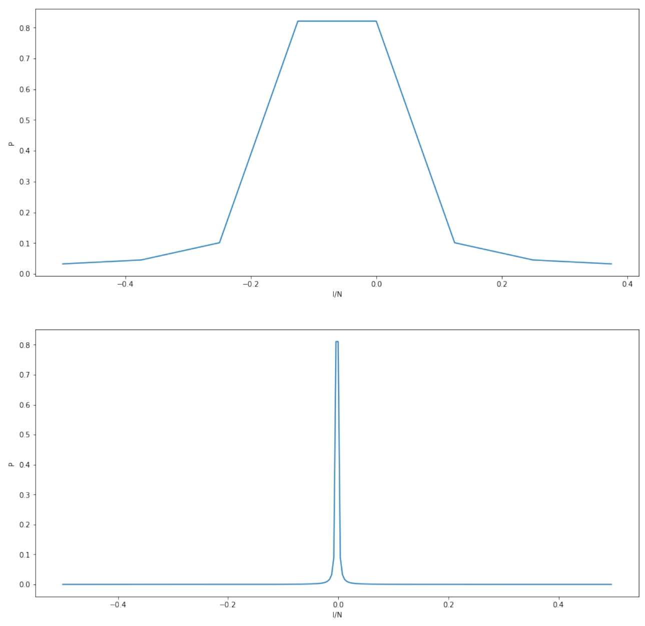

Figure 2: Amplitudes a_l for different values of N

eigenvalue. It turns out that the QPE algorithm does that for us without any modifications.

In fact, let us now assume that we are given an arbitrary state $|\psi\rangle$. We do not know whether $|\psi\rangle$ is an eigenstate, but what we know by general linear algebra is that we can write $|\psi\rangle$ as a linear combination

$$|\psi\rangle = \sum_j c_j |\psi_j\rangle$$

where each $|\psi_j\rangle$ is an normalized eigenstate with eigenvalue $e^{2\pi i \Phi_j}$, $c_j \in \mathbb{R}^+$ and the eigenvalues are all distinct. In particular, the $|\psi_j\rangle$ are orthonormal (but not necessarily a basis, as some eigenspaces might be degenerate). In fact, the $c_j|\psi_j\rangle$ are simply the non-zero projections of $|\psi\rangle$ onto the eigenstates of U.

What happens to such a state when we apply the quantum phase estimation algorithm? Both the circuit shown in diagram 1 as well as the inverse quantum Fourier transform are of course linear operations. If Φ_j is the phase corresponding to the eigenstate $|\psi_j\rangle$ and we again choose best approximations

$$\Phi_j = \frac{t_j}{N} + \delta_j$$

then, by linearity, we can write the state after applying the Fourier transform as

$$\frac{1}{2^m} \sum_j \sum_{s=0}^{N-1} \Big[\sum_{k=0}^{N-1} c_j e^{2\pi i \frac{k}{N}((t_j - s) + \delta N)} \Big] |s\rangle |\psi_j\rangle$$

or

$$\sum_j \sum_{s=0}^{N-1} c_j a_{j,s} |s\rangle |\psi_j\rangle$$

where now

$$a_{j,s} = \frac{1}{2^m} \sum_{k=0}^{N-1} e^{2\pi i \frac{k}{N}((t_j - s) + \delta N)}$$

But we already know that for sufficiently large N, the sum in this expression will have a very sharp peak around $s = t_j$. Thus, only those combinations of j and s for which $s \approx t_j$ will have a substantial amplitude, and this amplitude will be close to c_j.

If we now measure the first register only, the system will end up in a state which is very close to one of the states $|t_j\rangle|\psi_j\rangle$. Thus, after the measurement, the second register will aproximately be in one of the eigenstates $|\psi_j\rangle$, and the probability to obtain a specific $|\psi_j\rangle$ in this way is $|c_j|^2$.

Let us now try to make this intuition a bit more precise. After applying the Fourier transform, our state is

$$\sum_j \sum_{s=0}^{N-1} c_j a_{j,s} |s\rangle |\psi_j\rangle$$

The amplitudes depend on the two degrees of freedom described by s and j. Let us now choose some large values of d and N such that $\frac{d}{N}$ is much smaller than the minimum distance between any two eigenvalues. If we now plot the possible combinations of s and j in a plane, we have a situation as shown in figure 3.

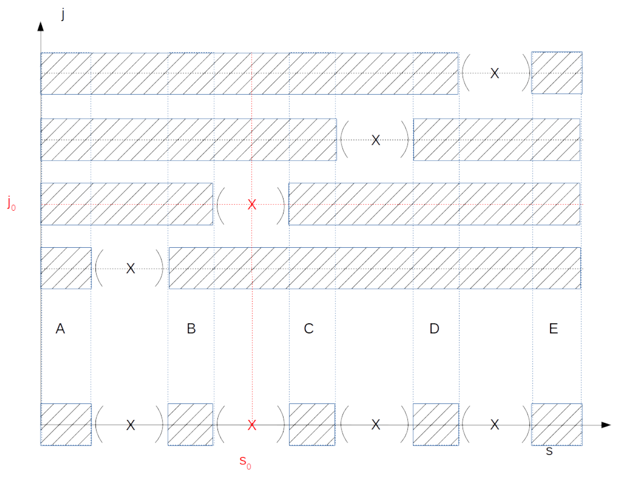

Figure 3: Amplitudes on the (j,s)-lattice

Here each horizontal line represents one different possible of j, i.e. one of the $|\psi_j\rangle$. The parentheses at each line mark the interval of size $2d$ around $\Phi_j N$, i.e. "good" values of s close to $\Phi_j N$. The shaded areas are "bad" areas, i.e. values of s outside this range.

From our previous consideration, we know that if we sum up the squares $|a_{s,j}|^2$ for a given value of j, i.e. along a horizontal line, and for the values of s in the shaded areas, the result will be less at most $\frac{1}{d}$. Let us now use this to find the probability that the measured value of s has a distance of more than d from one of the $\Phi_j N$, i.e. would be in the shaded area on the x-axis of our diagram. The probability for measuring one such s is obtained by adding up the probabilities along j for each s in the shaded area, i.e. the squared amplitudes for all combinations of j and s which are located in the vertical stripes labeled A - E.

Clearly, this is less than the sum over *all* amplitudes in the shaded area, i.e. bounded above by

$$\sum_{j,s \,\text{shaded}} |c_j|^2 |a_{s,j}|^2 \leq \sum_j |c_j|^2 \frac{1}{d} = \frac{1}{d}$$

where we have assumed that our state is normed so that $\sum_j |c_j|^2 = 1$. Therefore, we again obtain that the probability to obtain a "bad" outcome which is not close to one of the $\Phi_j N$ is at most $\frac{1}{d}$, which we can make again arbitrarily small by choosing large values of d and N.

Now suppose that we have measured and obtained a "good" value s_0, close to some $\Phi_{j_0} N$. After the measurement, the state will be

$$\sum_j c_j a_{j,s_0} |s_0\rangle |\psi_j\rangle$$

We see that there are contributions from values of j other than j_0. But again, for all of those j, the index (s_0, j) is in the shaded area. Therefore each individual squared amplitude is bounded from above by $|c_j|^2 \frac{1}{d}$, and we can estimate

$$\sum_{j \neq j_0} |c_j a_{j,s_0}|^2 \leq \sum_{j \neq j_0} |c_j|^2 \frac{1}{d} \leq \frac{1}{d} \sum_j |c_j|^2 = \frac{1}{d}$$

This implies that the contribution of the $|\psi_j\rangle$ for $j \neq j_0$ is small for large d, and in this sense, the state is close to an eigenvector with eigenvalue Φ_{j_0}.

The upshot of this discussion is that the quantum phase estimation does not only allow us to find the *eigenvalue*, starting with an eigenstate, but can also put the system into an approximate *eigenstate* and at the same time tell us the corresponding eigenvalue. If the initial state $|\psi\rangle$ was already close to an eigenstate, i.e. if one of the c_j dominates, we will most likely end up close to $|\psi_j\rangle$. If, on the other hand, the c_j are in the same range, then we will pick one of the eigenstates $|\psi_j\rangle$ according to an almost uniform distribution.

As pointed out in [5], this fact can be used to find the energy eigenvalues and eigenstates of a Hamiltonian. The situation considered in this application is as follows. We are given a quantum system whose state space is finite dimensional and has been mapped in the Hilbert space of a quantum register. The time evolution of this system is described by a hermitian matrix H called the **Hamiltonian**. We are interested in finding the eigenvectors and eigenvalues of H.

Given some time period t, the laws of quantum mechanics tell us that the unitary operator that describes how a state transforms in the time period t is given as

$$U(t) = e^{-\frac{i}{\hbar} H t}$$

Now if λ is an eigenvalue of H with eigenstate $|\psi\rangle$, then of course $e^{i\lambda t}$ is an eigenvalue of $U(t)$ with eigenstate $|\psi\rangle$ and vice versa. More precisely, given an eigenvalue of $U(t)$, we can reconstruct the corresponding eigenvalue of H up to multiples of $\frac{h}{t}$.

The idea is now to apply the quantum phase estimation algorithm to find the eigenvalues of U which will also yield an eigenstate of the Hamiltonian. If we are working in a low energy regime, chances are that the overlap of the initial state with the ground state is bigger than that with any excited state, so that we have good reasons to hope that this procedure will give us the ground state of the Hamiltonian which is of particular interest.

To be able to efficiently apply the quantum phase estimation, we need a way to implement the unitary operations $U(t)^{2^j}$. Now the first observation we can make is that

$$U(t)^{2^j} = U(2^j t)$$

so that if we can implement $U(t)$ for arbitrary t efficiently, we can also implement the needed powers efficiently.

Now let us suppose that our Hamiltonian can be written as a sum of hermitian matrices each of which can be efficiently implemented, for instance because it only acts on a small subspace, i.e.

$$H = \sum_i H_i$$

with efficiently implementable H_i. Then the Trotter-formula gives us an approximation

$$e^{-iHt} \approx \left[\prod e^{-iH_i \frac{t}{n}}\right]^n$$

so that we can apply $U(t)$ approximately by applying each of the operators $e^{-iH\frac{t}{n}}$ in turn and repeating this n times, for some large value of n, see [4] for a more detailed analysis. This does in fact apply to a very large and rich class of Hamiltonians which are of practical interest. For these Hamiltonians, we can therefore implement the $U(t)$-operations and their powers efficiently and therefore carry out the quantum phase estimation algorithm even for a large number of qubits to obtain energy eigenstates. This algorithm is sometimes considered to be a form of **simulation** of the original system that will yield an energy eigenstate, because applying $U(t)$ simulates the time evolution of the original system.

3 QPE and Shor's algorithm

In this section, we will see that there is an intimate relation between the quantum phase estimation and Shor's algorithm for integer factorization, an observation which was first made in [7].

To describe this relation, let us first recall the problem which is at the heart of Shor's factoring algorithm - finding the period. Specifically, suppose we are given a large integer M and a number a which is a unit in $\mathbb{Z}/M\mathbb{Z}$. We want to find the period r, i.e. the smallest number such that $a^r = 1 \mod M$.

As a is a unit, multiplication by a is a permutation of the elements of the group $\mathbb{Z}/n\mathbb{Z}$. Therefore, the linear operator U defined by

$$U|x\rangle = |ax \,\mathrm{mod} M\rangle$$

for $0 \leq x < M$ and trivially on all other members of the computational basis is a unitary operator (here we need a sufficiently large number of qubits to represent M).

Moreover, one can also write down eigenvectors for this operator. In fact, for any integer $0 < k < r$, the state

$$|\psi_k\rangle = \sum_{s=0}^{r-1} e^{2\pi i \frac{sk}{r}} |a^s \operatorname{mod} M\rangle$$

is an eigenvector. To see that this is true, let us apply U to it. We have

$$U|\psi_k\rangle = \sum_{s=0}^{r-1} e^{2\pi i \frac{sk}{r}} |a^{s+1} \operatorname{mod} M\rangle$$

Now let us shift the summation index by one and handle the last term separately. We find that

$$U|\psi_k\rangle = \sum_{s=1}^{r-1} e^{2\pi i \frac{(s-1)k}{r}} |a^s \operatorname{mod} M\rangle + e^{2\pi i \frac{(r-1)k}{r}} |1\rangle$$

But this is the same as

$$U|\psi_k\rangle = e^{-2\pi i \frac{k}{r}} \sum_{s=1}^{r-1} e^{2\pi i \frac{sk}{r}} |a^s \operatorname{mod} M\rangle + e^{-2\pi i \frac{k}{r}} |1\rangle$$

so that we eventually find that

$$U|\psi_k\rangle = e^{-2\pi i \frac{k}{r}} \sum_{s=0}^{r-1} e^{2\pi i \frac{sk}{r}} |a^s \operatorname{mod} M\rangle = e^{-2\pi i \frac{k}{r}} |\psi_k\rangle$$

This shows that $|\psi_k\rangle$ is an eigenstate of U with eigenvalue $\exp(-2\pi i \frac{k}{r})$. Now of course this raises hope. If we could use the phase estimation algorithm to estimate $\frac{k}{r}$, we could again try to use continued fractions as in Shor's algorithm to find the period r.

Unfortunately, it is not obvious how we could prepare the system in one of the states $|\psi_k\rangle$, given that we do not know r. But let us see what happens if we sum up all these states. We find that

$$\begin{aligned}\sum_{k=0}^{r-1} |\psi_k\rangle &= \sum_{k=0}^{r-1} \sum_{s=0}^{r-1} e^{2\pi i \frac{sk}{r}} |a^s \operatorname{mod} M\rangle \\ &= \sum_{s=0}^{r-1} \sum_{k=0}^{r-1} \left[e^{2\pi i \frac{s}{r}}\right]^k |a^s \operatorname{mod} M\rangle = \sum_{s=0}^{r-1} \sum_{k=0}^{r-1} q_s^k |a^s \operatorname{mod} M\rangle\end{aligned}$$

where

$$q_s^k = e^{2\pi i \frac{s}{r}}$$

So again, we find that the sum is in fact a geometric series. If $s = 0$, the series adds up to r. For all other values of s, however, we have that

$$\sum_{k=0}^{r-1} q_s^k = \frac{1 - q_s^r}{1 - q_s} = 0$$

as $q_s^r = 1$. Therefore

$$\sum_{k=0}^{r-1} |\psi_k\rangle = r|1\rangle$$

So we have represented a state that we can easily prepare - the state $|1\rangle$ - as a superposition of eigenstates of our operator U. We have also seen that given such a superposition, the QPE algorithm will randomly select one of the eigenvalues when we perform the final measurement and leave the system in a state which is - at least approximately - in one of the eigenspaces. Specifically, the measurement will give us a value s such that(with high probability)

$$\frac{s}{2^m} \approx \frac{k}{r}$$

We are now in exactly the same position as after performing the final measurement in Shor's factoring algorithm - we known s and m and need to determine k and r. Thus, we can again use a continued fraction expansion to determine the unknown value of r (at least if k and r are co-prime) and therefore the period.

At the first glance, it seems that we have found an alternative to Shor's algorithm for finding the period, given by the following sequence of processing steps.

1. Prepare the system in the state $|1\rangle$
2. Apply the circuit shown in figure 1
3. Apply a quantum Fourier transform
4. Measure the first register and call the result s.
5. Perform a continued fraction expansion to determine the period r

However, it turns out - as observed in [7] and [3] - that this is more or less identical to Shor's algorithm. In fact, let us look at the state after applying the circuit 1. We know that the circuit will transform a state $|\psi\rangle$ into

$$\frac{1}{\sqrt{2}^m} \sum_{k=0}^{2^m-1} |k\rangle U^k |\psi\rangle$$

Let us now apply this to the state $|1\rangle$ to understand the state of the system after the second step of the algorithm above. We obtain

$$\begin{aligned}
\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle U^k |1\rangle &= \frac{1}{\sqrt{N}} \frac{1}{r} \sum_{k=0}^{N-1} |k\rangle \sum_{s=0}^{r-1} U^k |\psi_s\rangle \\
&= \frac{1}{\sqrt{N}} \frac{1}{r} \sum_{k=0}^{N-1} |k\rangle \sum_{s=0}^{r-1} e^{-2\pi i \frac{sk}{r}} |\psi_s\rangle \\
&= \frac{1}{\sqrt{N}} \frac{1}{r} \sum_{k=0}^{N-1} \sum_{s=0}^{r-1} e^{-2\pi i \frac{sk}{r}} |k\rangle \sum_{t=0}^{r-1} e^{2\pi i \frac{ts}{r}} |a^t \operatorname{mod} M\rangle \\
&= \frac{1}{\sqrt{N}} \frac{1}{r} \sum_{k=0}^{N-1} \sum_{t=0}^{r-1} \Big[\sum_{s=0}^{r-1} e^{-2\pi i \frac{s(t-k)}{r}} \Big] |k\rangle |a^t \operatorname{mod} M\rangle
\end{aligned}$$

Now let us look at the inner sum in this expression. This is again a geometric series, depending on $t - k$. For $t = k \mod r$, the series adds up to r, but for $t \neq k \mod r$, the series adds up to zero. Therefore all terms for which $a^t \neq a^k$ disappear, and we are left with

$$\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle U^k |1\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle |a^k \,\mathrm{mod} M\rangle$$

But this is exactly the state that Shor's algorithm produces right before applying the quantum Fourier transform. Even more, as observed in [3], the circuit in figure 1 is performing what is called modular exponentiation in Shor's algorithm. Thus we recognize that Shor's factoring algorithm has an interpretation as an instance of the quantum phase estimation algorithm.

4 Matrix inversion - the HHL algorithm

Let us now turn to another application of the quantum phase estimation to a problem which is at the heart of numerical linear algebra - inverting a matrix.

More specifically, let us assume that we are given a hermitian $N \times N$ matrix A (extensions of the algorithm exists for the non-hermitian case as well) which is invertible, and that we are given a vector $b \in \mathbb{C}^N$. We wish to find a vector $x \in \mathbb{C}^N$ such that

$$Ax = b$$

Doing this for all vectors of a base amounts to finding the columns of the inverse A^{-1}, i.e. solving this problem for arbitrary b is essentially equivalent to finding the inverse of A.

This problem is classically very hard and scales at least linearly in N. Even though this does not sound too bad, it is a desaster for many applications as the dimension N itself tends to grow exponentially with the size of the actual underlying problem.

Suppose, for instance, we are given a linear partial differential equation on $\mathbb{R}^d$. Solving this equation on, say, the unit cube, could be attempted by applying the *finite difference method*, i.e. by introducing a lattice, describing the function by its values on the lattice points and approximating the partial derivatives by finite differences, which turns the partial differential equation into a matrix equation. If $\epsilon < 1$ denotes the step size in all dimensions, then the number of lattice points needed grows like ϵ^{-d}. Thus, the dimension of the underlying vector space grows exponentially with the number of variables d, which makes the calculations intractable on classical hardware for large values of d.

It appears natural to ask whether quantum computing can be applied to solve this problem. Late in 2008, A.W. Harrow, A. Hassidim and S. Lloyd presented an algorithm (see [8]) that could be used for this purpose. To explain the algorithm, let us assume that the dimension N is a power of two, i.e. $N = 2^m$ for some m - this is not a real restriction, as we can always enlarge our vector space and the matrix A if needed. Then we can represent the vector b as a linear combination

$$|b\rangle = \sum_{j=0}^{N-1} b_j |j\rangle$$

where b_j are the coefficients of b. The matrix A then defines a transformation on the Hilbert space that we again denote by A, and the purpose of the algorithm is to place the system in a state $|x\rangle$ such that $A|x\rangle = |b\rangle$. We will call the quantum register in which this state is prepared the working register.

The basic idea of the algorithm is to reduce the problem to the case that the matrix A is diagonal in the standard basis. Then, A is acting by multiplication with the eigenvalues, and finding the solution is trivial - we just need to divide each component of b by the respective eigenvalue. The reduction to this case is done using the quantum phase estimation.

Specifically, let us denote the eigenvalues of A by λ_j, where $j = 0, \dots, N-1$ (i.e. we repeat each eigenvalue according to its multiplicity). After potentially rescaling the matrix A, we can also assume that all eigenvalues are between 0 and 2π.

We start by adding an additional register (called the *clock register* C, for a reason that we will explain in a second) with k qubits in the initial state $|0\rangle$. We then pick a time $t_0 < 1$ and consider the operator

$$U = e^{iAt_0}$$

Then the eigenvalues of U will be $e^{it_0\lambda_j}$, and as $\lambda_j \in (0, 2\pi)$, we can also obtain λ_j from the eigenvalues of U unambiguously.

Let $|\psi_j\rangle$ denote an eigenvector of A with eigenvalue λ_j. Recall that approximately, i.e. up to a term that we treat as an error term, a quantum phase estimation for the operator U amounts to the transformation

$$|0\rangle|\psi_j\rangle \mapsto |\tilde{\lambda_j}\rangle|\psi_j\rangle \tag{2}$$

where $\tilde{\lambda_j}$ is a k-bit approximation of

$$\tilde{\lambda_j} = \frac{2^k t_0}{2\pi}\lambda_j$$

Let us now try to understand how the quantum phase estimation acts on the state $|b\rangle$. As any state, this state can be written as a superposition of eigenvectors, i.e.

$$|b\rangle = \sum_{j=0}^{N-1} \beta_j |\psi_j\rangle$$

for some complex coefficients β_j. Consequently, applying the quantum phase estimation procedure yields the state

$$\sum_{j=0}^{N-1} \beta_j |\tilde{\lambda}_j\rangle|\psi_j\rangle$$

Before we proceed to explain the remaining part of the algorithm, let us see what we want to do on an intuitive level. We want to divide the coefficient of $|\psi_j\rangle$ by λ_j and then "forget" the clock register, i.e. we would like to obtain the state

$$\sum_{j=0}^{N-1} \frac{\beta_j}{\lambda_j}|\psi_j\rangle$$

This state is in fact the solution $|x\rangle$ we are looking for, as the matrix A acts on $|\psi_j\rangle$ simply as multiplication by the eigenvalue λ_j and thus maps this state onto $|b\rangle$!

Unfortunately, the division by λ_j is not a unitary operation. But there is a unitary approximation to this operation which works as follows. We add another ancillary register S with one qubit. On this register, we implement a conditional rotation, i.e. a unitary operator realizing the mapping

$$|k\rangle^C |0\rangle^S \mapsto |k\rangle^C \big[\sqrt{1 - \frac{C^2}{k^2}} |0\rangle^S + \frac{C}{k} |1\rangle^S\big]$$

for $k \neq 0$ - here the upper index indicates the register in which a state is living, and C is a constant that we need to choose. We refer to the appendix D or [10] for details on how this sort of conditional rotations can be realized. After applying this transformation to our full state, we end up with

$$\sum_{j=0}^{N-1} \beta_j |\tilde{\lambda}_j\rangle |\psi_j\rangle \big[\sqrt{1 - \frac{C^2}{\tilde{\lambda}_j^2}} |0\rangle^S + \frac{C}{\tilde{\lambda}_j} |1\rangle^S\big]$$

Now we revert the quantum phase estimation, i.e. apply its inverse to realize the inverse of the mapping rule (2). This will give us the state

$$\sum_{j=0}^{N-1} \beta_j |0\rangle |\psi_j\rangle \big[\sqrt{1 - \frac{C^2}{\tilde{\lambda}_j^2}} |0\rangle^S + \frac{C}{\tilde{\lambda}_j} |1\rangle^S\big]$$

We now ignore the clock register (which is again in state $|0\rangle$) and measure the ancillary register S. With a certain probability, this will yield the outcome $|1\rangle$ in the register S and therefore leave the working register in a state proportional to

$$\sum_{j=0}^{N-1} \frac{\beta_j}{\tilde{\lambda}_j} |\psi_j\rangle$$

which is our solution $|x\rangle$.

It is helpful to visualize the overall flow of the algorithm graphically, as in figure 4 (reproduced from [9]).

After this short overview, let us now try to understand some of the subleties and ramifications of this approach. First, let us discuss the value of the constant C and the success probability. Clearly, for our conditional rotation to make sense, we need to choose C such that $C < \lambda_j$ for all j, i.e. C can be at most the smallest eigenvalue. Without loss of generality, let us assume that the eigenvalues are sorted, so that λ_0 is the smallest eigenvalue and λ_{N-1} is the largest. Then we could set $C = \tilde{\lambda}_0$.

Let us now try to understand the success probability of the algorithm. For the sake of simplicity, let us assume that $|b\rangle$ is normalized, so that $\sum_j |\beta_j|^2 = 1$. Immediately before the measurement, the state is a sum

$$\sum_{j=0}^{N-1} \beta_j \sqrt{1 - \frac{C^2}{\tilde{\lambda}_j^2}} |\psi_j\rangle |0\rangle^S + \sum_{j=0}^{N-1} \beta_j \frac{C}{\tilde{\lambda}_j} |\psi_j\rangle |1\rangle^S$$

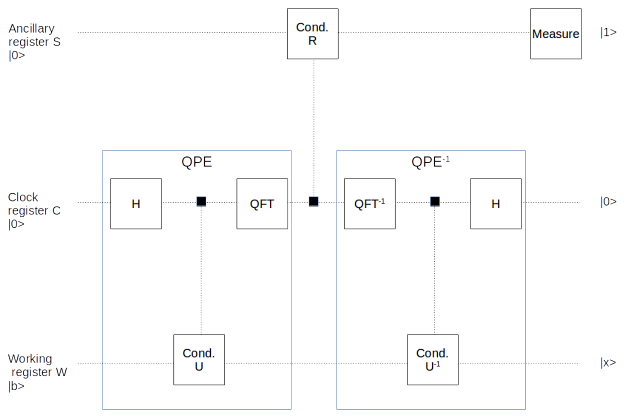

Figure 4: HHL Algorithm - an overview

The success probability is given by the probability to obtain 1 when measuring the register S, which we can estimate by

$$P = \sum_{j=0}^{N-1} |\beta_j|^2 \frac{C^2}{\tilde{\lambda}_j^2} = \sum_{j=0}^{N-1} |\beta_j|^2 (\frac{\lambda_0}{\lambda_j})^2 \geq (\frac{\lambda_0}{\lambda_{N-1}})^2 = \frac{1}{\kappa^2}$$

where the number κ, defined as the ratio between the largest and the smallest eigenvalue, is called the **condition number** of the matrix A.

Thus for large values of the condition number κ, the success probability can become very small, depending on the coefficients β_j. Matrices with this property are called **ill-conditioned**. Intuitively, these matrices are close to being non-invertible, so that for certain vectors b, the inversion becomes numerically unstable. As a high condition number decreases the number of trial required to be successful, it increases the runtime by a factor κ^2 (which, however, can be reduced to κ by applying Grover's amplitude amplification). In addition, as explained in [8], the algorithm can be modified to handle also ill-conditioned matrices - this is done by using a three-dimensional Hilbert space for the ancillary register S instead of a two-dimensional Hilbert space, where the third state in addition to our previous $|0\rangle$ and $|1\rangle$ represents "limited success".

Further improvements can be made by applying more refined versions of the phase estimation algorithm. To make contact with the terminology used in [8], let us quickly recall how the first step of the phase estimation algorithm works. Here we apply powers of U condition on the clock register, i.e. we apply the transformation

$$|k\rangle|\psi\rangle \mapsto |k\rangle U^k|\psi\rangle$$

If we spell this out using A instead of U and call the variable in the clock register t instead of k, this becomes

$$|t\rangle|\psi\rangle \mapsto |t\rangle e^{iAtt_0}|\psi\rangle$$

In a physical interpretation, the operator A would be the Hamiltonian of a physical system, and the operator $e^{iA\tau}$ represents the time evolution over the time τ. Thus, the mapping above can be interpreted as simulating the evolution of the state $|\psi\rangle$ over different time periods $t \cdot t_0$ for different values of t, given by the content of the clock register - this is the reason why this register is called the clock register, and is also what the authors mean in [8] when describing their algorithm as "applying e^{iAt} to $|b\rangle$ for a superposition of different times t".

Now it turns out that instead of putting the clock register in the initial state $\sum_t |t\rangle$ before applying the controlled U-operation as we have done it, we could as well start with properly chosen initial states of the form

$$|\Phi_0\rangle = \sum_{t=0}^{T-1} c_t |t\rangle$$

for some coefficients c_t, where $T = 2^m$. The coefficients c_t can then be chosen to sharpen the peak around the eigenvalues after applying the quantum Fourier transform even further and to therefore improve the precision of the algorithm, we refer to [8] or [9] for further details and the exact calculations.

Another critical point in the algorithm is the efficiency of applying the matrix e^{iAtt_0}, i.e. the "simulation" of the time evolution given by A. We need to be able to do this efficiently, which - as already discussed in the section on simulation - is not possible for arbitrary A. In [8], it is assumed that the matrix A is s-sparse (i.e. it has at most s non-zero entries per row) and efficiently row computable, meaning that for each row, these entries can be efficiently computed in a time scaling at most linearly with s. Under these conditions, the overall runtime of the algorithm is

$$\tilde{O}(\log(N)\frac{s^2\kappa^2}{\epsilon})$$

where ϵ is the required precision. According to [9], the best classical algorithm that is known to apply for this class of matrices has runtime

$$\tilde{O}(Ns\kappa\log(\frac{1}{\epsilon}))$$

Thus assuming that κ and s grow slowly with N, the quantum algorithm provides an exponential speedup.

However, there is a caveat - we need to be able to prepare the input state $|b\rangle$ efficiently and only obtain a quantum version $|x\rangle$ of the result x. Measuring a component of $|x\rangle$ will again destroy the state, so we cannot easily obtain the corresponding vector x without repeating the procedure $O(N)$ times, destroying the speed advantage over classical algorithms. As pointed out in [8], the algorithm unfolds its real power therefore in situations where we are not interested in $|x\rangle$ itself, but in the expectation values of some hermitian operators, which we can measure and estimate with a low number of repetitions. If, for instance, the solution represents the solution of a physically motivated partial differential equation, this could give us access to expectation values of physical observables and thus yield insights that would be difficult to obtain using classical computation.

A The quantum Fourier transform

In this section, we briefly summarize the most important facts about the quantum Fourier transform. As the quantum Fourier transform is the quantum equivalent of the discrete Fourier transform, let us describe the discrete Fourier transform first.

Given an integer N, the discrete Fourier transform is usually defined to be a mapping from the space of complex sequences with N elements to itself. To simplify our notation a bit, we will denote the i-th element of a sequence of N complex numbers as $x[i]$ instead of x_i and let the index i start at zero, so that such a sequence is given by the N complex numbers $x[0], x[1], \dots, x[N-1]$. We combine these numbers into a vector $x \in \mathbb{C}^N$.

Given such a sequence x, the discrete Fourier transform of x is defined to be the sequence X with elements

$$X[k] = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{-\frac{2\pi i}{N} jk} = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \eta^{-jk}$$

where we denote by $\eta = e^{\frac{2\pi i}{N}}$ the standard N-th root of unity. Mapping vectors to vectors, we can think of the Fourier transform as a mapping

$$\mathcal{F} \colon \mathbb{C}^N \to \mathbb{C}^N$$

which is clearly linear and which can be shown to be in fact unitary. The formula for the inverse Fourier transform, is given by

$$x[k] = \frac{1}{\sqrt{N}} \sum_s X[s] \eta^{sk}$$

Let us now return to the world of quantum computers. Imagine that we have a quantum computer with n qubits. The states of this quantum computer are then described by rays in a Hilbert space with $N = 2^n$ dimensions, namely the n-fold tensor product of the one-qubit Hilbert space. With respect to the usual standard basis labeled by the vectors $|x\rangle$, with x ranging from 0 to N-1, we can then consider any vector as a sequence, using the identification

$$x = \sum_k x[k] |x\rangle$$

To this sequence, we can apply the Fourier transform. This will give us a unitary transformation $\mathcal{F}$, described by

$$\mathcal{F}(\sum_k x[k] |k\rangle) = \sum_k X[k] |k\rangle = \frac{1}{\sqrt{N}} \sum_{s,k} x[s] \eta^{-sk} |k\rangle$$

As any unitary transformation, this transformation can be realized by a quantum circuit. In fact, one can show (see [2] and the references therein) that this can be done with a number of quantum gates that scales as O(n^2).

B The success probability of the QPE algorithm

In this section, we will follow the treatment in [3] and take a closer look at the amplitudes a_l that we have used in section 1. Let us quickly recall the setup. We have defined an approximation

$$\Phi = \frac{t}{N} + \delta$$

with $0 \leq t < N$ and $|\delta| < \frac{1}{2N}$. Let us now ignore the special case $\delta = 0$ that we have already considered).

It turns out that we have to distinguish between the case of a positive and a negative δ. Let us start with the **case** $\delta > 0$. We have seen that with

$$a_l = \frac{1}{N} \frac{1 - e^{2\pi i \delta N}}{1 - e^{2\pi i (\frac{l}{N} + \delta)}}$$

the state that our quantum registers have after applying the inverse Fourier transform in the QPE algorithm is

$$\sum_{-\frac{N}{2} \leq l < \frac{N}{2}} a_l |t - l \,\mathrm{mod} N\rangle |\psi\rangle$$

We have good reasons to believe that this has a peak around $l = 0$. To qualify this peak, let us assume that we are given some integer d with $2 \leq d < \frac{N}{2}$ and try to understand the probability that the algorithm will yield a value outside the band with width $\frac{d}{N}$ around Φ. The situation is shown in figure 5.

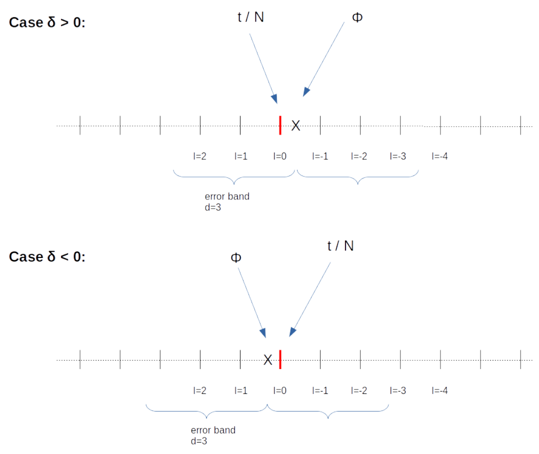

Figure 5: Approximation of Φ

AS $\delta > 0$, the first "bad" values of l that take us out of the error band of width $2d$ around Φ are $l = 3 = d$ to the left and $l = -4 = -(d+1)$ to the right. In other words, we are looking for an upper bound for the sum of the squares of all amplitudes a_l with $l \geq d$ or $l \leq -(d+1)$, i.e. for the expression

$$P = \sum_{l=-\frac{N}{2}}^{-(d+1)} |a_l|^2 + \sum_{l=d}^{\frac{N}{2}-1} |a_l|^2 \tag{3}$$

Let us first try to find an upper bound for each invididual a_l. To find an upper bound for a_l, we need to find a lower bound for the denominator. Now by assumption, we have

$$-\frac{N}{2} \leq l \leq \frac{N}{2} - 1$$

i.e.

$$-\frac{1}{2} \leq \frac{l}{N} \leq \frac{1}{2} - \frac{1}{N}$$

At the same time, we know that δ is less than $\frac{1}{N}$ and not negative, so that

$$-\frac{1}{2} \leq \frac{l}{N} + \delta \leq \frac{1}{2}$$

We can therefore apply the estimate in section C and find that

$$|1 - e^{2\pi i(\frac{l}{N}+\delta)}| \geq 2\frac{2\pi|\frac{l}{N} + \delta|}{\pi} = 4|\frac{l}{N} + \delta|$$

Now let us see what this estimate implies in combination with equation (3). Let us first look at the second sum in equation (3). For the values of l in the range that appear in that sum, we clearly have

$$\frac{l}{N} + \delta \geq \frac{l}{N}$$

and thus obtain

$$\sum_{l=d}^{\frac{N}{2}-1} |a_l|^2 \leq \sum_{l=d}^{\frac{N}{2}-1} \frac{1}{N^2}\Big[\frac{2}{4(\frac{l}{N}+\delta)}\Big]^2 \leq \sum_{l=d}^{\frac{N}{2}-1} \frac{1}{N^2}\frac{1}{4(\frac{l}{N}^2)} = \sum_{l=d}^{\frac{N}{2}-1} \frac{1}{4l^2}$$

For the first sum in (3), we can use a similar argument. Here, l is negative. Thus

$$|\frac{l}{N} + \delta| = -\frac{l}{N} - \delta \geq -\frac{l}{N} - \frac{1}{2N} = \frac{-l - \frac{1}{2}}{N}$$

By passing from $-l$ to l in the sum, we therefore obtain the estimate

$$\sum_{l=-\frac{N}{2}}^{-(d+1)} |a_l|^2 \leq \sum_{l=d+1}^{\frac{N}{2}} \frac{1}{N^2}\Big[\frac{2}{4(\frac{l-\frac{1}{2}}{N})}\Big]^2 = \sum_{l=d+1}^{\frac{N}{2}} \frac{1}{4(l-\frac{1}{2})^2}$$

Now let us put all this together. We find that the probability to obtain a measured value which deviates by at least d from t is bounded from above by

$$P \leq \sum_{l=d+1}^{\frac{N}{2}} \frac{1}{(2l-1)^2} + \sum_{l=d}^{\frac{N}{2}-1} \frac{1}{(2l)^2} = \sum_{x=2d}^{N-1} \frac{1}{x^2} = \sum_{x=2d-1}^{N-2} \frac{1}{(x+1)^2}$$

Now, as illustrated in figure 6, the step function which is equal to $(x+1)^{-2}$ on the interval from x to $x+1$ is everywhere at most equal to x^{-2}. The sum as written is the integral of that step function, and therefore we can estimate the sum by the integral of x^{-2} and obtain

$$P \leq \int_{2d-1}^{N-2} \frac{1}{x^2}dx \leq \int_{2d-1}^{\infty} \frac{1}{x^2}dx = \frac{1}{2d-1} \leq \frac{1}{d}$$

Thus, the probability to obtain a measured value outside of an interval with width d around the best approximation t drops with d^{-1} if we make d large. In might be irritating that this value does not depend on N, but in fact d is just

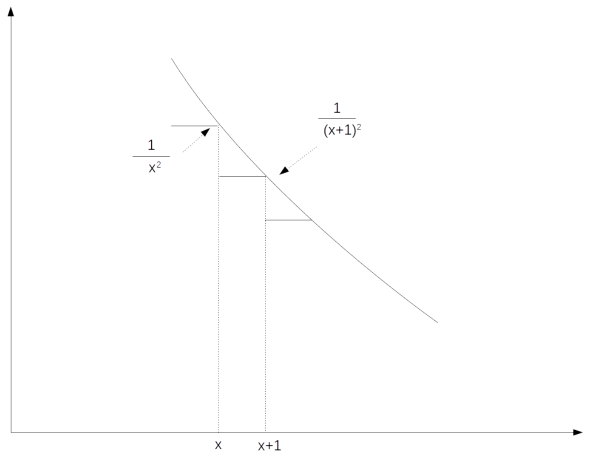

Figure 6: Step function approximation

the relative error, not the absolute error which is d divided by N. Thus we can make the error arbitrarily small by making N large.

In fact, suppose we are given numbers $\epsilon, \Delta > 0$. Let us now choose d sufficiently large such that

$$\frac{1}{d} \leq \epsilon$$

and at the same time $N \geq 4d$ so large that

$$\frac{d}{N} \leq \Delta$$

If we now let s again denote the result of our measurement, we find that

$$P(|\frac{s}{N} - \Phi| \geq \Delta) \leq P(|\frac{s}{N} - \Phi| \geq \frac{d}{N}) \leq \frac{1}{d} \leq \epsilon$$

In other words, if we make N sufficiently large, the probability that the value of s/N - which we use an approximation for Φ - deviates from the best possible m-bit approximation t/N by more than a given threshold Δ can be made arbitrarily small. Phrased differently, by choosing N large, the peak of a_l around $l = 0$ can be made arbitrarily sharp.

Let us now complete our argument by considering the **case** $\delta < 0$. In this case, it is useful to write the quantum state after applying the quantum Fourier transform slightly differently, namely as

$$\sum_{-\frac{N}{2} < l \leq \frac{N}{2}} a_l |t + l \,\mathrm{mod} N\rangle |\psi\rangle$$

Thus we have now excluded $-\frac{N}{2}$, but included $\frac{N}{2}$. As these two numbers are equivalent module N, it is clear that this does not change the sum. With this choice of the range of l, we now have

$$-\frac{1}{2} + \frac{1}{N} \leq \frac{l}{N} \leq \frac{1}{2}$$

Now δ is greater than $-\frac{1}{N}$ and negative, so that

$$-\frac{1}{2} \leq \frac{l}{N} + \delta \leq \frac{1}{2}$$

Now we can again apply the estimate in section C and find that our previous estimate

$$|1 - e^{2\pi i(\frac{l}{N}+\delta)}| \geq 2\frac{2\pi|\frac{l}{N} + \delta|}{\pi} = 4|\frac{l}{N} + \delta|$$

remains valid for the allowed values of l.

Let us now again write down the probability for a deviation or more than d/N. With the new convention for the range of l, this probability is now

$$P = \sum_{l=-\frac{N}{2}+1}^{-d} |a_l|^2 + \sum_{l=d+1}^{\frac{N}{2}} |a_l|^2 \tag{4}$$

see again figure 5. Now let us again look at each sum in turn. If $l < 0$, we can estimate

$$|\frac{l}{N} - \delta| \geq |\frac{l}{N} + \frac{1}{2N}|$$

and obtain

$$|a_l|^2 \leq \frac{1}{4(l + \frac{1}{2})^2}$$

For the second sum, $l > 0$, and therefore (as $\delta < 0$),

$$|\frac{l}{N} - \delta| \geq |\frac{l}{N}|$$

so that

$$|a_l|^2 \leq \frac{1}{4l^2}$$

If we again replace l by $-l$ in the first sum, we therefore find that

$$P \leq \sum_{d}^{l=\frac{N}{2}+1} \frac{1}{4(l - \frac{1}{2})^2} + \sum_{l=d+1}^{\frac{N}{2}} \frac{1}{4l^2}$$

which can also be written as

$$P \leq \sum_{d}^{l=\frac{N}{2}+1} \frac{1}{(2l - 1)^2} + \sum_{l=d+1}^{\frac{N}{2}} \frac{1}{(2l)^2}$$

This sum is again the sum over x^{-2} for all values from $2d - 1$ up to $N + 1$, except $2d$. As the missing value is positive, we can add it to the sum without making its value smaller and find that

$$P \leq \sum_{x=2d-1}^{N+1} \frac{1}{x^2} = \sum_{x=2d-2}^{N} \frac{1}{(x + 1)^2} \leq \int_{2d-2}^{\infty} \frac{1}{x^2}dx = \frac{1}{2d - 2} \leq \frac{1}{d}$$

as before.

C Estimates for points on the circle

Here we will take a closer look at the estimates for $1 - e^{i\Phi}$ used before. First, let us try to find a bound from above. The absolute value $|1 - e^{i\Phi}|$ is the length of the straight line connecting the point 1 and the point $e^{i\Phi}$, as shown in diagram 7.

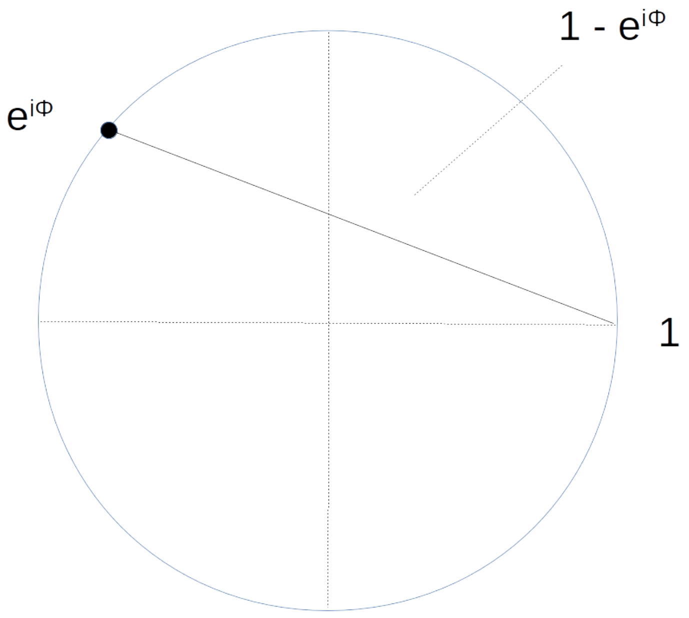

Figure 7

Intuitively, it is clear that this path is shorter than going around, i.e. that

$$|1 - e^{\Phi}| \leq |\Phi|$$

for all angles Φ. Formally, this can be seen as follows. First, using the Euler identities, a short calculation shows that for all angles,

$$|1 - e^{i\Phi}|^2 = 2(1 - \cos\Phi)$$

Now consider the two functions f and g given by $f(\Phi) = 2(1 - \cos\Phi)$ and $g(\Phi) = \Phi^2$. At the origin, both functions are zero. Their derivatives are both zero at the origin as well, and for their second derivatives, we have $f'' \leq g''$ for all values. As furthermore both functions are symmetric around the origin, we can conclude that $f \leq g$ everywhere. This proves our inequality.

Next, let us try to apply a similar argument to prove our second inequality. First, we rewrite our expression for $|1 - e^{i\Phi}|$ a bit. By the additional theorem for the cosine, we have

$$1 - \cos\Phi = 2\sin^2\frac{\Phi}{2}$$

so that we obtain the simple expression

$$|1 - e^{i\Phi}| = 2\sin\frac{\Phi}{2}$$

whenever the right hand side is non-negative. To use this to derive a lower bound, we therefore need a lower bound for the sine. So let us now consider the function F given by

$$F(x) = \sin x - \frac{2x}{\pi}$$

Clearly, this function has zeros at $x = 0$ and $x = \frac{\pi}{2}$. We claim that it does not have any zeros between these two points. There are several ways to see this. One possible argument is that the function represents the difference between the function cos and the straight line from the origin to the value of cos at $\frac{\pi}{2}$. As the cosine is concave in this region, this line is always below the graph of the cosine and does not intersect it in any other points within this interval. Thus we have shown that for $x \in [0, \frac{\pi}{2}]$, the inequality

$$\sin x \geq \frac{2x}{\pi}$$

holds. Consequently, we immediately find that for $\Phi \in [0, \pi]$, we have the inequality

$$|1 - e^{i\Phi}| = 2\sin\frac{\Phi}{2} \geq 2\frac{\Phi}{\pi}$$

Using the symmetry of the right hand side with respect to reflexion at the origin, we therefore finally obtain that for all $\Phi \in [-\pi, \pi]$, we have the inequality

$$|1 - e^{i\Phi}| \geq 2\frac{|\Phi|}{\pi}$$

D Conditional rotations

In this section, we will summarize a few basic facts about a procedure known as conditional rotation that appears in the HHL algorithm.

In its simplest version, the conditional rotation is rotating an ancillary qubit by an angle that is contained in a control register. More precisely, let

$$\sigma_Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = -iY$$

be the Pauli Y-matrix. Clearly $\sigma_Y^2 = 1$, and therefore

$$\begin{aligned} e^{i\sigma_Y\Phi} &= \sum_{k=0}^{\infty} \frac{i^k\Phi^k}{k!}\sigma_Y^k \\ &= \sum_k (-1)^k \frac{\Phi^{2k}}{(2k)!} \cdot 1 + i\sum_k (-1)^k \frac{\Phi^{2k+1}}{(2k+1)!}\sigma_Y \\ &= \cos\Phi \cdot 1 + i\sin\Phi \cdot \sigma_Y \\ &= \begin{pmatrix} \cos\Phi & \sin\Phi \\ -\sin\Phi & \cos\Phi \end{pmatrix} \end{aligned}$$

Consequently, the unitary operation

$$R(\Phi) = e^{-i\Phi\sigma_Y} = \begin{pmatrix} \cos\Phi & -\sin\Phi \\ \sin\Phi & \cos\Phi \end{pmatrix}$$

is the rotation by the angle Φ in the counter-clockwise direction and maps $|0\rangle$ to

$$\cos\Phi|0\rangle + \sin\Phi|1\rangle$$

After these preliminaries, we can now define the **conditioned rotation** as follows. We are given a control register C and a one-qubit register. The conditioned rotation is the unitary operation that is acting as

$$|\Phi\rangle|0\rangle \mapsto \cos\Phi|\Phi\rangle|0\rangle + \sin\Phi|\Phi\rangle|1\rangle$$

In other words, we rotate the one qubit-register by the angle contained in the working register.

Let us now see how such a conditional rotation can be implemented as a quantum circuit. First, given a reference state $|a\rangle$ of the computational basis and a unitary operator U acting on some working register, we can always implement a conditional operation that acts as U if a control register is in the state $|a\rangle$ and trivially otherwise. In fact, using CNOT gates, we can first implement the operation

$$|x\rangle \mapsto |x \oplus a\rangle$$

on the control register. This will put $|0\rangle$ into the control register if and only if $x = a$. We can then invert all qubits in the control register and use the result as the control register of a controlled U-operation on the working register to obtain the desired transformation.

With this primitive, we could now implement a conditional rotation as a sequence of rotations $R(\Phi)$ where, as above, each $R(\Phi)$ is applied if and only if the control register is in the state $|\Phi\rangle$. This approach is simple, but has a major drawback - we need one rotation circuit for each possible value of Φ representable in the control register, and therefore the number of steps grows as 2^m, where m is the number of qubits in the control register.

Fortunately, there is a different approach. Let Φ_k denote the k-th digit in the m-bit representation of Φ, i.e.

$$\Phi = \sum_{k=0}^{2^m-1} \Phi_k 2^k$$

Then we can express the rotation by the angle Φ as

$$R(\Phi) = e^{-i\Phi\sigma_Y} = \prod_k \left(e^{-i2^k\sigma_Y}\right)^{\Phi_k}$$

In other words, the entire rotation is a sequence of rotations around the angles 2^k, applied only if the k-th bit of Φ is set. We can therefore implement the conditional rotation as a sequence of m controlled rotations by angles 2^k, controlled by the k-th qubit of the working register. The resulting circuit is shown on the left hand side of figure 8 (note the similarity with the exponentation step of the quantum phase estimation, which is in fact just a conditional application of U^k).

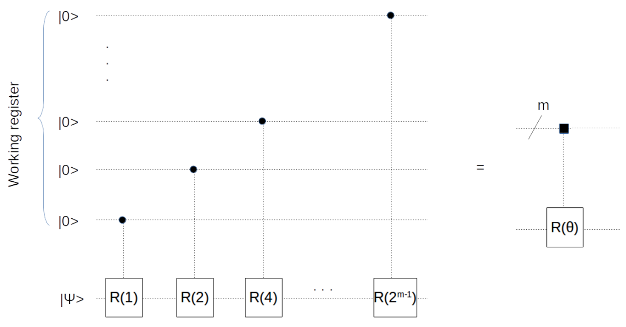

Figure 8: Conditional rotation

As indicated on the right hand side of figure 8, a conditional operation is often indicated by a solid box on the control register, with a slash indicating the number of qubits in the control register.

Now we can take this even further. Suppose we are given a control register and a function f which is classically efficiently computable. We known on general grounds (see for instance the discussion in chapter 6 of [?]) that we can then efficiently construct a quantum circuit U_f that acts as

$$|x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$$

We can now use the second register as the control register for a conditional rotation. This will give us a circuit that acts as

$$|x\rangle|0\rangle|0\rangle \mapsto \cos f(x)|x\rangle|f(x)\rangle|0\rangle + \sin f(x)|x\rangle|f(x)\rangle|1\rangle$$

Thus we rotate the third register by the angle $f(x)$, there x is the content of the first register, as graphically indicated in figure 9.

This is exactly what we do in the HHL algorithm. Here, our control register contains a state $|\tilde{\lambda}\rangle$. We add one ancillary qubit and want to apply the transformation

$$|\tilde{\lambda}\rangle|0\rangle \mapsto |\tilde{\lambda}\rangle[\sqrt{1-\frac{C^2}{\tilde{\lambda}}^2}|0\rangle + \frac{C}{\tilde{\lambda}}|1\rangle]$$

If we set

$$f(\tilde{\lambda}) = \arcsin\frac{C}{\tilde{\lambda}}$$

then this is exactly the situation as above, and we see that this transformation can be implemented as a conditional as shown in this section. Of course, this only works if $\tilde{\lambda}$ is between 0 and C, but we can extend f arbitrarily outside this range as our conditions on the matrix and the choice of C make sure that the states to which we apply the transformation are restricted to the allowed range.

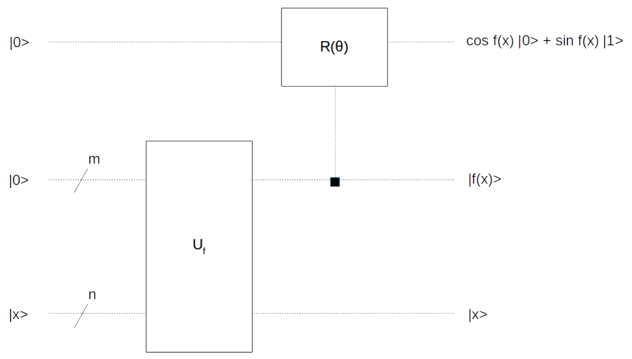

Figure 9: Conditional rotation by f(x)

References

[1] M.A. Nielsen, I.L. Chaung, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, 2010

[2] P. Shor, *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*, SIAM J.Sci.Statist.Comput. Vol. 26 Issue 5 (1997), pp 1484–1509, available as arXiv:quant-ph/9508027v2

[3] R. Cleve, A. Ekert, C. Macchiavello, M. Mosca, *Quantum Algorithms Revisited*, Proceedings of the Royal Society of London, Series A, 454:339354, 1998, also available as arXiv:quant-ph/9708016

[4] S. Lloyd, *Universal quantum computers*, Science, New Series, Vol. 273, No. 5278 (Aug. 23, 1996), pp. 1073–1078

[5] D.S. Abrams, S. Lloyd, A quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors, Phys. Rev. Lett.83 5162–5165, 1999, also available as arXiv:quant-ph/9807070

[6] G.H. Hardy, E.M. Wright, *An introduction to the theory of numbers*, Oxford University Press, Oxford, 1975

[7] A.Yu. Kitaev, *Quantum measurements and the Abelian Stabilizer Problem*, arXiv:quant-ph/9511026

[8] A.W. Harrow,A. Hassidim,S. Lloyd, *Quantum algorithm for linear systems of equations*, Phys. Rev. Lett. vol. 15, no. 103, pp. 150502 (2009) or arXiv:0811.3171

[9] D. Dervovic,M. Hebster,P. Mountney, S. Severini,N. Usher,L. Wossnig, *Quantum linear systems algorithms: a primer*, arXiv:1802.08227

[10] Y. Cao, A. Papageorgiou, I. Petras,J. Traub, S. Kais, *Quantum algorithm and circuit design solving the Poisson equation*, New J. Phys. 15 (2013) or arXiv:1207.2485