Exact and approximate quantum Fourier transform

Mathis Beaudoin

Summer 2025

Contents

Preface

This is a collection of notes about the quantum Fourier transform and the approximate quantum Fourier transform I've made during my internship at Université de Sherbrooke under the supervision of Pr. Cunlu Zhou and Pr. Dave Touchette.

1 Discrete Fourier transform (DFT)

The discrete Fourier transform (DFT) corresponds to the discrete version of the Fourier transform (FT), a mathematical tool which gives a frequency description of the content present in some function. For example, the FT can be applied to music in order to find what frequencies (musical notes) are present in it. The FT acts on some analytical function, whereas the DFT acts on a set of data points. Naturally, the latter is the version used by computers. Mathematically, the DFT takes a vector $\vec{x} = [x_0, ..., x_{N-1}]^T \in \mathbb{C}^N$ and transforms it into a new vector $\vec{y} = [y_0, ..., y_{N-1}]^T \in \mathbb{C}^N$ with components

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{-i\frac{2\pi}{N}jk} \tag{1.1}$$

There also exists an inverse operation called the inverse Fourier transform (IFT) which, from the known frequencies, allows the original function to be reconstructed. Yet again, there is a discrete equivalent called the inverse discrete Fourier transform (IDFT) which can be obtained by isolating x_j in (1.1).

$$x_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} y_k e^{i\frac{2\pi}{N}jk} \tag{1.2}$$

On the canonical basis $\{\vec{e}_0, ..., \vec{e}_{N-1}\}$, the DFT is simply a change of basis. Indeed, because $\vec{e}_j$ only has one non-zero element at the j-th index, we get

$$\text{DFT}(\vec{e}_j) = \frac{1}{\sqrt{N}} \left[1, e^{-i\frac{2\pi}{N}j}, ..., e^{-i\frac{2\pi}{N}j(N-1)}\right]^T = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{-i\frac{2\pi}{N}jk} \vec{e}_k = \vec{u}_j \tag{1.3}$$

Therefore, $\{\vec{e}_j\}$ becomes $\{\vec{u}_j\}$ and we can verify that $\{\vec{u}_j\}$ is an orthonormal basis by only checking the orthonormality property. This is because orthonormality ensures linear independence and because the number of vectors in the set matches the dimension of the vector space.

$$\vec{u}_a \cdot \vec{u}_b = \frac{1}{N} \sum_{k,k'=0}^{N-1} e^{-i\frac{2\pi}{N}(k'b - ka)} \vec{e}_k \cdot \vec{e}_{k'} = \frac{1}{N} \sum_{k=0}^{N-1} e^{-i\frac{2\pi}{N}k(b-a)} \xrightarrow{a=b} \vec{u}_a \cdot \vec{u}_b = \frac{1}{N} \sum_{k=0}^{N-1} e^{-i\frac{2\pi}{N}k\cdot 0} = 1$$

$$\xrightarrow{a\neq b} \vec{u}_a \cdot \vec{u}_b = \frac{1}{N} \sum_{k=0}^{N-1} \left(e^{-i\frac{2\pi}{N}(b-a)}\right)^k = \frac{1}{N} \frac{1 - e^{-i\frac{2\pi}{N}(b-a)N}}{1 - e^{-i\frac{2\pi}{N}(b-a)}} = 0$$

This shows that $\{\vec{u}_j\}$ is an orthonormal basis. So, the change of basis matrix for the DFT is

$$U = (\vec{u}_0, ..., \vec{u}_{N-1}) = \frac{1}{\sqrt{N}} \left[e^{-i\frac{2\pi}{N}jk}\right]_{j,k=0}^{N-1} \tag{1.4}$$

It is straightforward to see that U is symmetric and unitary. Also, by (1.2), we determine that the IDFT is described by $U^\dagger$.

$$e^{-i\frac{2\pi}{N}jk} = e^{-i\frac{2\pi}{N}kj} \implies U^\intercal = U, \quad U^\dagger U = [\vec{u}_j \cdot \vec{u}_k]_{j,k=0}^{N-1} = \mathbb{I}$$

2 Quantum Fourier transform (QFT)

By convention, the quantum Fourier transform (QFT) is the quantum version of the IDFT. So, we work with kets in a Hilbert space of dimension $N = 2^n$ where n is the number of qubits. The QFT is an important subroutine in many algorithms such as in the quantum phase estimation (QPE).

2.1 Derivation

On a vector $|j\rangle$ from the computational basis, based on 1.3, we expect

$$\text{QFT}\,|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{i\frac{2\pi}{N}jk}\,|k\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{i\frac{2\pi}{2^n}jk}\,|k\rangle = |\phi(j)\rangle \tag{2.1}$$

Knowing that k can be represented in binary notation as $k_{n-1}...k_0 = k_0 2^0 + ... + k_{n-1}2^{n-1} = \sum_{l=0}^{n-1} k_l 2^l$, we get

$$\text{QFT}\,|j\rangle = \frac{1}{\sqrt{2^n}} \sum_{k_{n-1}=0}^{1} ... \sum_{k_0=0}^{1} e^{i2\pi j \sum_{l=0}^{n-1} \frac{k_l}{2^{n-l}}}\,|k_{n-1}...k_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{k_{n-1}=0}^{1} ... \sum_{k_0=0}^{1} \prod_{l=0}^{n-1} e^{i2\pi j \frac{k_l}{2^{n-l}}}\,|k_{n-1}...k_0\rangle$$

From that, we group each $e^{i2\pi j \frac{k_l}{2^{n-l}}}$ with their corresponding state $|k_l\rangle$ and explicitly do all summations. This yields

$$\text{QFT}\,|j\rangle = \bigotimes_{l=0}^{n-1} \left(\frac{|0\rangle + e^{i2\pi \frac{j}{2^{n-l}}}\,|1\rangle}{\sqrt{2}} \right)$$

In binary, $j = j_{n-1}...j_0 = j_0 2^0 + ... + j_{n-1}2^{n-1}$ which means that $\frac{j}{2^{n-l}} = j_{n-1}...j_{n-l}.j_{n-l-1}...j_0$ moves the bits to the right by $n-l$ positions. Here, we split the integer part to the left and the decimal part to the right by the "." symbol. As j is in the argument of a complex exponential, its integer part can be discarded to then only keep the decimal part $0.j_{n-l-1}...j_0 = j_{n-l-1}2^{-1} + ... + j_0 2^{-(n-l)}$. With that,

$$\text{QFT}\,|j\rangle = \bigotimes_{l=0}^{n-1} \left(\frac{|0\rangle + e^{i2\pi\ 0.j_{n-l-1}...j_0}\,|1\rangle}{\sqrt{2}} \right) = |\phi(j)\rangle \tag{2.2}$$

As we see, for the l-th qubit in $|j\rangle$, which we note $|j_l\rangle$,

$$|j_l\rangle \xrightarrow{\text{QFT}} \frac{|0\rangle + e^{i2\pi\ 0.j_{n-l-1}...j_0}\,|1\rangle}{\sqrt{2}} = |\phi(j)_{n-l-1}\rangle \tag{2.3}$$

2.2 Implementation with a quantum circuit

To build a quantum circuit that applies the QFT, we will use Hadamard gates H and phase gates R_k.

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},\ R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{i2\pi/2^k} \end{bmatrix} \tag{2.4}$$

On the basis state $|j\rangle$, applying H on the last qubit $|j_{n-1}\rangle$ brings j_{n-1} to the first position of the decimal number.

$$H\,|j_{n-1}\rangle\,|j_{n-2}...j_0\rangle = \frac{|0\rangle + (-1)^{j_{n-1}}\,|1\rangle}{\sqrt{2}}\,|j_{n-2}...j_0\rangle = \frac{|0\rangle + e^{i2\pi\ 0.j_{n-1}}\,|1\rangle}{\sqrt{2}}\,|j_{n-2}...j_0\rangle$$

Then, a R_2 gate controlled by $|j_{n-2}\rangle$ and applied to $|j_{n-1}\rangle$ puts j_{n-2} in the second position.

$$CR_2\left(\frac{|0\rangle + e^{i2\pi\ 0.j_{n-1}}\,|1\rangle}{\sqrt{2}}\,|j_{n-2}\rangle\right)|j_{n-3}...j_0\rangle = \frac{|0\rangle + e^{i2\pi\ 0.j_{n-1}j_{n-2}}\,|1\rangle}{\sqrt{2}}\,|j_{n-2}...j_0\rangle$$

By repeating the same process with $CR_3, ..., CR_n$ controlled by $|j_{n-3}\rangle, ..., |j_0\rangle$ respectively and all applied to $|j_{n-1}\rangle$, we find

$$\frac{|0\rangle + e^{i2\pi\ 0.j_{n-1}...j_0}\,|1\rangle}{\sqrt{2}}\,|j_{n-2}...j_0\rangle = |\phi(j)_{n-1}\rangle\,|j_{n-2}...j_0\rangle$$

According to (2.2), $|\phi(j)_{n-1}\rangle$ should normally be stored in the first qubit of the register and currently it is stored in the last qubit of the register. We will solve this problem later. The pattern is repeated to all other $|j_l\rangle$ from the bottom of the register to the top where we first apply a Hadamard gate followed by $CR_2, ..., CR_{l+1}$ controlled by $|j_{l-1}\rangle, ..., |j_0\rangle$ respectively. As said earlier, this results in an output state where the l-th qubit stores $|\phi(j)_l\rangle$ that should normally be stored in the $n-1-l$-th qubit. To solve the problem, we can swap at the end the first qubit with the last qubit, the second qubit with the second last qubit, etc... Finally, this gives us (2.2) and measurements can be added at the end if required.

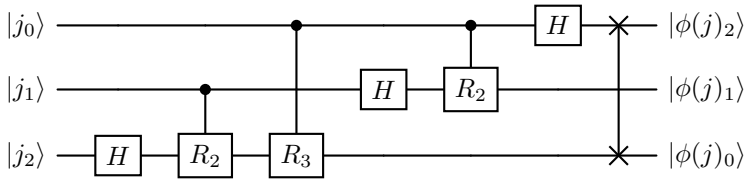


Figure 1: Quantum circuit for the QFT on $n = 3$ qubits

The circuit requires n Hadamard gates, $\sum_{i=1}^{n}(n-i) = \frac{n(n-1)}{2}$ phase gates and at most $\frac{n}{2}$ SWAPs. In general, this corresponds to $\mathcal{O}(n^2)$ gates and, depending on the connectivity of the qubits, additional SWAPs may be needed for the controlled phase gates to be done in practice. Also, the circuit has a width of n and, by recursion, we can show that the circuit has a depth of $2n \implies \mathcal{O}(n)$. If we don't want the SWAPs at the end of the circuit, we can pass all of them through the circuit from right to left and interchange the gates that we face along the way. Then, figure 1 becomes

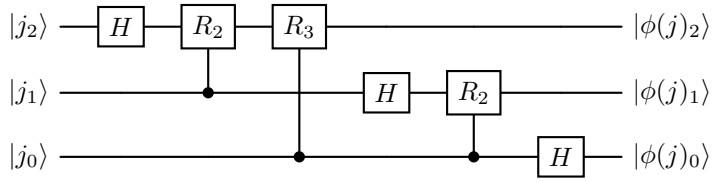


Figure 2: Alternative quantum circuit for the QFT without any SWAPs on $n = 3$ qubits

2.3 Implementation with a dynamic quantum circuit

2.3.1 Dynamic quantum circuits

Generally, at the end of a quantum circuit, all qubits or a subset of qubits are measured to obtain a bitstring that can be post-processed in order to get a comprehensible result. On the other end, dynamic quantum circuits have measurements within the circuit (in between gates). The measurement outputs are post-processed so that one or more quantum gates can be added (or not) classically to the quantum circuit depending on the result. This is called a "feed-forward" operation and is often used in the field of quantum error correction for example. If a given measurement requires a gate to be added on multiple qubits, a fan-out gate can be used to do so in parallel.

Dynamic quantum circuits can be generated from quantum circuits with their measurements at the end by the deferred measurement principle. This principle states that a controlled gate commutes with a measurement when the measured qubit is the one that controls the gate. Intuitively, this makes sense, because measuring a control qubit after the gate is applied tells us whether or not the target qubit was affected by the gate. Therefore, we could measure the control qubit before this controlled gate and add it to the circuit on the target qubit classically depending on the measurement outcome.

2.3.2 Dynamic quantum circuit for the QFT

In the QFT, there are a lot of controlled operations, so the deferred measurement principle could be used here if all qubits are measured at the end (like in the QPE). To derive a dynamic quantum circuit for the QFT, we'll need to change the circuit by interchanging the control and target qubit of all controlled phase gates.

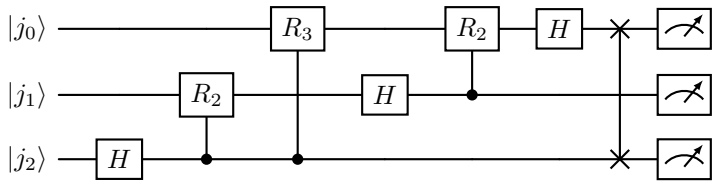


Figure 3: Quantum circuit for the QFT on $n = 3$ qubits with control and target qubits interchanged

This action still gives us (2.2) because the controlled phase gates are symmetric (like a CZ gate). Then, each measurement can be passed through the SWAPs (we will swap the measured bitstrings after the measurements) and through the control portion of the controlled phase gates by the deferred measurement principle. After measuring, the controlled phase gates can be added classically depending on the measurement outcomes.

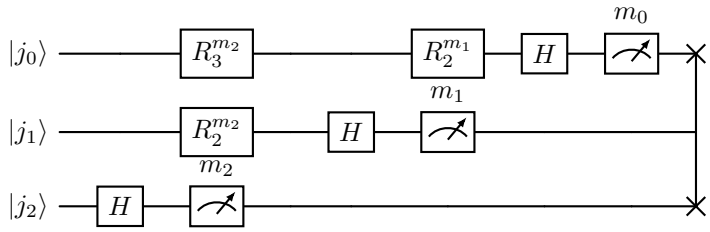


Figure 4: Dynamic quantum circuit for the QFT on $n = 3$ qubits

2.3.3 Benefits of a dynamic quantum circuit for the QFT

The dynamic version of the QFT assumes that measurements are performed at the end, which isn't always the case when the QFT is used as a subroutine. But, when this is the case like in the QPE, it can be a powerful tool when coupled with error suppression techniques like dynamical decoupling (DD). This helps to preserve a better fidelity throughout the process as shown in [2].

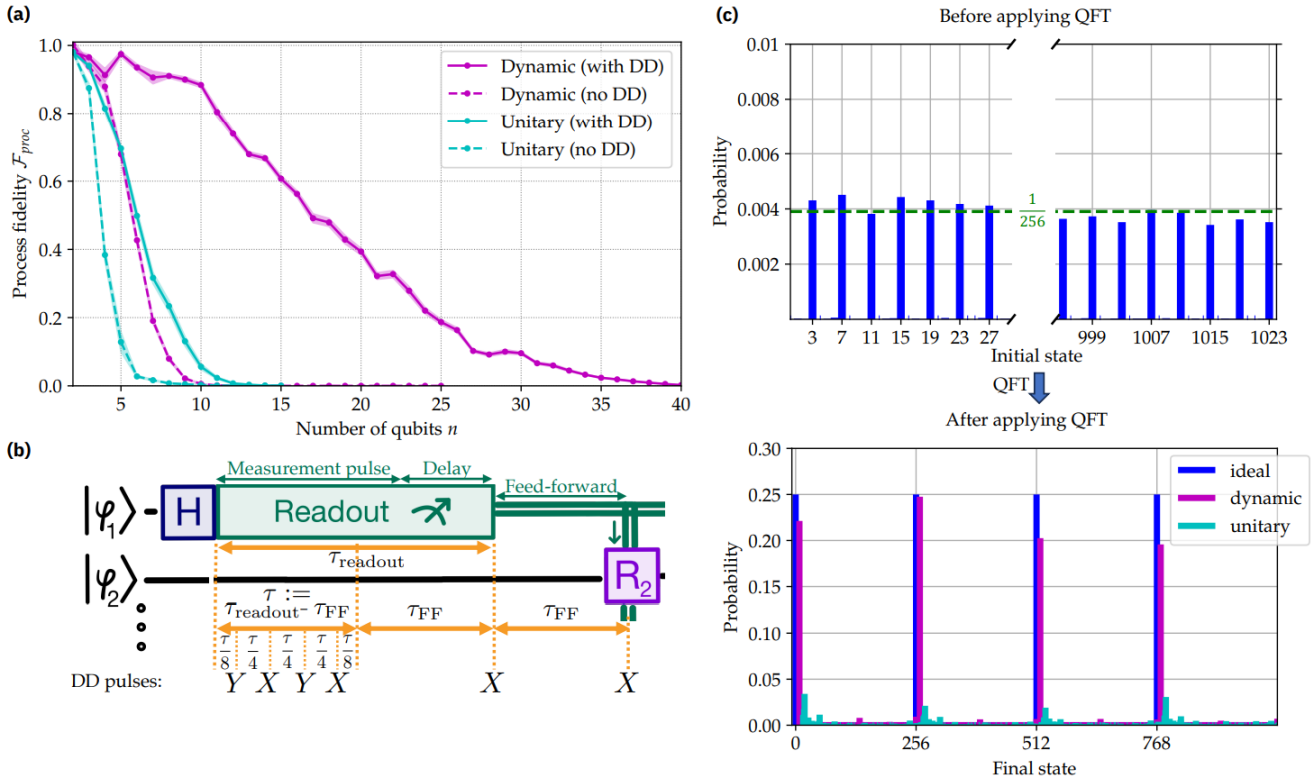


Figure 5: (a) Performance of the QFT for different implementations (b) Dynamical decoupling during the measurement and the feed-forward operation (c) Example with 10 qubits

The dynamic version of the QFT also has practical benefits when attempting to run it on real hardware, more precisely in the transpilation step. Since the dynamic quantum circuit for the QFT only requires one qubit gates and no specific connectivity, the transpilation step is simpler compared to the regular implementation of the QFT. Indeed, on IBM's quantum computers, the phase gates are native (so they are unchanged) and the Hadamard gate has a simple decomposition on the native gates. Therefore, when transpiled, the dynamic version of the QFT doesn't change that much and, importantly, remains a compact circuit which improves the results. For the basic implementation of the QFT, the transpilation step changes the circuit a lot because of the connectivity constraints and the gates on two qubits which are sometimes over a long range.

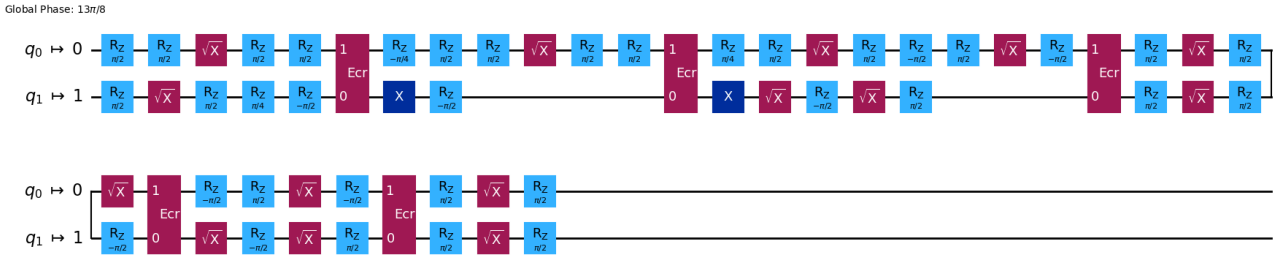


Figure 6: QFT for $n = 2$ qubits after the transpilation step on FakeSherbrooke

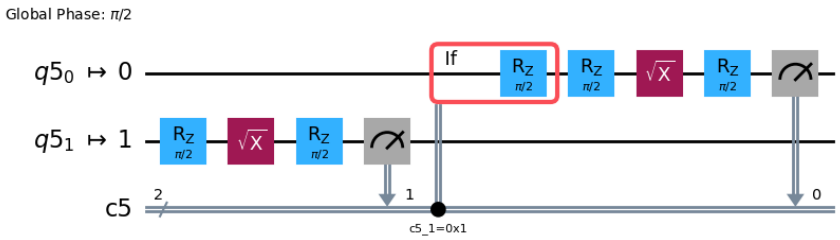


Figure 7: Dynamic version of the QFT for $n = 2$ qubits after the transpilation step on FakeSherbrooke

2.4 Implementation with nearest-neighbour connectivity

Here, we want to have a version of the all-to-all circuit in figure 1 that is made on a line of qubits (with nearest-neighbour connectivity). We will use additional SWAPs to move the qubits so that the long-range controlled gates are done on neighbouring qubits. Essentially, starting from the bottom qubit to the top one of figure 1, we apply its gates in a staircase pattern using SWAPs so that the controlled phase gates are done on neighbouring qubits. Also, this construction naturally moves the qubits in their correct final position, mimicking the final layer of SWAPs found at the end of figure 1. This implementation can be found in [6].

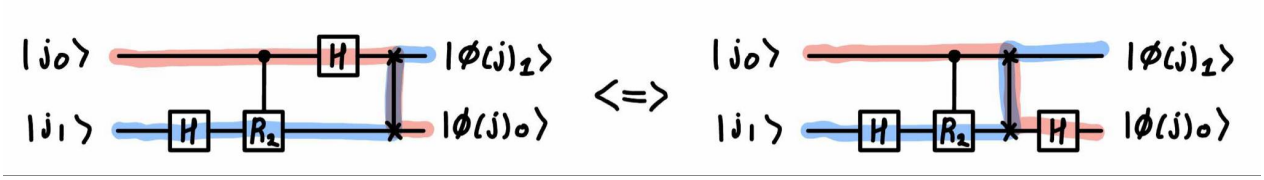


Figure 8: All-to-all QFT (on the left) and QFT on a line (on the right) for $n = 2$ qubits

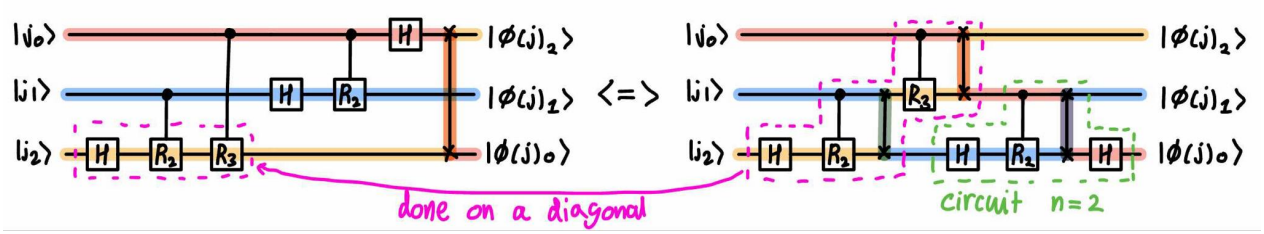


Figure 9: All-to-all QFT (on the left) and QFT on a line (on the right) for $n = 3$ qubits

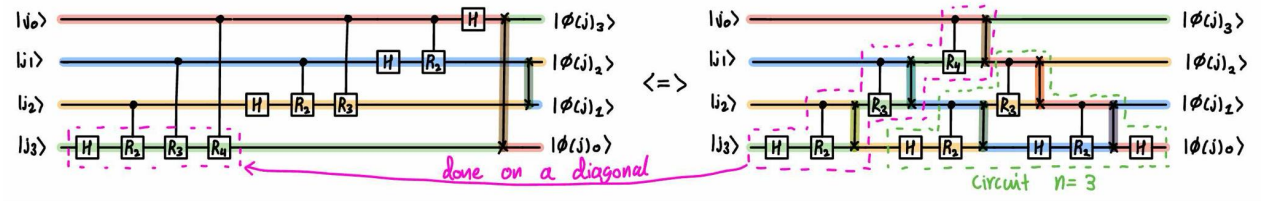


Figure 10: All-to-all QFT (on the left) and QFT on a line (on the right) for $n = 4$ qubits

With a closer look, each staircase in the circuit on a line applies all the gates of one of the qubits in the all-to-all circuit. This qubit, with the positioning of the SWAPs in the circuit on a line, progressively moves to its correct final position and all qubits initially above it get shifted down by one position. Thus, the next qubit we need to apply gates to always ends up at the bottom and staircase layers can be applied one after the other.

As we can see, this implementation on a line can be generalized recursively to obtain the circuit on a line for any number of qubits. Indeed, we can take the circuit on a line for $n-1$ qubits and add on top of it, in a staircase pattern, the gates found on the last qubit of the all-to-all circuit on n qubits.

To compute the depth of this circuit, we can observe that each time a staircase is added, there is "room" under it to do some operations of the circuit on $n-1$ qubits in parallel. Visually, we "slide" the circuit on $n-1$ qubits to the left under the new staircase and obtain all previous figures. But, there will only be room after the CR_3 gate of the new staircase. Therefore, a new staircase increases the depth by 4 with its 1 Hadamard gate, 2 controlled phase gates and 1 SWAP on the left which don't allow for more operations to be done in parallel. Since the circuit on n qubits is built recursively from the circuit for $n=2$ which also has a depth of 4, this means we always add 4 to the depth with each new qubit for $n \geq 2$. This yields a total depth of $4(n-1)\ \forall n \geq 2 \implies \mathcal{O}(n)$ for the QFT on a line.

2.5 Reversed implementation with nearest-neighbour connectivity

In this case, we want a QFT implementation on a line of n qubits where the output is reversed. This is the same as trying to do the all-to-all circuit without the last layer of SWAPs on a line. We propose the following circuit, which is similar in a way to what we found previously in figures 8-10.

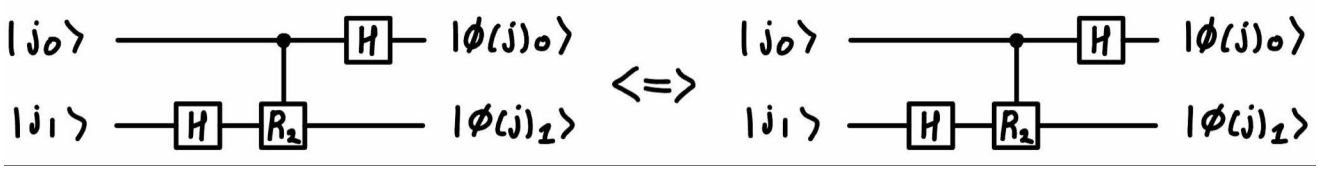


Figure 11: Reversed all-to-all QFT (on the left) and reversed QFT on a line (on the right) for $n=2$ qubits

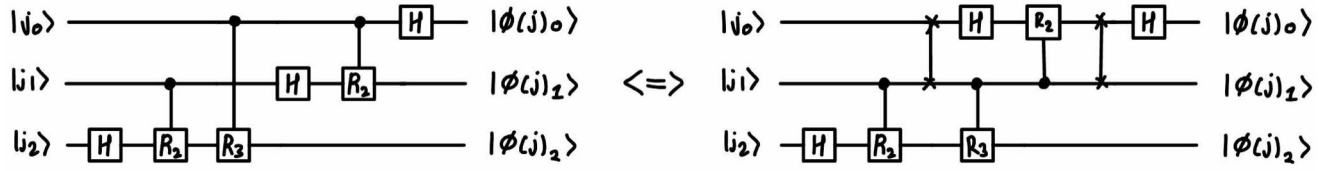


Figure 12: Reversed all-to-all QFT (on the left) and reversed QFT on a line (on the right) for $n=3$ qubits

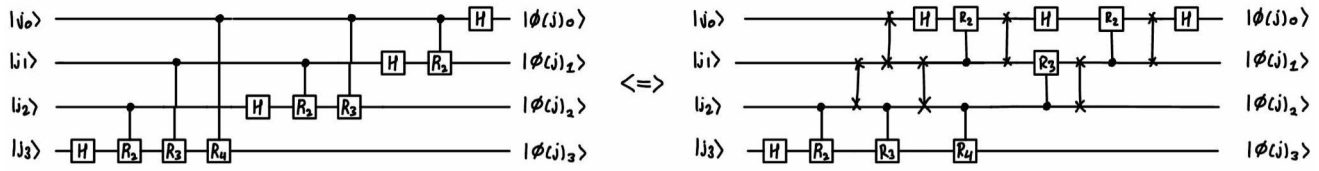
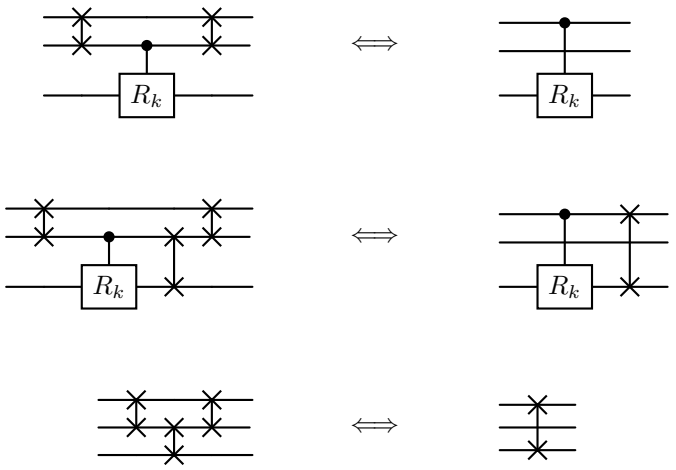


Figure 13: Reversed all-to-all QFT (on the left) and reversed QFT on a line (on the right) for $n=4$ qubits

One can show recursively, by analyzing the circuit and seeing what operations can be done in parallel, that the depth of this circuit is $5n-8\ \forall n \geq 3$ and $5n-7$ for $n=2$. In general, this corresponds to $\mathcal{O}(n)$ for the depth of this implementation.

2.6 Reversed implementation with nearest-neighbour connectivity on two meshed registers

The goal is to build a circuit for the reversed QFT on a line where two registers A and B are meshed together in an alterning pattern $|A_0, B_0, ..., A_{n-1}, B_{n-1}\rangle$. We want the reversed QFT on a line for n qubits to be computed on the register B and for it to still be done with nearest-neighbour connectivity in linear depth. So, in this situation, A will act as a spacer to separate the qubits of B by one tick and is left untouched. First, we consider the following identities.



Then, consider the circuits for the reversed QFT on a line as seen in the figures 11-13 and do them on the register B. This "stretches" the circuits and they are no longer done on a line. To put everything back on a line, we can simply use the identities shown above and assert the resulting circuit has a linear depth.

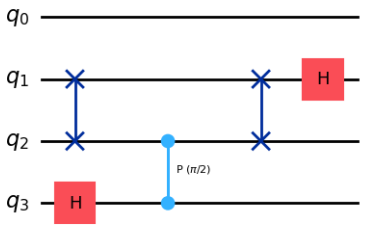


Figure 14: Reversed QFT on a line done on the register B for $n = 2$ qubits

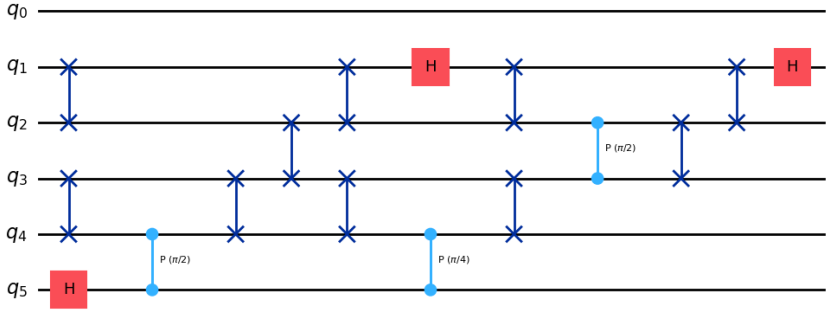


Figure 15: Reversed QFT on a line done on the register B for $n = 3$ qubits

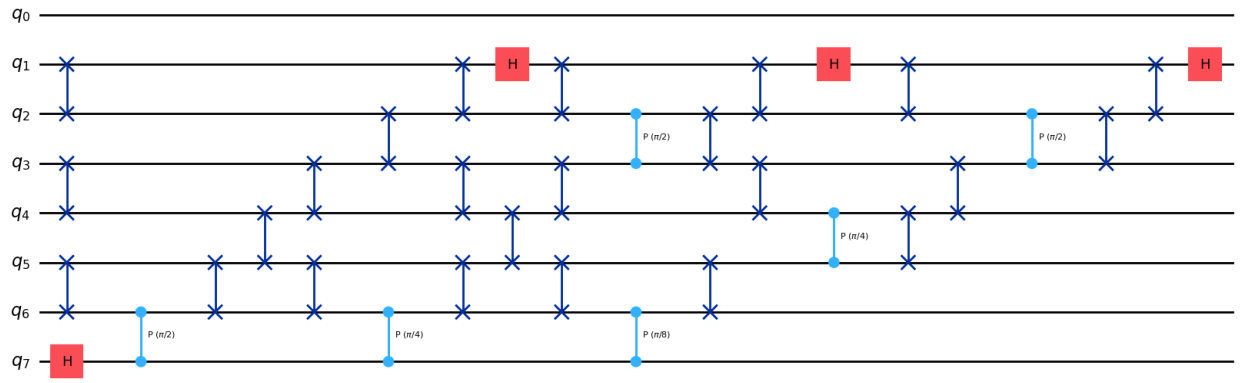


Figure 16: Reversed QFT on a line done on the register B for $n = 4$ qubits

For this implementation, one can show recursively, by analyzing the circuit and seeing what operations can be done in parallel, that the depth is $8n - 13$ $\forall n \geq 3$ and $8n - 12$ for $n = 2$. This corresponds to $\mathcal{O}(n)$ in general for this implementation.

Implementation	**All-to-all**	**Line**	**Line + reversed**	**Line + reversed + 2 meshed registers**
Depth	$2n$	$4(n-1)$	$5n-8$	$8n-13$

Table 1: Depth value for different QFT implementations

3 Approximate QFT (AQFT)

When the number of qubits n becomes very large, some of the phase gates R_k will add a negligible phase to the quantum state. For the l-th qubit, the gates $CR_2, ...CR_{l+1}$ are applied, so in some cases the subscripts become large. When this happens, the added phase $e^{i2\pi/2^k}$ will be very close to 1. In that case, it is computationally interesting to see if an approximate version of the QFT (AQFT) is possible, where gates that add a negligible phase are removed from the circuit.

3.1 Derivation

We rewrite (2.1) with both j and k in binary notation,

$$\mathrm{QFT}\,|j\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{i\frac{2\pi}{2^n}\sum_{m=0}^{n-1}\sum_{l=0}^{n-1} j_m k_l 2^{l+m}} |k\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{i\frac{2\pi}{2^n}\sum_{0\le m,l\le n-1, 0\le l+m\le n-1} j_m k_l 2^{l+m}} |k\rangle \tag{3.1}$$

because $e^{i\frac{2\pi}{2^n} j_m k_l 2^{l+m}} = 1$ when $l+m \ge n$. This corresponds to the full QFT where all gates are present. Now, we check what happens when we remove all controlled phase gates from the circuit so that we are only left with Hadamard and SWAP gates. This yields the state

$$\begin{aligned}
\mathrm{SWAPs}\left(H^{\otimes n}|j\rangle\right) &= \frac{1}{\sqrt{2^n}}\left(|0...0\rangle + (-1)^{j_0}|10...0\rangle + ... + (-1)^{j_{n-1}+...+j_0}|1...1\rangle\right) \\
= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{i\frac{2\pi}{2^n}\sum_{m=0}^{n-1} j_m k_{n-1-m} 2^{n-1}} |k\rangle &= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{i\frac{2\pi}{2^n}\sum_{0\le m,l\le n-1, l+m=n-1} j_m k_l 2^{l+m}} |k\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{i\frac{2\pi}{2^n}\sum_{0\le m,l\le n-1, n-1\le l+m\le n-1} j_m k_l 2^{l+m}} |k\rangle
\end{aligned} \tag{3.2}$$

where "SWAPs" is the final sequence of SWAPs in figure 1. (3.1) and (3.2) are two extreme cases of the AQFT where (3.1) corresponds to the full QFT and (3.2) is a totally incomplete QFT. In the sum of the argument, the full QFT has the constraint $0 \le l+m \le n-1$ and the incomplete one has the constraint $n-1 \le l+m \le n-1$. Naturally, to transition from one extreme to the other, it would make sense to define the AQFT with an integer parameter $k_{\max}$ ($\mathrm{AQFT}_{k_{\max}}$) like so.

$$\mathrm{AQFT}_{k_{\max}}\,|j\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{i\frac{2\pi}{2^n}\sum_{0\le m,l\le n-1, n-k_{\max}\le l+m\le n-1} j_m k_l 2^{l+m}} |k\rangle \tag{3.3}$$

When $k_{\max} = n$, we obtain the full QFT and, when $k_{\max} = 1$, we get the totally incomplete QFT. But in between, to have a smooth transition between the two extremes, we hope that (3.3) gives the QFT where all R_k gates such that $k > k_{\max}$ are removed from the circuit. We can verify that by simply applying the constraint $n-k_{\max} \le l+m \le n-1$ to the full QFT and extending the equation. First, by (2.2),

$$\mathrm{QFT}\,|j\rangle = \bigotimes_{l=0}^{n-1}\left(\frac{|0\rangle + e^{2\pi i\sum_{m=0}^{n-l-1} j_m 2^{l+m}/2^n}|1\rangle}{\sqrt{2}}\right) \tag{3.4}$$

where

$$|j_l\rangle \xrightarrow{\text{QFT}} |\phi(j)_{n-l-1}\rangle = \frac{|0\rangle + e^{2\pi i \sum_{m=0}^{n-l-1} j_m 2^{l+m}/2^n} |1\rangle}{\sqrt{2}} \tag{3.5}$$

Then, we consider the previous constraint $n-k_{\max} \leq l+m \leq n-1$, which is the same as $n-k_{\max}-l \leq m \leq n-l-1$. This gives us bounds on the range of values m can take in the sum of (3.5). Therefore, if l is small enough, we will have $0 < n-k_{\max}-l \leq m \leq n-l-1$ and (3.5) changes to

$$\frac{|0\rangle + e^{2\pi i \sum_{m=n-k_{\max}-l}^{n-l-1} j_m 2^{l+m}/2^n} |1\rangle}{\sqrt{2}} = \frac{|0\rangle + e^{2\pi i \left(j_{n-l-1}2^{-1}+\ldots+j_{n-k_{\max}-l}2^{-k_{\max}}\right)} |1\rangle}{\sqrt{2}} = \frac{|0\rangle + e^{2\pi i \ 0.j_{n-l-1}\ldots j_{n-k_{\max}-l}} |1\rangle}{\sqrt{2}}$$

As we see, the phase gates will not go further than $R_{k_{\max}}$. This makes sense, because when l is small enough, there will be a lot of bits in the decimal number $0.j_{n-l-1}...j_0$, which will require phase gates R_k with high values of k that we want to remove in the approximate QFT. The other case is when l is large enough that $n-k_{\max}-l \leq 0$ and the original range in the sum of (3.5) doesn't need to be truncated. This is due to the fact that $0.j_{n-l-1}...j_0$ will not contain that many bits, so the phase gates used will respect the threshold and nothing gets removed. Overall, the $\text{AQFT}_{k_{\max}}$ defined above behaves the way we want.

Instead of looking at both cases when l is small or large enough, we can derive a general equation for $\text{AQFT}_{k_{\max}}$ that encapsulates everything and doesn't explicitly tell us to take into consideration some constraint. We know that when a qubit needs gates to be removed, that is when $n-k_{\max}-l > 0$, it is described by

$$\frac{|0\rangle + e^{2\pi i \sum_{m=n-k_{\max}-l}^{n-l-1} j_m 2^{l+m}/2^n} |1\rangle}{\sqrt{2}}$$

Also, when $n-k_{\max}-l \leq 0$, no gates are removed and we always get

$$\frac{|0\rangle + e^{2\pi i \sum_{m=0}^{n-l-1} j_m 2^{l+m}/2^n} |1\rangle}{\sqrt{2}}$$

for such a qubit. In this last case, we can extend the summation's lower bound to $n-k_{\max}-l$ in order to match the equation where gates are removed. If we do that, the sum will partially be on indices of j that are negative, which are bits for the decimals positions. Since j is an integer, those indices all have the value 0 and they don't contribute anything to the phase. Therefore,

$$\frac{|0\rangle + e^{2\pi i \sum_{m=n-k_{\max}-l}^{n-l-1} j_m 2^{l+m}/2^n} |1\rangle}{\sqrt{2}}$$

can be used to describe all qubits and the $\text{AQFT}_{k_{\max}}$ is rewritten as

$$\text{AQFT}_{k_{\max}} |j\rangle = \bigotimes_{l=0}^{n-1} \left(\frac{|0\rangle + e^{i2\pi \sum_{m=n-k_{\max}-l}^{n-l-1} j_m 2^{l+m}/2^n} |1\rangle}{\sqrt{2}} \right) = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{i\frac{2\pi}{2^n} \sum_{l=0}^{n-1} \sum_{m=n-k_{\max}-l}^{n-1-l} j_m k_l 2^{l+m}} |k\rangle$$

Finally, we can show that $\sum_{l=0}^{n-1} \sum_{m=n-k_{\max}-l}^{n-1-l} j_m k_l 2^{l+m} = \sum_{m=0}^{n-1} \sum_{l=n-k_{\max}-m}^{n-1-m} j_m k_l 2^{l+m}$ by deleting the negative indices for j, adding negative indices for k (which all have the value 0 because k is also an integer) and reordering the sums.

$$\mathrm{AQFT}_{k_{\max}} |j\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{i\frac{2\pi}{2^n}\sum_{m=0}^{n-1}\sum_{l=n-k_{\max}-m}^{n-1-m} j_m k_l 2^{l+m}} |k\rangle \tag{3.6}$$

To write the QFT in a similar fashion, we set $k_{\max} = n$ in (3.6). Since negative indices of k have the value 0, we delete them from the sum.

$$\mathrm{QFT}\,|j\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{i\frac{2\pi}{2^n}\sum_{m=0}^{n-1}\sum_{l=-m}^{n-1-m} j_m k_l 2^{l+m}} |k\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{i\frac{2\pi}{2^n}\sum_{m=0}^{n-1}\sum_{l=0}^{n-1-m} j_m k_l 2^{l+m}} |k\rangle \tag{3.7}$$

3.2 Implementation with a quantum circuit

To build a quantum circuit for the $\mathrm{AQFT}_{k_{\max}}$, we simply take the circuit for the exact QFT and remove all gates R_k where $k > k_{\max}$.

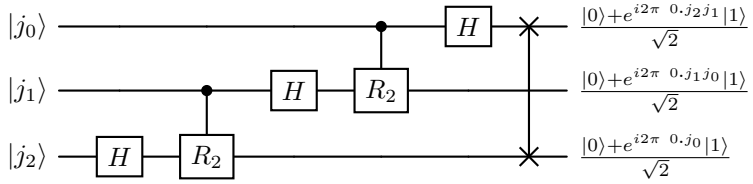


Figure 17: Quantum circuit for the AQFT with $k_{\max} = 2$ on $n = 3$ qubits

Normally, the first output in figure 17 should be $\frac{|0\rangle+e^{i2\pi\ 0.j_2j_1j_0}|1\rangle}{\sqrt{2}}$ instead of $\frac{|0\rangle+e^{i2\pi\ 0.j_2j_1}|1\rangle}{\sqrt{2}}$, but a R_3 has been removed (compare with figure 1). Still, this circuit assumes an all-to-all connectivity and has a width of n. For the number of gates and the depth, looking at both extreme cases of the AQFT, we can go from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ and from $\mathcal{O}(n)$ to $\mathcal{O}(1)$ respectively depending on how pronounced the approximation is (how many gates are deleted from the circuit).

3.3 Bounds on the error

By how much do the exact QFT and AQFT differ? By slightly rearranging (3.6),

$$\mathrm{AQFT}_{k_{\max}} |j\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{i2\pi\left(\sum_{m=0}^{n-1}\sum_{l=0}^{n-1-m} j_m k_l 2^{l+m-n} - \sum_{m=0}^{n-1}\sum_{l=0}^{n-k_{\max}-m-1} j_m k_l 2^{l+m-n}\right)} |k\rangle$$

In the argument, the sum on the left is the phase we would find in the exact QFT. Therefore, the sum on the right is what we remove from the phase of the exact QFT to obtain the AQFT or, in other words, the difference in the phase between the exact and approximated version for the state $|k\rangle$ given the input $|j\rangle$. If we note this difference $\zeta_{j,k}$, then

$$0 \leq \zeta_{j,k} = \sum_{m=0}^{n-1} \sum_{l=0}^{n-k_{\max}-m-1} j_m k_l 2^{l+m-n} \leq \sum_{m=0}^{n-1} \sum_{l=0}^{n-k_{\max}-m-1} 2^{l+m-n}$$

and this is true for all j and k. Further expanding the sums, we see that some terms can be removed

$$0 \leq \zeta_{j,k} \leq \underbrace{\sum_{m=n-k_{\max}}^{n-1} \sum_{l=0}^{n-m-k_{\max}-1} 2^{l+m-n}}_{\text{no terms}} + \sum_{m=0}^{n-k_{\max}-1} \sum_{l=0}^{n-m-k_{\max}-1} 2^{l+m-n} = \sum_{m=0}^{n-k_{\max}-1} \sum_{l=0}^{n-m-k_{\max}-1} 2^{l+m-n}$$

due to the fact that the lower bound goes from 0 to a negative number, which indicates there are no terms. If we explicitly do the last double summation, the previous equation can be rearranged like so.

$$0 \leq \zeta_{j,k} \leq \sum_{p=k_{\max}+1}^{n} \sum_{q=k_{\max}+1}^{p} 2^{-q} = 2^{-k_{\max}} \left(n - k_{\max} - 1\right) + 2^{-n} \tag{3.8}$$

This holds for all j and k.

4 Quantum Fourier state computation (QFS)

4.1 Exact QFS

The quantum Fourier state computation accomplishes the following operation between two registers A and B :

$$|j\rangle_A |\phi(b)\rangle_B \xrightarrow{\text{QFS}_{AB}} |j\rangle_A |\phi(b+j)\rangle_B \tag{4.1}$$

Essentially, we are building a quantum adder in the Fourier basis between two registers. Notice that

$$|\phi(b)\rangle_B = \bigotimes_{l=0}^{n-1} \left(\frac{|0\rangle_B + e^{i\frac{2\pi}{2^{n-l}}b} |1\rangle_B}{\sqrt{2}} \right)$$

and

$$|\phi(b+j)\rangle_B = \bigotimes_{l=0}^{n-1} \left(\frac{|0\rangle_B + e^{i\frac{2\pi}{2^{n-l}}(b+j)} |1\rangle_B}{\sqrt{2}} \right) = \bigotimes_{l=0}^{n-1} \prod_{k=0}^{n-l-1} R_{n-l-k}^{j_k} \left(\frac{|0\rangle_B + e^{i\frac{2\pi}{2^{n-l}}b} |1\rangle_B}{\sqrt{2}} \right)$$

where $R_{n-l-k}^{j_k}$ is a phase gate of parameter $n-l-k$ controlled by qubit j_k. Therefore, 4.1 is implemented like so.

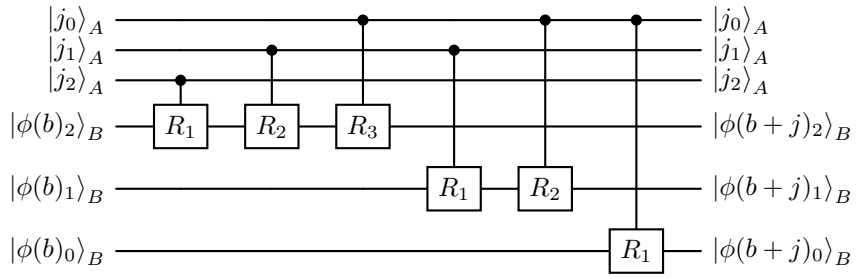


Figure 18: Quantum circuit for the exact QFS_{AB} on $n = 3$ qubits

For our needs, we will look at the case where $b = 0$. Then, the input state in the register B is $|\phi(0)\rangle_B = \text{SWAPs}\left(H^{\otimes n} |0\rangle_B\right)$ and the QFS outputs $|\phi(j)\rangle_B$ in this register.

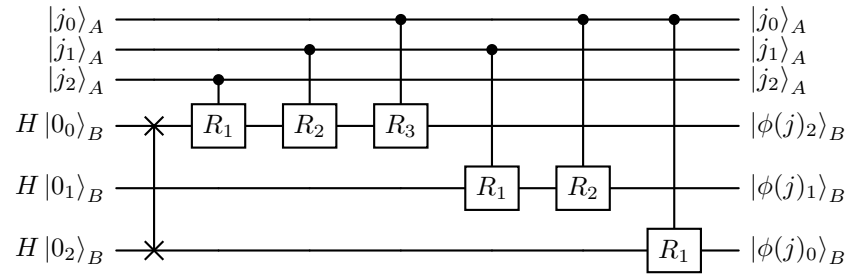


Figure 19: Quantum circuit for the exact QFS_{AB} on $n = 3$ qubits with $b = 0$

If we pass the SWAPs through the circuit, we get a circuit that looks like this.

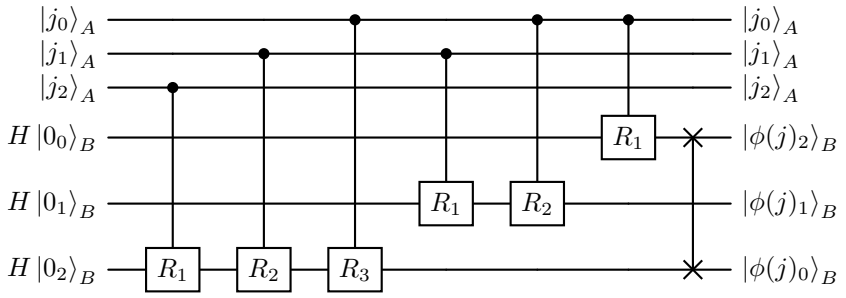


Figure 20: Alternative quantum circuit for the exact QFS_{AB} on $n = 3$ qubits with $b = 0$

Looking at this situation closely, we see that we are performing a QFT where the result of the computation is stored on a separate register than where the input j is. Since figure 20 is essentially a QFT, there are $\mathcal{O}(n^2)$ gates and a depth of $\mathcal{O}(n)$ just like in the regular QFT. The width is now $2n$ because of the two registers and this circuit assumes an all-to-all connectivity.

For a line connectivity, we can mesh the two registers such that $|j_l\rangle$ is above $H|0_l\rangle$ initially. Also, we use additional SWAPs to have controlled operations over qubits that are right next to each other. Finally, the last layer of SWAPs at the end in figure 20 is omitted to preserve a line connectivity. This results in the output being reversed for the register B. In practice, this change of order is fine because if some other operations happen after the QFS, we can design them to take into account this new order.

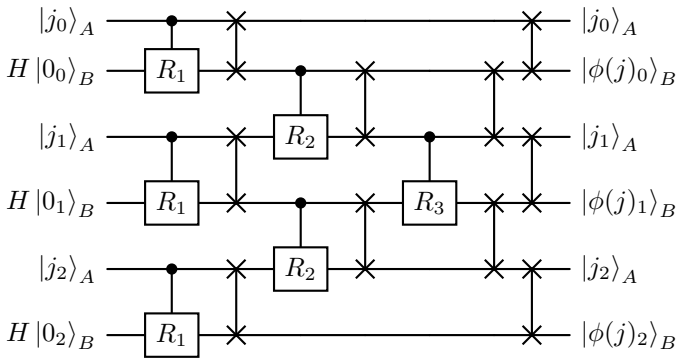


Figure 21: Quantum circuit for the exact QFS_{AB} on a line with the output reversed for $n = 3$ qubits with $b = 0$

For the depth, we have $+1$ for the CR_1 gates, $+(n-1)$ for the remaining phase gates, $+(n-1)$ for the SWAPs interlaced with the phase gates and $+(n-1)$ for the SWAPs at the very end. In total, this is a depth of $3n-2 \implies \mathcal{O}(n)$. The width is still $2n$ and the number of gates is $\mathcal{O}(n^2)$.

4.2 Approximate QFS

In this section, we want to build an approximate version of the QFS for the case $b = 0$. Naturally, we will do the same thing as with the $\text{AQFT}_{k_{\max}}$ and remove phase gates that add a negligible phase. In the implementation on a line, removing some phase gates will also result in SWAPs cancelling each other, giving a more compact circuit.

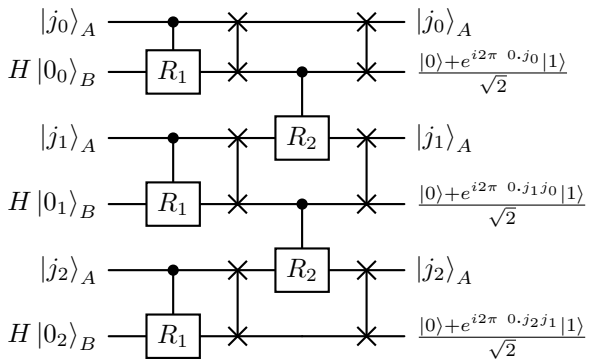


Figure 22: Approximate QFS_{AB} on a line with the output reversed for $n = 3$ qubits, $k_{\max} = 2$ and $b = 0$

Now, suppose that we pick $k_{\max} = \mathcal{O}(\log\left(\frac{n}{\varepsilon}\right))$ where $\frac{1}{\text{poly}(n)} \leq \varepsilon < 1$. Then, 3.8 becomes

$$0 \leq \zeta_{j,k} \leq 2^{-\log(n/\varepsilon)}\left(n - \log(n/\varepsilon) - 1\right) + 2^{-n} = \varepsilon\left(1 - \frac{1}{n}\log(n/\varepsilon) - \frac{1}{n}\right) + 2^{-n} \leq \varepsilon + 2^{-n} \leq 2\varepsilon \tag{4.2}$$

where the last step assumes that $2^{-n} \leq \varepsilon$. This will be true if n is large enough, which is the case when the approximate QFS is most useful. By a similar reasoning as in the previous section, the depth of the circuit in figure 22 is $3k_{\max} - 2 = 3\mathcal{O}(\log\left(\frac{n}{\varepsilon}\right)) - 2 \implies \mathcal{O}(\log\left(\frac{n}{\varepsilon}\right))$. So, with a logarithmic depth, the approximate QFS on a line for the case $b = 0$ can be done with an arbitrary precision $\frac{1}{\text{poly}(n)} \leq \varepsilon < 1$ for the difference in phase between the exact and approximate QFS. It would be more rigorous to use a distance measure betweem the output state of the exact and approximate QFS to compute the precision like in [1]. In that case, with the same conditions, it is also possible to show that the distance can be made arbitrarily close for states in

$$\mathcal{S} = \left\{ |\psi\rangle_{ABE} \in \mathcal{H}_{ABE} : |\psi\rangle_{ABE} = \sum_{j=0}^{2^n-1}\sum_{m=0}^{M-1} \alpha_{j,m}\,|j\rangle_A\,|0\rangle_B\,|m\rangle_E \right\} \tag{4.3}$$

where E is some environment of dimension $M \geq 2^{2n}$ and $|\psi\rangle_{ABE} \in \mathcal{H}_{ABE}$ are pure and normalized states of a tripartite Hilbert space $\mathcal{H}_{ABE}$.

4.3 Special case of the QFS

What happens when the input of the register B is not in the Fourier basis? In other words, what does the QFS yield when the input is $|b\rangle_B$ instead of $|\phi(b)\rangle_B$? One way to find out is the write $|b\rangle_B$ in the Fourier basis and give that as an input to the QFS. Using (2.1),

$$\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1} e^{-i\frac{2\pi}{2^n}bx}\,|\phi(x)\rangle_B = \frac{1}{2^n}\sum_{y=0}^{2^n-1}\sum_{x=0}^{2^n-1} e^{-i\frac{2\pi}{2^n}x(b-y)}\,|y\rangle_B = \frac{1}{N}\sum_{y=0}^{2^n-1} 2^n\delta_{b,y}\,|y\rangle_B = |b\rangle_B \tag{4.4}$$

Then,

$$\text{QFS}\left(|j\rangle_A\,|b\rangle_B\right) = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1} e^{-i\frac{2\pi}{2^n}bx}\text{QFS}\left(|j\rangle_A\,|\phi(x)\rangle_B\right) = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1} e^{-i\frac{2\pi}{2^n}bx}\,|j\rangle_A\,|\phi(x+j)\rangle_B$$

$$= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{-i\frac{2\pi}{2^n}bx} |j\rangle_A \left(\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{i\frac{2\pi}{2^n}(x+j)k} |k\rangle_B \right) = \frac{1}{2^n} \sum_{k=0}^{2^n-1} e^{i\frac{2\pi}{2^n}jk} \sum_{x=0}^{2^n-1} e^{-i\frac{2\pi}{2^n}(b-k)x} |j\rangle_A |k\rangle_B$$

$$= \frac{1}{2^n} \sum_{k=0}^{2^n-1} e^{i\frac{2\pi}{2^n}jb} 2^n \delta_{b,k} |j\rangle_A |k\rangle_B = e^{i\frac{2\pi}{2^n}jb} |j\rangle_A |b\rangle_B \tag{4.5}$$

5 Fourier phase estimation (FPE)

5.1 Exact FPE

The Fourier phase estimation (FPE) performs the following operation between two registers A and B both containing n qubits :

$$|b\rangle_A |\phi(j)\rangle_B \xrightarrow{\text{FPE}_{AB}} |b \oplus j\rangle_A |\phi(j)\rangle_B \tag{5.1}$$

Here, $\oplus$ is the mod 2 addition (XOR) and we will focus on the case $b = j$.

$$|j\rangle_A |\phi(j)\rangle_B \xrightarrow{\text{FPE}_{AB}} |j \oplus j\rangle_A |\phi(j)\rangle_B = |0\rangle_A |\phi(j)\rangle_B \tag{5.2}$$

To do this operation, we could start by applying the inverse QFT on the register B. Then, for all l, use a CNOT to perform a mod 2 addition between the l-th qubit of each register. The l-th qubit in the register A will be the target of that CNOT and the l-th qubit in the register B will control it. Finally, all that is left would be to apply the QFT on the register B to bring it back in the Fourier basis.

$$|j\rangle_A |\phi(j)\rangle_B \xrightarrow{\text{QFT}_B^\dagger} |j\rangle_A |j\rangle_B \xrightarrow{\text{CNOTs}_{A,B}} |j \oplus j\rangle_A |j\rangle_B = |0\rangle_A |j\rangle_B \xrightarrow{\text{QFT}_B} |0\rangle_A |\phi(j)\rangle_B$$

5.2 Approximate FPE

To do 5.2 approximately, we use will the circuit proposed in [1] :

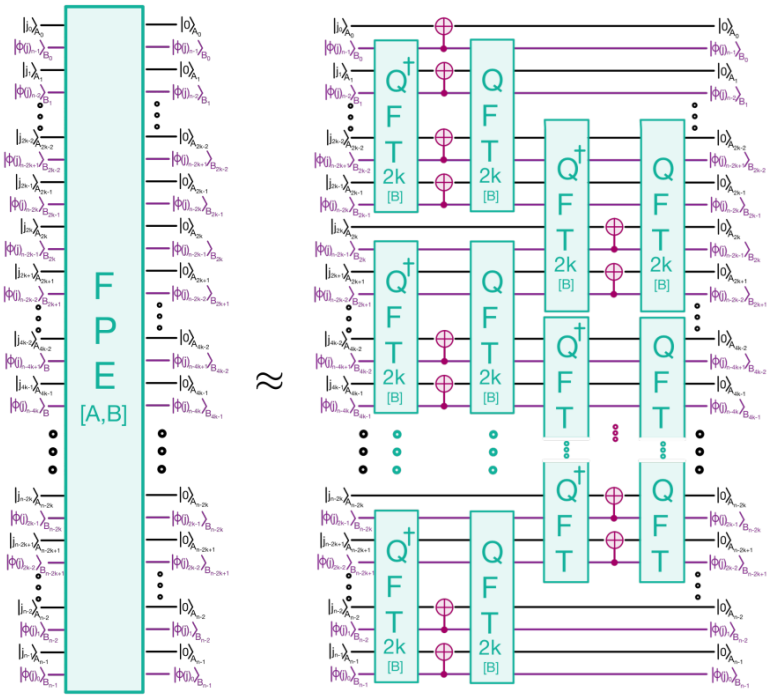


Figure 23: Quantum circuit for the approximate FPE

Rather than applying one big QFT like in the exact FPE, the approximate FPE tries to estimate some of the bits of j on the register B using smaller but exact QFT†s. Again, after that, CNOTs are used to reset the bits of j in the register A that just got estimated on the register B. Finally, smaller but exact QFTs are performed to put the register B back in the Fourier basis. Since only some of the bits of j in the register A are reset, we will need to repeat this operation more than once to completely reset the register A.

5.2.1 Derivation

First of all, suppose that $2k|n$. Then, according to the figure 23, we have to start by performing $\frac{n}{2k}$ QFT†s that are applied on $2k$ qubits of the register B. Therefore, there are $\frac{n}{2k}$ groups of $2k$ qubits which we index by $m \in \{0, ..., \frac{n}{2k} - 1\}$. In each group, the first and last k qubits have indices

$$(\mathbf{q}_m^{'}, \mathbf{q}_m) = ((n - 2km - 1, ..., n - 2km - k), (n - 2km - k - 1, n - 2km - 2k))$$

respectively since the first qubit of $|\phi(j)\rangle_B$ is $|\phi(j)_{n-1}\rangle_B$, then $|\phi(j)_{n-2}\rangle_B$ for the second qubit and so on. So, in the register B, each group corresponds to the state

$$\left|\phi(j)_{(\mathbf{q}'_m,\mathbf{q}_m)}\right\rangle_B = \bigotimes_{l=n-2km-2k}^{n-2km-1} \left(\frac{|0\rangle_B + e^{i2\pi j/2^{n-l}} |1\rangle_B}{\sqrt{2}}\right) = \frac{1}{2^k} \bigotimes_{l=0}^{2k-1} \left(|0\rangle_B + e^{i2\pi j/2^{-l+2km+2k}} |1\rangle_B\right)$$

$$= \frac{1}{2^k} \sum_{r_{2k-1}=0}^{1} ... \sum_{r_0=0}^{1} e^{i2\pi j \sum_{l=0}^{2k-1} r_l 2^l/2^{2km+2k}} |r_{2k-1}...r_0\rangle_B = \frac{1}{2^k} \sum_{r=0}^{2^{2k}-1} e^{i2\pi jr/2^{2km+2k}} |r\rangle_B$$

Then, if we apply a QFT† on $2k$ qubits on this state,

$$\text{QFT}_{2k}^\dagger \left|\phi(j)_{(\mathbf{q}'_m,\mathbf{q}_m)}\right\rangle_B = \frac{1}{2^k} \sum_{r=0}^{2^{2k}-1} e^{i2\pi jr/2^{2km+2k}} \text{QFT}_{2k}^\dagger |r\rangle_B = \frac{1}{2^{2k}} \sum_{x=0}^{2^{2k}-1} \sum_{r=0}^{2^{2k}-1} e^{i\frac{2\pi}{2^{2k}} r(j2^{-2km}-x)} |x\rangle_B$$

$$= \frac{1}{2^{2k}} \sum_{x=0}^{2^{2k}-1} \frac{1 - e^{i2\pi(j2^{-2km}-x)}}{1 - e^{i\frac{2\pi}{2^{2k}}(j2^{-2km}-x)}} |x\rangle_B = \sum_{x=0}^{2^{2k}-1} \gamma_x^{j_m} |x\rangle_B \tag{5.3}$$

where we have used a geometric series. From that point, multiple outcomes are possible. First, if j is a multiple of 2^{-2km}, it results that $j2^{-2km} = j_{n-1}...j_{2km}.0...0 = j_{n-1}...j_{2km}$ is also an integer. In that case,

$$\text{QFT}_{2k}^\dagger \left|\phi(j)_{(\mathbf{q}'_m,\mathbf{q}_m)}\right\rangle_B = \frac{1}{2^{2k}} \sum_{x=0}^{2^{2k}-1} \frac{1 - e^{i2\pi(j_{n-1}...j_{2km}-x_{2k-1}...x_0)}}{1 - e^{i\frac{2\pi}{2^{2k}}(j_{n-1}..j_{2km}-x_{2k-1}...x_0)}} |x\rangle_B$$

$$= \frac{1}{2^{2k}} \sum_{x=0}^{2^{2k}-1} \frac{0}{1 - e^{i2\pi(0.j_{2km+2k-1}...j_{2km}-0.x_{2k-1}...x_0)}} |x\rangle_B = \frac{1}{2^{2k}} \sum_{x=0}^{2^{2k}-1} \frac{0}{1 - e^{i\frac{2\pi}{2^{2k}}(j_{2km+2k-1}...j_{2km}-x)}} |x\rangle_B$$

The term in the sum is 0 when $j_{2km+2k-1}...j_{2km} \neq x$. Otherwise, when $j_{2km+2k-1}...j_{2km} = x$, we can go further back in the equations of 5.3 to treat it separately and find that the final state is $|j_{2km+2k-1}...j_{2km}\rangle$. So, in that case,

$$\text{QFT}_{2k}^\dagger \left|\phi(j)_{(\mathbf{q}'_m,\mathbf{q}_m)}\right\rangle_B = |j_{2km+2k-1}...j_{2km}\rangle_B = \left|(\mathbf{j}_m^{'}, \mathbf{j}_m)\right\rangle_B$$

Since j will always be multiple of 2^{-2km} when $m=0$, we are sure to obtain $|j_{2k-1}...j_0\rangle_B$ on the first $2k$ qubits of the register B. This is why the top left group in figure 23 has CNOTs on all qubits. On the other end, if j is not a multiple of 2^{-2km}, we can't simplify 5.3 further. With that knowledge, we try to estimate the last k bits of the group, that is $(\mathbf{j}'_m)$. For that, rewtite 5.3 so that $(\mathbf{j}'_m)$ is in evidence.

$$\mathrm{QFT}^\dagger_{2k}\left|\phi(j)_{(\mathbf{q}'_m,\mathbf{q}_m)}\right\rangle_B = \sum_{x_0=0}^{2^k-1}\gamma^{j_m}_{(\mathbf{j}'_m,x_0)}\left|(\mathbf{j}'_m,x_0)\right\rangle_B + \sum_{x_1\neq\mathbf{j}'_m}^{2^k-1}\sum_{x_0=0}^{2^k-1}\gamma^{j_m}_{(x_1,x_0)}\left|(x_1,x_0)\right\rangle_B$$

The state on the left is what we aim to get beacause it gives us the last k bits of the group $(\mathbf{j}'_m)$. The state on the right gives a garbage state that we don't care about. Define

$$\sum_{x_0=0}^{2^k-1}\gamma^{j_m}_{(\mathbf{j}'_m,x_0)}\left|x_0\right\rangle_B = \sqrt{1-\varepsilon_{\mathbf{j}'_m}}\left|a_{\mathbf{j}'_m}\right\rangle_B$$

Then,

$$\mathrm{QFT}^\dagger_{2k}\left|\phi(j)_{(\mathbf{q}'_m,\mathbf{q}_m)}\right\rangle_B = \sqrt{1-\varepsilon_{\mathbf{j}'_m}}\left|\mathbf{j}'_m\right\rangle_B\left|a_{\mathbf{j}'_m}\right\rangle_B + \sqrt{\varepsilon_{\mathbf{j}'_m}}\left|\perp^{\mathbf{j}'_m}_{2k}\right\rangle_B$$

tells us we have a probability $1-\varepsilon_{\mathbf{j}'_m}$ of getting the state estimating the last k bits of the group and a probability $\varepsilon_{\mathbf{j}'_m}$ of getting the garbage state on $2k$ qubits which is orthogonal to what is on the left of the sum. With that,

$$\begin{aligned}\left(\mathrm{QFT}^\dagger_{2k}\right)^{\otimes\frac{n}{2k}}_B\left|\phi(j)\right\rangle_B &= \bigotimes_{m=0}^{\frac{n}{2k}-1}\left(\sqrt{1-\varepsilon_{\mathbf{j}'_m}}\left|\mathbf{j}'_m\right\rangle_B\left|a_{\mathbf{j}'_m}\right\rangle_B + \sqrt{\varepsilon_{\mathbf{j}'_m}}\left|\perp^{\mathbf{j}'_m}_{2k}\right\rangle_B\right)\\ &= \sqrt{1-\varepsilon_{\mathbf{j}'}}\bigotimes_{m=0}^{\frac{n}{2k}-1}\left|\mathbf{j}'_m\right\rangle_B\left|a_{\mathbf{j}'_m}\right\rangle_B + \sqrt{\varepsilon_{\mathbf{j}'}}\left|\perp^{\mathbf{j}'}_{n}\right\rangle_B \end{aligned} \tag{5.4}$$

where we have defined $\left|\perp^{\mathbf{j}'}_{n}\right\rangle_B$ to be the resulting garbage state orthogonal to what is on the left of the sum that's now on n qubits and $\varepsilon_{\mathbf{j}'} = 1-\prod_{m=0}^{\frac{n}{2k}-1}\left(1-\varepsilon_{\mathbf{j}'_m}\right)$ as the new probability of obtaining the garbage state. From that, we can apply CNOTs controlled by the last k qubits in each group of the register B to reset the corresponding qubits in the register A like in figure 23.

$$\begin{aligned}\left(\mathrm{QFT}^\dagger_{2k}\right)^{\otimes\frac{n}{2k}}_B\left|j\right\rangle_A\left|\phi(j)\right\rangle_B &= \sqrt{1-\varepsilon_{\mathbf{j}'}}\bigotimes_{m=0}^{\frac{n}{2k}-1}\left|\mathbf{j}'_m\right\rangle_A\left|\mathbf{j}'_m\right\rangle_B\left|a_{\mathbf{j}'_m}\right\rangle_B + \sqrt{\varepsilon_{\mathbf{j}'}}\left|j\right\rangle_A\left|\perp^{\mathbf{j}'}_{n}\right\rangle_B\\ &\xrightarrow{\text{CNOTs}_{A,B}} \sqrt{1-\varepsilon_{\mathbf{j}'}}\bigotimes_{m=0}^{\frac{n}{2k}-1}\left|\mathbf{j}'_m\oplus\mathbf{j}'_m\right\rangle_A\left|\mathbf{j}_m\right\rangle_A\left|\mathbf{j}'_m\right\rangle_B\left|a_{\mathbf{j}'_m}\right\rangle_B + \sqrt{\varepsilon_{\mathbf{j}'}}\left|\perp^{(0,\mathbf{j}')}_{2n}\right\rangle_{A,B}\\ &= \sqrt{1-\varepsilon_{\mathbf{j}'}}\bigotimes_{m=0}^{\frac{n}{2k}-1}\left|0\right\rangle_A\left|\mathbf{j}_m\right\rangle_A\left|\mathbf{j}'_m\right\rangle_B\left|a_{\mathbf{j}'_m}\right\rangle_B + \sqrt{\varepsilon_{\mathbf{j}'}}\left|\perp^{(0,\mathbf{j}')}_{2n}\right\rangle_{A,B}\end{aligned}$$

So,

$$(\mathrm{CNOT})_{A,B}^{\otimes\frac{n}{2k}}\left(\mathrm{QFT}_{2k}^{\dagger}\right)_{B}^{\otimes\frac{n}{2k}}|j\rangle_A|\phi(j)\rangle_B=\sqrt{1-\varepsilon_{\mathbf{j}'}}\bigotimes_{m=0}^{\frac{n}{2k}-1}|0\rangle_A|\mathbf{j}_m\rangle_A\left|\mathbf{j}_m'\right\rangle_B\left|a_{\mathbf{j}_m'}\right\rangle_B+\sqrt{\varepsilon_{\mathbf{j}'}}\left|\perp_{2n}^{(0,\mathbf{j}')}\right\rangle_{A,B}\tag{5.5}$$

Then, figure 23 tells us we have to apply QFT_{2k} on all groups to complete $\mathrm{FPE}_{1/2}^{(\varepsilon)}$ the first half of the approximate FPE.

$$\begin{aligned}\mathrm{FPE}_{1/2}^{(\varepsilon)}|j\rangle_A|\phi(j)\rangle_B&=(\mathrm{QFT}_{2k})_B^{\otimes\frac{n}{2k}}(\mathrm{CNOT})_{A,B}^{\otimes\frac{n}{2k}}\left(\mathrm{QFT}_{2k}^{\dagger}\right)_B^{\otimes\frac{n}{2k}}|j\rangle_A|\phi(j)\rangle_B\\&=\sqrt{1-\varepsilon_{\mathbf{j}'}}(\mathrm{QFT}_{2k})_B^{\otimes\frac{n}{2k}}\bigotimes_{m=0}^{\frac{n}{2k}-1}|0\rangle_A|\mathbf{j}_m\rangle_A\left|\mathbf{j}_m'\right\rangle_B\left|a_{\mathbf{j}_m'}\right\rangle_B+\sqrt{\varepsilon_{\mathbf{j}'}}(\mathrm{QFT}_{2k})_B^{\otimes\frac{n}{2k}}\left|\perp_{2n}^{(0,\mathbf{j}')}\right\rangle_{A,B}\end{aligned}$$

Using 5.4, we can apply $(\mathrm{QFT}_{2k})_B^{\otimes\frac{n}{2k}}$ on both sides to say that

$$\begin{aligned}|\phi(j)\rangle_B&=\sqrt{1-\varepsilon_{\mathbf{j}'}}(\mathrm{QFT}_{2k})_B^{\otimes\frac{n}{2k}}\bigotimes_{m=0}^{\frac{n}{2k}-1}\left|\mathbf{j}_m'\right\rangle_B\left|a_{\mathbf{j}_m'}\right\rangle_B+\sqrt{\varepsilon_{\mathbf{j}'}}(\mathrm{QFT}_{2k})_B^{\otimes\frac{n}{2k}}\left|\perp_n^{\mathbf{j}'}\right\rangle_B\\\implies\sqrt{1-\varepsilon_{\mathbf{j}'}}(\mathrm{QFT}_{2k})_B^{\otimes\frac{n}{2k}}\bigotimes_{m=0}^{\frac{n}{2k}-1}\left|\mathbf{j}_m'\right\rangle_B\left|a_{\mathbf{j}_m'}\right\rangle_B&=|\phi(j)\rangle_B-\sqrt{\varepsilon_{\mathbf{j}'}}(\mathrm{QFT}_{2k})_B^{\otimes\frac{n}{2k}}\left|\perp_n^{\mathbf{j}'}\right\rangle_B\\&=\left(1-\varepsilon_{\mathbf{j}'}\right)|\phi(j)\rangle_B+\sqrt{\varepsilon_{\mathbf{j}'}\left(1-\varepsilon_{\mathbf{j}'}\right)}\left|\perp_n^{\phi(j)}\right\rangle_B\end{aligned}$$

because of the normalization. We can substitute this in $\mathrm{FPE}_{1/2}^{(\varepsilon)}$.

$$\begin{aligned}\mathrm{FPE}_{1/2}^{(\varepsilon)}|j\rangle_A|\phi(j)\rangle_B&=(\mathrm{QFT}_{2k})_B^{\otimes\frac{n}{2k}}(\mathrm{CNOT})_{A,B}^{\otimes\frac{n}{2k}}\left(\mathrm{QFT}_{2k}^{\dagger}\right)_B^{\otimes\frac{n}{2k}}|j\rangle_A|\phi(j)\rangle_B\\&=\left(1-\varepsilon_{\mathbf{j}'}\right)\bigotimes_{m=0}^{\frac{n}{2k}-1}(|0\rangle_A|\mathbf{j}_m\rangle_A)|\phi(j)\rangle_B+\sqrt{\varepsilon_{\mathbf{j}'}\left(1-\varepsilon_{\mathbf{j}'}\right)}\bigotimes_{m=0}^{\frac{n}{2k}-1}(|0\rangle_A|\mathbf{j}_m\rangle_A)\left|\perp_n^{\phi(j)}\right\rangle_B+\sqrt{\varepsilon_{\mathbf{j}'}}(\mathrm{QFT}_{2k})_B^{\otimes\frac{n}{2k}}\left|\perp_{2n}^{(0,\mathbf{j}')}\right\rangle_{A,B}\\&=\left(1-\varepsilon_{\mathbf{j}'}\right)\bigotimes_{m=0}^{\frac{n}{2k}-1}(|0\rangle_A|\mathbf{j}_m\rangle_A)|\phi(j)\rangle_B+\sqrt{2\varepsilon_{\mathbf{j}'}-\varepsilon_{\mathbf{j}'}^2}\left|\perp_{2n}^{\mathrm{FPE}_{1/2}}\right\rangle_B\end{aligned}$$

Now, we need to reset the remaining qubits in the register A which we can done by repeating the operations we just did but shifted down by k qubits. This yields

$$\mathrm{FPE}^{(\varepsilon)}|j\rangle_A|\phi(j)\rangle_B=(1-\varepsilon)|0\rangle_A|\phi(j)\rangle_B+\sqrt{2\varepsilon-\varepsilon^2}\left|\perp_{2n}^{\mathrm{FPE}}\right\rangle_B\tag{5.6}$$

5.2.2 Implementation details

The QFTs/QFT† on $2k$ qubits used in figure 23 can be done on a line with $\mathcal{O}(k^2)$ gates and a depth of $\mathcal{O}(k)$ as shown in section 2. Therefore, the approximate FPE has a linear depth in k. In [1], they show that for $\frac{1}{\mathrm{poly}(n)}\leq\varepsilon<1$, the approximated FPE can be made ε close to the exact FPE with $2n$ qubits on a line with a depth of $\mathcal{O}(\log\left(\frac{n}{\varepsilon^2}\right))$ by choosing $k=\mathcal{O}(\log\left(\frac{n}{\varepsilon^2}\right))$. This is valid for states in

$$\mathcal{T}_{\text{uni}^{(p)}} = \left\{ |\psi\rangle_{ABE} \in \mathcal{H}_{ABE} : |\psi\rangle_{ABE} = \sum_{j=0}^{2^n-1} \sum_{m=0}^{M-1} \beta_{j,m} |j\rangle_A |\phi(j)\rangle_B |m\rangle_E , \left| \sum_m \beta_{j,m} \beta^*_{l,m} \right| \leq \frac{p(n)}{N} \delta_{j,l} \forall j, l \right\} \quad (5.7)$$

where $p(n)$ is a polynomial in n with fixed degree independent of n and $\delta_{j,l}$ is the Kronecker delta.

5.2.3 Bounds

We can bound

$$\left|\gamma_x^{j_m}\right| = \frac{1}{2^{2k}} \frac{\left|1 - e^{i2\pi(j_{2km+2k-1}\cdots j_{2km} \cdot j_{2km-1}\cdots j_0 - x)}\right|}{\left|1 - e^{i\frac{2\pi}{2^{2k}}(j_{2km+2k-1}\cdots j_{2km} \cdot j_{2km-1}\cdots j_0 - x)}\right|} = \frac{1}{2^{2k}} \frac{\sqrt{2 - 2\cos\left(2\pi(j_{2km+2k-1}\cdots j_{2km} \cdot j_{2km-1}\cdots j_0 - x)\right)}}{\sqrt{2 - 2\cos\left(\frac{2\pi}{2^{2k}}(j_{2km+2k-1}\cdots j_{2km} \cdot j_{2km-1}\cdots j_0 - x)\right)}}$$

$$= \frac{1}{2^{2k}} \frac{\sqrt{2\sin^2\left(\pi(j_{2km+2k-1}\cdots j_{2km} \cdot j_{2km-1}\cdots j_0 - x)\right)}}{\sqrt{2\sin^2\left(\frac{\pi}{2^{2k}}(j_{2km+2k-1}\cdots j_{2km} \cdot j_{2km-1}\cdots j_0 - x)\right)}} = \frac{1}{2^{2k}} \frac{\left|\sin\left(\pi(j_{2km+2k-1}\cdots j_{2km} \cdot j_{2km-1}\cdots j_0 - x)\right)\right|}{\left|\sin\left(\frac{\pi}{2^{2k}}(j_{2km+2k-1}\cdots j_{2km} \cdot j_{2km-1}\cdots j_0 - x)\right)\right|}$$

$$\leq \frac{1}{2^{2k}} \frac{1}{\left|\sin\left(\frac{\pi}{2^{2k}}(j_{2km+2k-1}\cdots j_{2km} \cdot j_{2km-1}\cdots j_0 - x)\right)\right|}$$

where some trigonometric identities were employed.

6 $\mathbf{QFT_{uni}}$

In this section, we want to build a QFT that acts on the set of state

$$\mathcal{S}_{\text{uni}}^{(p)} = \left\{ |\psi\rangle_{ABE} \in \mathcal{H}_{ABE} : |\psi\rangle_{ABE} = \sum_{j=0}^{2^n-1} \sum_{m=0}^{M-1} \beta_{j,m} |j\rangle_A |0\rangle_B |m\rangle_E , \left| \sum_m \beta_{j,m} \beta_{l,m}^* \right| \leq \frac{p(n)}{N} \delta_{j,l} \forall j, l \right\} \tag{6.1}$$

which is closely related to $\mathcal{T}_{\text{uni}}^{(p)}$. Notice that $\mathcal{S}_{\text{uni}}^{(p)} \subseteq \mathcal{S}$. To do this operation exaclty given an input $|j\rangle_A |0\rangle_B$ on two registers A and B of n qubits, we could apply the following gates.

$$|j\rangle_A |0\rangle_B \xrightarrow{H} |j\rangle_A |\phi(0)\rangle_B \xrightarrow{\text{QFS}} |j\rangle_A |\phi(j)\rangle_B \xrightarrow{\text{FPE}} |0\rangle_A |\phi(j)\rangle_B \xrightarrow{\text{SWAPs}} |\phi(j)\rangle_A |0\rangle_B$$

For an approximate version, we could use the approximate circuit for all the subroutines used in the exact version. Since the approximate QFS reverses its output, the FPE needs to take that into account by using the reversed implementation found in section 2 for the $\text{QFT}^\dagger$s/QFTs.

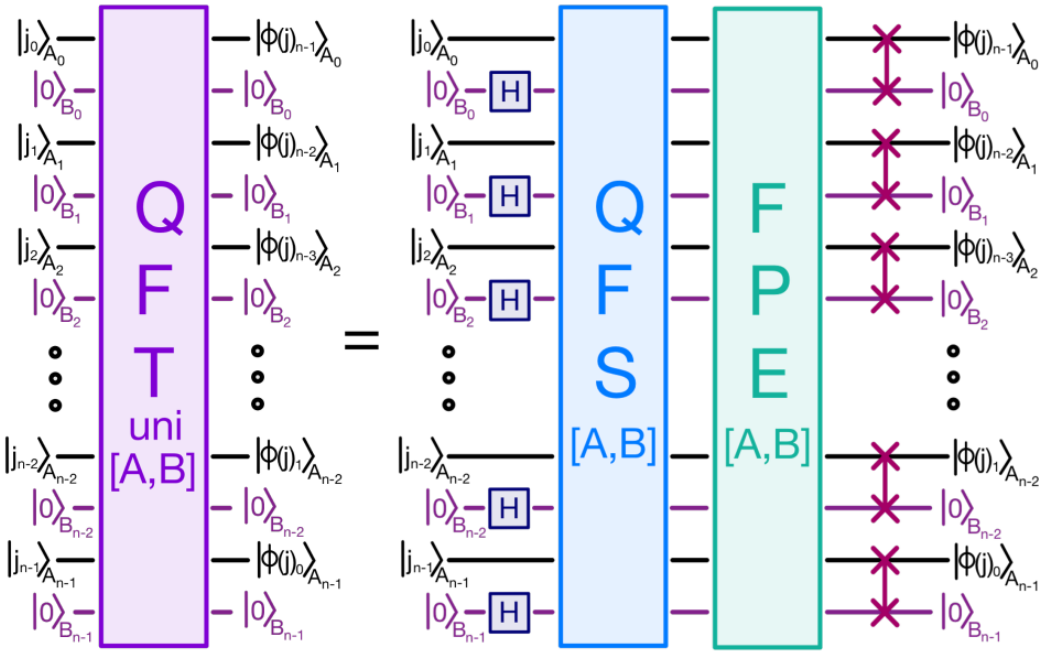


Figure 24: Quantum circuit for the QFT_{uni}

In [1], they show that for $\frac{1}{\text{poly}(n)} \leq \varepsilon < 1$, the approximate QFT_{uni} can be implemented on a line of $2n$ qubits in depth $\mathcal{O}(\log(n/\varepsilon^2))$ for states in $\mathcal{S}_{\text{uni}}$.

7 Optimistic quantum circuits

Often times, in practice, a general algorithm will receive certain inputs more frequently than others on average. For example, this can correspond to realistic inputs in the context of what that algorithm is solving. In that case, it can be profitable to optimize the algorithm specifically for those realistic inputs in order to have a solution that works really well for the average case. Of course, such an adaptation won't perform as great as the general solution for extreme cases. But, since they aren't as relevant as the common inputs, this sacrifice is justified. Naturally, this same idea can be applied to quantum algorithms and one way it occurs is through optimistic quantum circuits. The objective of these quantum circuits is to perform a good approximation of some operation on average.

7.1 Definitions

Given some unitary operation $\mathcal{U}$ and a quantum circuit $\mathcal{C}$ that implements $\tilde{\mathcal{U}}$ on a Hilbert space $\mathcal{H}$, $\mathcal{C}$ is an optimistic circuit for $\mathcal{U}$ with error bound ϵ if, for any complete set of basis state $\{|\phi_j\rangle\}$ of $\mathcal{H}$,

$$\frac{1}{\dim \mathcal{H}} \sum_i \left| \tilde{\mathcal{U}} |\phi_j\rangle - \mathcal{U} |\phi_j\rangle \right|^2 < \epsilon \tag{7.1}$$

In other words, this definition states that, on average for any complete basis, the norm of the error vector between the regular operation $\mathcal{U}$ and the optimistic operation $\tilde{\mathcal{U}}$ most be smaller than ϵ. So, if 7.1 is satisfied, $\tilde{\mathcal{U}}$ will be a good approximation of $\mathcal{U}$ on average. Another equivalent definition, which is basis independent and thus easier to work with, is the following.

$$\frac{\left\| \tilde{\mathcal{U}} - \mathcal{U} \right\|_F^2}{\dim \mathcal{H}} \leq \epsilon \tag{7.2}$$

where $\|X\|_F^2 = \mathrm{Tr}\left[X^\dagger X\right] = \sum_i \sum_j |x_{i,j}|^2$ is the Frobenius norm squared. In a way, the Frobenius norm gives a number which indicates how much a matrix would "stretch" a vector if the matrix was applied to it. So, when dividing it by the dimension of the Hilbert space, we get how much each component is stretched on average. In the context of an optimistic circuit, for it to be a good approximation of $\mathcal{U}$ on average, the average stretch caused by their difference has to be smaller than some error ϵ, which yields 7.2.

7.2 Bad subspace

Consider the subspace $\mathcal{H}_{\text{bad}} \subseteq \mathcal{H}$ with its projector Π_{bad} corresponding to the subspace where the error $\mu = \left\| \left(\tilde{\mathcal{U}} - \mathcal{U} \right) \Pi_{\text{bad}} \right\|_F^2 / \dim \mathcal{H}_{\text{bad}}$ is greater than ϵ. This corresponds to the cases where the approximation resulting from the optimistic circuit is not good enough (not the average cases). Then,

$$\begin{aligned} \epsilon \geq \frac{\left\| \tilde{\mathcal{U}} - \mathcal{U} \right\|_F^2}{\dim \mathcal{H}} \geq \frac{\left\| \left(\tilde{\mathcal{U}} - \mathcal{U} \right) \Pi_{\text{bad}} \right\|_F^2}{\dim \mathcal{H}} = \frac{\dim \mathcal{H}_{\text{bad}}}{\dim \mathcal{H}} \frac{\left\| \left(\tilde{\mathcal{U}} - \mathcal{U} \right) \Pi_{\text{bad}} \right\|_F^2}{\dim \mathcal{H}_{\text{bad}}} = \frac{\dim \mathcal{H}_{\text{bad}}}{\dim \mathcal{H}} \mu \\ \implies \dim \mathcal{H}_{\text{bad}} \leq \dim \mathcal{H} \cdot \frac{\epsilon}{\mu} \end{aligned} \tag{7.3}$$

This tells us that the dimension of $\mathcal{H}_{\text{bad}}$ is inversely proportional to how bad the error is for a given ϵ and $\dim \mathcal{H}$.

7.3 Optimistic QFT

To build an optimistic QFT, we first need to derive an alternative approximate QFT than the one where we truncate phase gates.

7.3.1 Alternative approximate QFT

We start by copying the definition of the QFT.

$$|\phi(x)\rangle = \mathrm{QFT}\,|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{i\frac{2\pi}{2^n}xy}\,|y\rangle = \bigotimes_{l=0}^{n-1}\left(\frac{|0\rangle + e^{i\frac{2\pi}{2^n}x2^l}\,|1\rangle}{\sqrt{2}}\right)$$

Then, we will need to split the register on n qubits into blocks of $m = \mathcal{O}\left(\log\left(\frac{n}{\epsilon}\right)\right)$ qubits. For simplicity, we assume that $m|n$ and we denote the integer on m bits for the j-th block as X_j so that

$$x = \sum_{j=0}^{\frac{n}{m}-1} 2^{mj} X_j = \sum_{j=0}^{\frac{n}{m}-1} 2^{mj} \sum_{k=0}^{m-1} 2^k x_{k+mj} \tag{7.4}$$

So, in terms of the blocks, the QFT is rewritten as

$$|\phi(x)\rangle = \bigotimes_{l=0}^{n-1}\left(\frac{|0\rangle + e^{i\frac{2\pi}{2^n}x2^l}\,|1\rangle}{\sqrt{2}}\right) = \bigotimes_{j=0}^{\frac{n}{m}-1}\left[\bigotimes_{l=0}^{m-1}\left(\frac{|0\rangle + e^{i\frac{2\pi}{2^n}x2^{mj+l}}\,|1\rangle}{\sqrt{2}}\right)\right] = \bigotimes_{j=0}^{\frac{n}{m}-1}\left[\frac{1}{\sqrt{2}^m}\sum_{Y_j=0}^{2^m-1} e^{i\frac{2\pi}{2^n}xY_j2^{mj}}\,|Y_j\rangle\right]$$

$$= \bigotimes_{j=0}^{\frac{n}{m}-1}\left[\frac{1}{\sqrt{2}^m}\sum_{Y_j=0}^{2^m-1} e^{i\frac{2\pi}{2^n}\sum_{k=0}^{\frac{n}{m}-1}2^{mk}2^{mj}X_kY_j}\,|Y_j\rangle\right] = \bigotimes_{j=0}^{\frac{n}{m}-1}\left[\frac{1}{\sqrt{2}^m}\sum_{Y_j=0}^{2^m-1} e^{i\frac{2\pi}{2^n}\sum_{k=0}^{\frac{n}{m}-1}2^{m(k+j)}X_kY_j}\,|Y_j\rangle\right]$$

In the exponential, some terms of the sum will vanish because $e^{i\frac{2\pi}{2^n}2^{m(k+j)}} = 1$ when $m(k+j) \geq n$. We also rearrange the sum.

$$|\phi(x)\rangle = \bigotimes_{j=0}^{\frac{n}{m}-1}\left[\frac{1}{\sqrt{2}^m}\sum_{Y_j=0}^{2^m-1} e^{i\frac{2\pi}{2^n}\sum_{k=0}^{\frac{n}{m}-1}2^{m(k+j)}X_kY_j}\,|Y_j\rangle\right] = \bigotimes_{j=0}^{\frac{n}{m}-1}\left[\frac{1}{\sqrt{2}^m}\sum_{Y_j=0}^{2^m-1} e^{i\frac{2\pi}{2^n}\sum_{k=0}^{\frac{n}{m}-1-j}2^{m(k+j)}X_kY_j}\,|Y_j\rangle\right]$$

$$= \bigotimes_{j=0}^{\frac{n}{m}-1}\left[\frac{1}{\sqrt{2}^m}\sum_{Y_j=0}^{2^m-1} e^{i\frac{2\pi}{2^n}\sum_{k=j}^{\frac{n}{m}-1}2^{n-m-km+mj}X_{\frac{n}{m}-1-k}Y_j}\,|Y_j\rangle\right] = \bigotimes_{j=0}^{\frac{n}{m}-1}\left[\frac{1}{\sqrt{2}^m}\sum_{Y_j=0}^{2^m-1} e^{i\frac{2\pi}{2^m}\sum_{k=j}^{\frac{n}{m}-1}X_{\frac{n}{m}-1-k}Y_j/2^{m(k-j)}}\,|Y_j\rangle\right]$$

By using 3.8 and 4.2, we know that we get an ϵ close general approximation (on all inputs) by only keeping the first two terms of the sum in the exponential. This truncation is allowed due to our choice for m and it reduces the state to

$$|\phi(x)\rangle \approx \bigotimes_{j=0}^{\frac{n}{m}-1}\left[\frac{1}{\sqrt{2}^m}\sum_{Y_j=0}^{2^m-1} e^{i\frac{2\pi}{2^m}Y_j(X_{\frac{n}{m}-1-j}+\frac{X_{\frac{n}{m}-2-j}}{2^m})}\,|Y_j\rangle\right] \tag{7.5}$$

So, the state in each block is

$$|\phi(x)\rangle_j = \frac{1}{\sqrt{2}^m} \sum_{Y_j=0}^{2^m-1} e^{i\frac{2\pi}{2^m} Y_j (X_{\frac{n}{m}-1-j} + \frac{X_{\frac{n}{m}-2-j}}{2^m})} |Y_j\rangle \tag{7.6}$$

To have a quantum circuit that implements 7.5, we will use a sequence of QFTs on m qubits and phase gates on $2m$ qubits. Applying QFT_m on the block $|X_j\rangle$ gives by definition

$$\mathrm{QFT}_m |X_j\rangle = \frac{1}{\sqrt{2}^m} \sum_{Y_j=0}^{2^m-1} e^{i\frac{2\pi}{2^m} X_j Y_j} |Y_j\rangle$$

After that, we would like to add a phase

$$e^{i\frac{2\pi}{2^{2m}} X_{j-1} Y_j} = e^{i\frac{2\pi}{2^{2m}} \sum_{k=0}^{m-1} 2^k x_{k+m(j-1)} \sum_{l=0}^{m-1} 2^l y_{l+mj}} = \prod_{k,l=0}^{m-1} e^{i\frac{2\pi}{2^{2m-k-l}} x_{k+m(j-1)} y_{l+mj}}$$

This can be obtained by the following subroutine which uses controlled phase gates

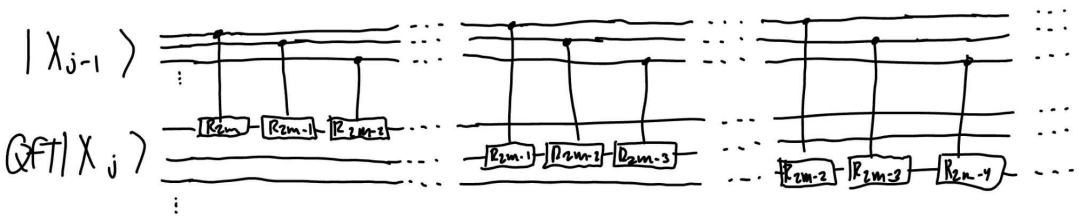


Figure 25: Subroutine for adding the phase $e^{i\frac{2\pi}{2^{2m}} X_{j-1} Y_j}$

and yields the state

$$\frac{1}{\sqrt{2}^m} \sum_{Y_j=0}^{2^m-1} e^{i\frac{2\pi}{2^m} Y_j \left(X_j + \frac{X_{j-1}}{2^m}\right)} |Y_j\rangle$$

If we repeat this process for all blocks starting from the last one up to the top, we get the circuit

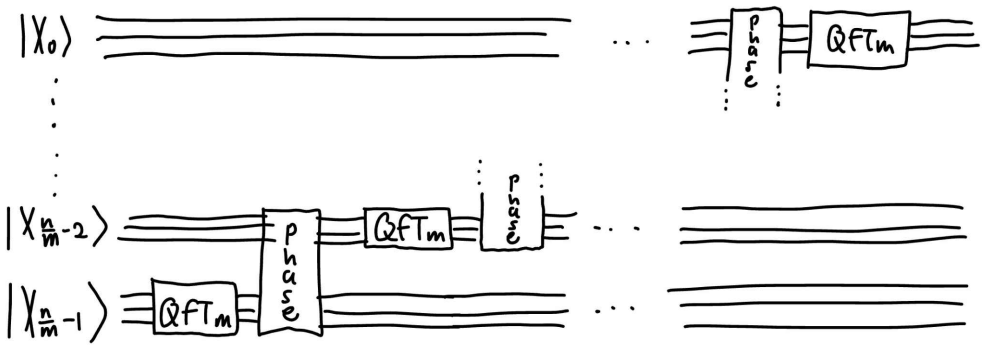


Figure 26: Circuit that almost implements 7.5

which outputs the state

$$\bigotimes_{j=0}^{\frac{n}{m}-1} \left[\frac{1}{\sqrt{2^m}} \sum_{Y_j=0}^{2^m-1} e^{i\frac{2\pi}{2^m} Y_j \left(X_j + \frac{X_{j-1}}{2^m} \right)} |Y_j\rangle \right]$$

This is close to 7.5 and, to obtain this equation, the last step is to reverse the circuit.

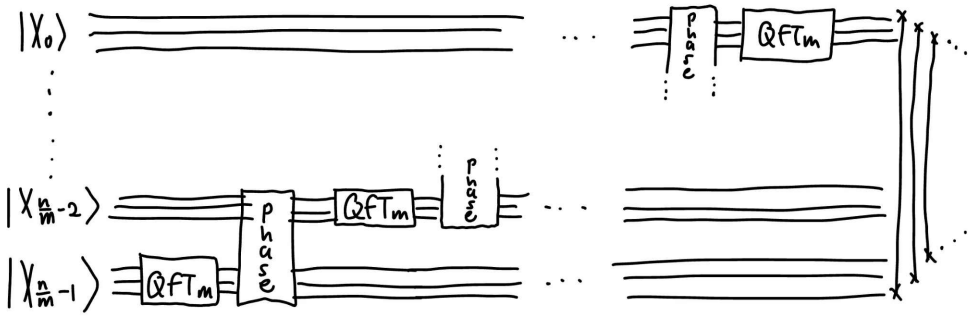


Figure 27: Circuit that implements 7.5

7.3.2 Derivation for the optimistic QFT

...

References

[1] Elisa Bäumer, David Sutter, and Stefan Woerner. Approximate quantum fourier transform in logarithmic depth on a line, 2025.

[2] Elisa Bäumer, Vinay Tripathi, Alireza Seif, Daniel Lidar, and Derek S. Wang. Quantum fourier transform using dynamic circuits, 2024.

[3] Richard Cleve and John Watrous. Fast parallel circuits for the quantum fourier transform, 2000.

[4] D. Coppersmith. An approximate fourier transform useful in quantum factoring, 2002.

[5] Lisa R. Hales. The quantum fourier transform and extensions of the abelian hidden subgroup problem, 2002.

[6] Adam Holmes, Sonika Johri, Gian Giacomo Guerreschi, James S Clarke, and A Y Matsuura. Impact of qubit connectivity on quantum algorithm performance. *Quantum Science and Technology*, 5(2):025009, March 2020.

[7] Gregory D. Kahanamoku-Meyer, John Blue, Thiago Bergamaschi, Craig Gidney, and Isaac L. Chuang. A log-depth in-place quantum fourier transform that rarely needs ancillas, 2025.

[8] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.