Exact quantum Fourier transforms and discrete logarithm algorithms

Michele Mosca[1 2 *] Christof Zalka[1 †]

[1] *Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1*
[2] *St. Jerome's University and Perimeter Institute for Theoretical Physics, Waterloo, Ontario, Canada*

Abstract. We show how the quantum fast Fourier transform (QFFT) can be made exact for arbitrary orders. For most quantum algorithms only the quantum Fourier transform of order 2^n is needed, and this can be done exactly. Kitaev [9] showed how to approximate the Fourier transform for any order. Here we show how his construction can be made exact by using the technique known as "amplitude amplification". This construction e.g. allows to make Shor's discrete logarithm quantum algorithm exact. Thus we have the first example of an exact non black box fast quantum algorithm, thereby giving more evidence that "quantum" need not be probabilistic. We also show that in a certain sense the family of circuits for the exact QFFT is uniform. Namely the parameters of the gates can be calculated efficiently.

Keywords: Quantum Fourier Transform, Discrete Logarithm Problem, Exact algorithms

1 The exact QFFT_p for large prime p

The quantum Fourier transform of order N acts on "computational" basis states $|x\rangle$ as follows:

$$\mathrm{QFFT}_N : \quad |x\rangle \quad \rightarrow \quad |\Psi_x\rangle = \tfrac{1}{N}\sum_{y=0}^{N-1} e^{2\pi i \frac{xy}{N}}|y\rangle.$$

For arbitrary, in particular non-smooth N, Kitaev [9] proposes to do this in two steps (second part of section 5 in [9], see also the review by Jozsa [8]):

$$|x\rangle \quad \rightarrow \quad |x, \Psi_x\rangle \quad \rightarrow \quad |\Psi_x\rangle$$

where, as usual, registers that "appear out of nowhere" are understood to have been initialised in the standard state $|0\rangle$. Similarly in the second step, one of the registers is reset to this state and can thus again be left away.

The first step constructs the Fourier state $|\Psi_x\rangle$ for a given x. This can be done exactly by first obtaining the "uniform amplitude" superposition $|\Psi_0\rangle$ of the first p basis states of a register and then "rephasing" it:

$$|x, 0\rangle \quad \rightarrow \quad |x, \Psi_0\rangle \quad \rightarrow \quad |x, \Psi_x\rangle. \qquad (1)$$

As pointed out by Kitaev, $|\Psi_0\rangle$ can be obtained from $|0\rangle$ by a sequence of SO(2) rotations applied to each qubit.

The second step of Kitaev's construction is the reverse of $|\Psi_x, 0\rangle \rightarrow |\Psi_x, x\rangle$. Kitaev shows how to approximate this transformation through a technique known as "eigenvalue estimation" (see also the article by Cleve et al. [3]), which details how to find the eigenvalue of an unknown eigenstate of some unitary U. Although this operation is not exact, it leaves the eigenstate $|\Psi_x\rangle$ unchanged. Thus it does:

$$|\Psi_x, 0\rangle \quad \rightarrow \quad |\Psi_x\rangle \sum_{x'} c_{x,x'}|x', g_{x,x'}\rangle \qquad (2)$$

where on the right hand side the superposition should be dominated by the term with $x' = x$, such that a measurement would yield x with good probability. We also included some (unwanted) "garbage" $g_{x,x'}$ which may be produced along with the eigenvalue. In the next sections we show how we can make this part of the algorithm exact using "amplitude amplification" [1] to eliminate all but the desired term $|x, g_{x,x}\rangle$.

We let the operator A correspond to the operation on the right hand register in eq. 2 where the state $|\Psi_x\rangle$ will be treated as a "spectator" that is not changed. We will modify A so that its success probability is reduced to $1/4$ (so that a single iteration of amplitude amplification leads exactly to the desired state). The main tools necessary for applying amplitude amplification are recognising the correct solution and knowing the success probability so that we may reduce it to exactly $1/4$.

1.1 "Recognising" the correct solution

Amplitude amplification requires a way to "recognise" the good states (i.e. apply the phase -1 to them and leave the orthogonal complement unchanged). We must check whether a number x' is the correct eigenvalue of $|\Psi_x\rangle$ (i.e. whether $x' = x$). This can be done because the eigenstate $|\Psi_x\rangle$ is still available exactly. Thus given a state of the form $|\Psi_x\rangle \sum_{x'} c_{x,x'}|x', g_{x,x'}\rangle$, we can check the second register against the first one by applying the reverse of the steps in eq. 1 to these two registers:

$$|x', \Psi_x\rangle \quad \rightarrow \quad |x', \Psi_{x-x'}\rangle \quad \rightarrow \quad |x', \theta_{x-x'}\rangle$$

where in the second step we only act on the second register. The state $|\Psi_0\rangle$ is mapped back to $|0\rangle$, while for $x' \neq x$ we get some state $|\theta_{x-x'}\rangle$ orthogonal to $|0\rangle$. We can now apply the phase -1 to the $|0\rangle$ state and undo the previous operations.

1.2 "Uniformising" the success probability

To use amplitude amplification to make algorithms exact the success probability of the "heuristic" algorithm A must be known. In our case the success probability of eigenvalue estimation on $|\Psi_x\rangle$ depends on the (unknown) value of x. We fix this problem by modifying A such that the new success probability will become independent of x and equal to the average over all instances for the original A. To do this uniformisation we pick an integer r uniformly at random from $\{0, 1, \ldots p-1\}$ (quantumly, i.e. by preparing a uniform superposition of these values) and replace $|\Psi_x\rangle$ with $|\Psi_{x+r}\rangle$, which is just a rephasing. We keep a record of r and subtract it again from the result of eigenvalue estimation.

So now exact amplitude amplification will allow us to do

$$|\Psi_x, 0\rangle \quad \rightarrow \quad |\Psi_x\rangle|x, g_{x,x}\rangle.$$

To get rid of the "garbage" we can do the usual trick

*mmosca@iqc.ca
†zalka@iqc.ca

of copying the wanted result x into an additional "save" register and then undoing the previous steps. In total this will lead to six applications of A for an exact QFFT.

1.3 Eigenvalue estimation

In the eigenvalue estimation phase we use a Fourier transform to estimate the eigenvalue $e^{-2\pi ix/p}$ of U for eigenstate $|\Psi_x\rangle$, where U acts on computational basis states as: $|x\rangle \rightarrow |(x+1) \bmod p\rangle$. By carefully post-selecting which y's we keep, the instance independent success probability of the uniformised algorithm is:

$$\bar{p} = \frac{1}{p}\sum_{k=0}^{p-1} f^2(k/p) \qquad \text{where} \quad f(z) = \frac{\sin(\pi z)}{N\sin(\pi z/N)}$$

and we have used that N and p are coprime and so for each x there is exactly one k.

1.3.1 Calculate and adjust success probability

Using this exact description of the success probability $\overline{p}$ for a given p, we modify the algorithm A so that it will succeed exactly with probability $1/4$. One way to do this is to add a qubit prepared in state $\cos(\alpha)|0\rangle + \sin(\alpha)|1\rangle$ with $\overline{p}\sin^2(\alpha) = \frac{1}{4}$ and additionally require for success that this qubit be in state $|1\rangle$. The preparation of this qubit will now require one "strange" gate in our algorithm, although its rotation angle α can be computed efficiently in the following sense. We can show that for each p and N, the success probability can be approximated efficiently in the sense that the computation time is polynomial in the number of digits we want to compute.

2 An exact discrete logarithm algorithm

An exact algorithm for the QFFT leads in a straightforward manner to an exact version of the discrete logarithm algorithm (see [11, 12] for the bounded-error case) of the same order. This was also observed for finite fields of prime order by Brassard and Høyer [2] (Theorem 12). For smooth orders (only small prime factors) the problem can easily be solved classically. We don't have room to review this here, but details are provided in the full version of this paper.

3 Further remarks and observations

The construction of the exact QFFT_q easily generalises to arbitrary orders q. Also the discrete logarithm algorithm can be generalised to arbitrary orders q. We also give a more involved solution for the case when the factorisation of the order q is not known. In our construction we take care not to introduce new "special" gates during the computation. This means that really the $O(\log q)$ quantum runs can be put together into one quantum circuit whose gates can be computed from q alone (without knowing its factorisation).

Let us also note that it is not clear how to make Shor's integer factorisation algorithm exact with the techniques used here. Thus this is a challenge that remains. We note that Mosca [10] shows how to make factorisation exact in a slightly generalised model of exact quantum computation.

3.1 Review of other work on the QFFT

It is interesting to note that after Kitaev [9] a more efficient and probably also more natural way to approximate the QFFT for arbitrary orders has been given by Hallgren and Hales [7]. In particular their construction uses fewer qubits, but it seems not to lend itself to the techniques used here to make it exact. Also note the simplified "semiclassical" version of the standard QFFT by Griffiths and Niu [6].

Acknowledgements

Ch.Z. would like to thank D.M. Jackson for discussions on summing the m^{th} powers of the first n integers. He is supported by CSE and MITACS. M.M. thanks R. Cleve, L. Hales, and J. Watrous for discussions at MSRI. M.M. holds a Canada Research Chair in Quantum Computation and is supported by NSERC, MITACS, CFI, ORDCF, and PREA.

References

[1] G. Brassard, P. Høyer and A. Tapp, *Quantum Counting*, ICALP'98.

[2] G. Brassard and P. Høyer, *An Exact Quantum Polynomial-Time Algorithm for Simon's Problem*, ISTCS'97.

[3] R. Cleve, A. Ekert, C. Macchiavello and M. Mosca, *Quantum Algorithms revisited*, Proc. R. Soc. Lond. A (1998) **454**, pp. 339-354.

[4] R. Cleve, *A note on computing Fourier transforms by quantum programs* Unpublished (1994) (`http://pages.cpsc.ucalgary.ca/~cleve/papers.html`)

[5] D. Coppersmith, IBM Research Report RC 19642 (1994) (also quant-ph/0201067).

[6] R. B. Griffiths and C. Niu, *Semiclassical Fourier Transform for Quantum Computation*, Phys. Rev. Lett. **76** (1996) pp.3228-3231.

[7] S. Hallgren and L. Hales, *An Improved Quantum Fourier Transform Algorithm and Applications*, FOCS 2000.

[8] R. Jozsa, *Quantum Algorithms and the Fourier Transform*, Proc. R. Soc. Lond. A (1998) **454**, pp. 323-337.

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

[10] M. Mosca, *On the Quantum Derandomization of Algorithms*, manuscript in preparation; based on presentation at MSRI QIP workshop, Dec. 2002.

[11] P. Shor, *Algorithms for Quantum Computation: Discrete Logarithms and Factoring*, FOCS 1994.

[12] D. Boneh and R. J. Lipton, *Quantum Cryptanalysis of Hidden Linear Functions, Advances in Cryptology*, CRYPTO 95.