Quantum Inf Process (2017) 16:82
DOI 10.1007/s11128-017-1515-0

Quantum Fourier transform in computational basis

S. S. Zhou[1,2] · T. Loke[1] · J. A. Izaac[1] · J. B. Wang[1]

Received: 4 September 2016 / Accepted: 3 January 2017
© Springer Science+Business Media New York 2017

Abstract The quantum Fourier transform, with exponential speed-up compared to the classical fast Fourier transform, has played an important role in quantum computation as a vital part of many quantum algorithms (most prominently, Shor's factoring algorithm). However, situations arise where it is not sufficient to encode the Fourier coefficients within the quantum amplitudes, for example in the implementation of control operations that depend on Fourier coefficients. In this paper, we detail a new quantum scheme to encode Fourier coefficients in the computational basis, with fidelity $1-\delta$ and digit accuracy ϵ for each Fourier coefficient. Its time complexity depends polynomially on $\log(N)$, where N is the problem size, and linearly on $1/\delta$ and $1/\epsilon$. We also discuss an application of potential practical importance, namely the simulation of circulant Hamiltonians.

Keywords Quantum algorithm · Quantum Fourier transform · Computational basis state · Controlled quantum gates

1 Introduction

Since the milestone introduction of Shor's quantum factoring algorithm [1] allows prime number factorization with complexity $\mathcal{O}(\text{polylog}N)$—an exponential speed-up compared to the fastest known classical algorithms—there has been an increasing number of quantum algorithm discoveries harnessing the unique properties of quantum mechanics in order to achieve significant increases in computational efficiency. The use

✉ J. B. Wang
jingbo.wang@uwa.edu.au

1 School of Physics, The University of Western Australia, Crawley, WA 6009, Australia

2 Department of Physics, Yale University, New Haven, CT 06520, USA

Published online: 10 February 2017

of the quantum Fourier transform (QFT) [2] in Shor's factoring algorithm is integral to the resulting speed-up.

The fast Fourier transform (FFT), an efficient classical implementation of the discrete Fourier transform (DFT), is a hugely important algorithm, with classical uses including signal processing and frequency analysis [3]. Due to its ubiquity and efficiency (with scaling $\mathcal{O}(N \log N)$), it has been regarded to be one of the most important non-trivial classical algorithms [4].

The QFT [with complexity $\mathcal{O}((\log N)^2)$] algorithm is the natural extension of the DFT to the quantum regime, with exponential speed-up realized compared to the FFT ($\mathcal{O}(N \log N)$), due to superposition and quantum parallelism. The QFT is essentially identical to the FFT in that it performs a DFT on a list of complex numbers, but the result of the QFT is stored as amplitudes of a quantum state vector. In order to extract the individual Fourier components, measurements need to be performed on the quantum state vector. As such, the QFT is not directly useful for determining the Fourier-transformed coefficients of the original list of numbers. However, the QFT is widely used as a subroutine in larger algorithms, including but not limited to Shor's algorithm [1], quantum amplitude estimation [5] and quantum counting [6,7].

Typically, there are two methods of encoding the result of a quantum algorithm: encoding within the computational basis of the quantum state [5] and encoding within the amplitudes of the quantum state [2]. The QFT fits the latter category and has been successfully used as a foundation for a plethora of other quantum algorithms—for example in the fields of quantum chemistry and simulations [8–10], signal and image processing [11,12], cryptography [13] and computer science [4,14]. However, situations arise where we need the Fourier coefficients in the computational basis, for example in order to efficiently implement circulant Hamiltonians with quantum circuits [15].

In this paper, we introduce a new quantum scheme for computing the Fourier transform and storing the results in the computational basis, namely quantum Fourier transform in the computational basis (QFTC). We begin in Sect. 2 by defining the notations and chosen conventions, before detailing the QFTC algorithm for computing the DFT in the computational basis in Sect. 3. This section also includes a thorough analytic derivation of the complexity and error analysis. One possible application of this algorithm, the implementation of circulant Hamiltonians, is then discussed in Sect. 5. In addition, we have provided supplementary material in the appendices, detailing the quantum arithmetic necessary for the QFTC algorithm in Appendix 1 and the implementation of circulant matrix operators in Appendix 2.

2 Definitions and notations

The DFT, applied to a unit vector $\boldsymbol{x} = (x_0\ x_1\ \cdots\ x_{N-1}) \in \mathbb{C}^N$, outputs a unit vector $\boldsymbol{y} = (y_0\ y_1\ \cdots\ y_{N-1})$, where

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \mathrm{e}^{2\pi i j k/N} x_j, \quad k = 0, 1, \ldots, N-1. \tag{1}$$

In the following sections, we assume that $N = 2^L$, where L is some integer, as in the conventional FFT and QFT algorithms. The QFT performs the discrete Fourier

transform in amplitudes:

$$\sum_{j=0}^{N-1} x_j \, |j\rangle \rightarrow \sum_{k=0}^{N-1} y_k \, |k\rangle \, . \tag{2}$$

The QFTC, on the other hand, enables the Fourier-transformed coefficients to be encoded in the computational basis:

$$|k\rangle \xrightarrow{\text{QFTC}} |k\rangle \, |y_k\rangle \tag{3}$$

where $|y_k\rangle$ corresponds to the fixed-point binary representation of $y_k \in (-1, 1)$ using two's complement format. Without loss of generality, we will assume the y_k coefficients are real in the following sections. If this is not the case, we can always redefine the inputs as the following:

$$x'_j = \frac{x_j + x^*_{N-j}}{2} \quad \text{(where } x_N = x_0 \text{ and } x^*_j \text{ is the complex conjugate of } x_j\text{)} \tag{4}$$

for all j. Applying the DFT to $\boldsymbol{x}'$ then produces a purely real result, $y'_k = \text{Re}(y_k)$. The imaginary components $\text{Im}(y_k)$ can be derived analogously, by applying the DFT to

$$x'_j = \frac{x_j - x^*_{N-j}}{2}. \tag{5}$$

In the proposed QFTC algorithm, the input vector $\boldsymbol{x}$ is provided by an oracle O_x such that

$$O_x \, |0\rangle = \sum_{j=0}^{N-1} x_j \, |j\rangle \, , \tag{6}$$

which can be efficiently implemented if $\boldsymbol{x}$ is efficiently computable [16,18] or by using the qRAM that takes complexity $\log N$ under certain conditions [17,19–22]. The number of calls to O_x and $O_x^\dagger$ will be included in the overall complexity of the QFTC algorithm. It is worth noting that this algorithm would not work if we do not know how the input vector $\boldsymbol{x}$ is generated.

3 Quantum Fourier transform in the computational basis

The steps involved in the QFTC algorithm are detailed below (with Fig. 1 depicting the circuit for *Step 1–Step 5* and Fig. 3 for *Step 6–Step 10*). We use 14 registers in our algorithm labelled A, B_1, B_1', B_2, B_2', C, C', D, D', E, E', F, F' and G, among which Reg A stores the subscript k in the Fourier coefficients, Reg G stores the value of y_k, and others are all ancillas. There are $p_0 + 1$ qubits in Reg G (meaning accuracy $\epsilon = 2^{-p_0}$).

Step 0 Initialize all qubits, including ancillas, to $|0\rangle$.

Step 1 Prepare Reg A of L qubits into a superposition of its computational basis states. Here we take $|k\rangle$ as an example:

$$\left|0^L\right\rangle \rightarrow |k\rangle\,, \tag{7}$$

where k is represented in binary as $k_1 k_2 \cdots k_L$ with L qubits. Note that subsequent steps can be trivially extended for arbitrary linear combinations, for example of the form $\sum_k u_k\,|k\rangle$.

Step 2 Prepare an ancillary qubit in Reg B_1 as:

$$|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle). \tag{8}$$

Step 3 Apply O_x to Reg B_2 of L qubits controlled by Reg B_1:

$$\left|0^L\right\rangle \frac{1}{\sqrt{2}}\big(\,|1\rangle + |0\rangle\,\big) \xrightarrow{O_x\otimes|1\rangle\langle 1|+\mathbb{I}\otimes|0\rangle\langle 0|} \frac{1}{\sqrt{2}}\left(\sum_{j=0}^{N-1} x_j\,|j\rangle\,|1\rangle + \left|0^L\right\rangle|0\rangle\right), \tag{9}$$

where j is represented in binary as $j_1 j_2 \cdots j_L$ with L digits.

Step 4 Apply $H^{\otimes L}$ to Reg B_2 of L qubits controlled by Reg B_1:

$$\begin{aligned}&\frac{1}{\sqrt{2}}\left(\sum_{j=0}^{N-1} x_j\,|j\rangle\,|1\rangle + \left|0^L\right\rangle|0\rangle\right) \xrightarrow{H^{\otimes L}\otimes|0\rangle\langle 0|+|+\rangle\langle +|\otimes|1\rangle\langle 1|} \\ &\sum_{j=0}^{N-1}\frac{1}{\sqrt{2}}\left(x_j\,|j\rangle\,|1\rangle + \frac{1}{\sqrt{N}}\,|j\rangle\,|0\rangle\right).\end{aligned} \tag{10}$$

Step 5 Apply a controlled phase operator on Reg A, B_1, B_2 (with details given in Fig. 1b):

$$\begin{aligned}&|k\rangle\sum_{j=0}^{N-1}\frac{1}{\sqrt{2}}\left(x_j\,|j\rangle\,|1\rangle + \frac{1}{\sqrt{N}}\,|j\rangle\,|0\rangle\right) \\ &\xrightarrow{\left(\sum_{j,k'} \mathrm{e}^{2\pi i j k'/N}|k'\rangle\langle k'|\otimes|j\rangle\langle j|\right)\otimes|1\rangle\langle 1|+\mathbb{I}\otimes|0\rangle\langle 0|} |k\rangle\,|\phi_k\rangle\,,\end{aligned} \tag{11}$$

in which we define $|\phi_k\rangle := \frac{1}{\sqrt{2}}\big(x_j \mathrm{e}^{2\pi i j k/N}\,|j\rangle\,|1\rangle + \frac{1}{\sqrt{N}}\,|j\rangle\,|0\rangle\,\big)$ for simplicity. The function of the controlled phase operator is to add a phase factor $\mathrm{e}^{2\pi i j k/N}$ to the quantum state $|k\rangle\,|j\rangle\,|1\rangle$ for arbitrary k and j and leave it unchanged when the ancillary qubit is $|0\rangle$.

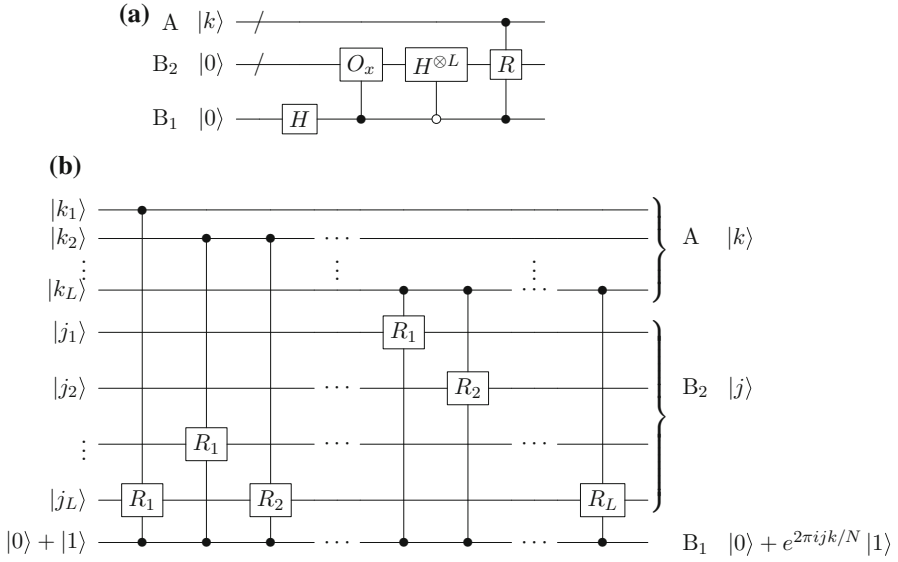

Fig. 1 **a** Quantum circuit for *Step 1–Step 5*; **b** Detailed quantum gates to implement the controlled phase operator in *Step 5*. Here $R_\ell = |0\rangle\langle 0| + \mathrm{e}^{2\pi i/2^\ell}|1\rangle\langle 1|$

Using the Hadamard gate and the pauli-Z gate, we can prepare Reg C, C' in the quantum states $|\phi^\pm\rangle$:

$$\begin{aligned}&\left|0^{L+1}\right\rangle \xrightarrow{(+):\, H^{\otimes L}\otimes H;\ (-):\, H^{\otimes L}\otimes ZH} \left|\phi^\pm\right\rangle \\ &= \frac{1}{\sqrt{2}}\left(\sum_{j=0}^{N-1}\frac{\pm 1}{\sqrt{N}}|j\rangle|1\rangle + \sum_{j=0}^{N-1}\frac{1}{\sqrt{N}}|j\rangle|0\rangle\right). \end{aligned} \tag{12}$$

We have

$$\left|\langle\phi^\pm|\phi_k\rangle\right|^2 = \frac{1}{4}(y_k^2+1) \pm \frac{y_k}{2}, \tag{13}$$

and

$$\left|\langle\phi^+|\phi_k\rangle\right|^2 - \left|\langle\phi^-|\phi_k\rangle\right|^2 = y_k, \tag{14}$$

which leads to the following steps (as detailed in Fig. 3).

Step 6 Prepare $|\phi^+\rangle$ in Reg C and perform the swap test (Fig. 2) with $|\phi_k\rangle$ in Reg B (= B1 + B2). We get

$$\left|\psi_k^+\right\rangle = \frac{1}{2}|0\rangle\left(|\phi_k\rangle\left|\phi^+\right\rangle + \left|\phi^+\right\rangle|\phi_k\rangle\right) + \frac{1}{2}|1\rangle\left(|\phi_k\rangle\left|\phi^+\right\rangle - \left|\phi^+\right\rangle|\phi_k\rangle\right). \tag{15}$$

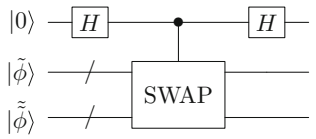

Fig. 2 Swap test. Here $\boxed{\text{SWAP}}\left|\tilde{\phi}\right\rangle\left|\tilde{\tilde{\phi}}\right\rangle=\left|\tilde{\tilde{\phi}}\right\rangle\left|\tilde{\phi}\right\rangle$. The probability to finally obtain $|0\rangle$ and $|1\rangle$ in the first register is $(1/2)\left(1+|\left\langle\tilde{\phi}|\tilde{\tilde{\phi}}\right\rangle|^2\right)$ and $(1/2)\left(1-|\left\langle\tilde{\phi}|\tilde{\tilde{\phi}}\right\rangle|^2\right)$, respectively. This procedure is often utilized to estimate the inner product of two quantum states $\left|\tilde{\phi}\right\rangle$ and $\left|\tilde{\tilde{\phi}}\right\rangle$ [23]

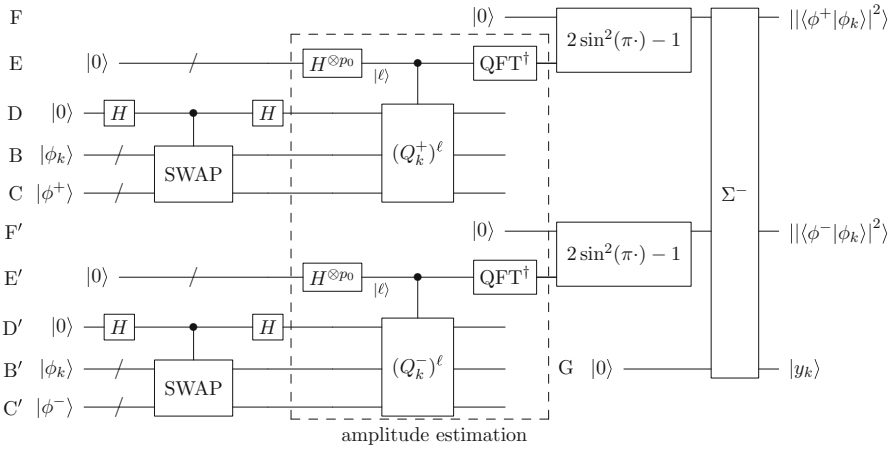

Fig. 3 Quantum circuit for *Step 6*–*Step 10*. The $\boxed{\Sigma^-}$ gate transforms $|\alpha\rangle\,|\beta\rangle\,|0\rangle$ into $|\alpha\rangle\,|\beta\rangle\,|\alpha-\beta\rangle$ (see Appendix 1)

Step 7 Run amplitude estimation for all k on state $\left|\psi_k^+\right\rangle$ and store the phases in Reg E:

$$\left|\psi_k^+\right\rangle\rightarrow\left|\frac{\theta_k}{\pi}\right\rangle\left|\psi_k^{\uparrow}\right\rangle+\left|1-\frac{\theta_k}{\pi}\right\rangle\left|\psi_k^{\downarrow}\right\rangle, \tag{16}$$

where $\left|\psi_k^+\right\rangle$ can be decomposed into the sum of $\left|\psi_k^{\uparrow}\right\rangle$ and $\left|\psi_k^{\downarrow}\right\rangle$ which are a pair of un-normalized orthogonal bases (corresponding to two distinct phases in the amplitude estimation procedure detailed below).

Step 8 Compute $\left|\left\langle\phi^+|\phi_k\right\rangle\right|^2=(y_k^2+1)/4+y_k/2$ using the quantum multiply–adder and sine gate (see Appendix 1 for details), for all values of k:

$$\left|\frac{\theta_k}{\pi}\right\rangle\left|\psi_k^{\uparrow}\right\rangle+\left|1-\frac{\theta_k}{\pi}\right\rangle\left|\psi_k^{\downarrow}\right\rangle\rightarrow\left|\left|\left\langle\phi^+|\phi_k\right\rangle\right|^2\right\rangle\left(\left|\frac{\theta_k}{\pi}\right\rangle\left|\psi_k^{\uparrow}\right\rangle+\left|1-\frac{\theta_k}{\pi}\right\rangle\left|\psi_k^{\downarrow}\right\rangle\right), \tag{17}$$

where the value of $\left|\left\langle\phi^+|\phi_k\right\rangle\right|^2=2\sin^2\theta_k-1$ is stored in Reg F.

In the above description of *Step 6–Step 10*,

$$\left|\psi_k^+\right\rangle = \sin\theta_k \left|\psi_k^0\right\rangle + \cos\theta_k \left|\psi_k^1\right\rangle \tag{18}$$

where $\left|\psi_k^0\right\rangle$ corresponds to the part of $\left|\psi_k^+\right\rangle$ whose first qubit is $|0\rangle$, $\left|\psi_k^1\right\rangle$ corresponds to the part of $\left|\psi_k^+\right\rangle$ whose first qubit is $|1\rangle$. We can choose $\theta_k \in [0, \pi/2]$ without loss of generality. It can be easily calculated from Eq. 15 that $\sin^2\theta_k = (1 + \left|\langle\phi^+|\phi_k\rangle\right|^2)/2$. We define $Q_k^+ := -\mathcal{A}_k^+ S_0 (\mathcal{A}_k^+)^\dagger S_\chi$, where $\mathcal{A}_k^+$ is the unitary operator performing $|0\rangle_{\mathrm{DBC}} \xrightarrow{\mathcal{A}_k^+} \left|\psi_k^+\right\rangle$, $S_0 = \mathbb{I} - 2\,|0\rangle_{\mathrm{DBC}}\langle 0|_{\mathrm{DBC}}$ and $S_\chi = \mathbb{I} - 2\,|0\rangle_{\mathrm{D}}\langle 0|_{\mathrm{D}}$ (subscripts denote labels of registers). According to the amplitude estimation algorithm [7],

$$\left(Q_k^+\right)^\ell \left|\psi_k^+\right\rangle = \sin(2\ell+1)\theta_k \left|\psi_k^0\right\rangle + \cos(2\ell+1)\theta_k \left|\psi_k^1\right\rangle. \tag{19}$$

For any $\ell \in \mathbb{N}$, Q_k^+ acts as a rotation in two-dimensional space $\mathrm{Span}\{\left|\psi_k^0\right\rangle, \left|\psi_k^1\right\rangle\}$, and it has eigenvalues $\mathrm{e}^{\pm i2\theta_k}$ with eigenstates $\left|\psi_k^{\uparrow,\downarrow}\right\rangle$ (un-normalized). Therefore, we can generate the state

$$\left|\psi_k^+\right\rangle = \left|\psi_k^\uparrow\right\rangle + \left|\psi_k^\downarrow\right\rangle \xrightarrow{\text{phase estimation}} \left|\frac{\theta_k}{\pi}\right\rangle\left|\psi_k^\uparrow\right\rangle + \left|1 - \frac{\theta_k}{\pi}\right\rangle\left|\psi_k^\downarrow\right\rangle, \tag{20}$$

by running amplitude estimation of $\mathcal{A}_k^+$ on $\left|\psi_k^+\right\rangle$ and obtain $\left|\left|\langle\phi^+|\phi_k\rangle\right|^2\right\rangle = \left|2\sin^2\theta_k - 1\right\rangle$ using the quantum multiply–adder and sine gate (see Appendix 1). The quantum circuit of amplitude estimation procedure is shown in Fig. 3.

Step 9 Repeat *Step 2–Step 8* in Reg B', C' ,E', F', with $\left|\phi^+\right\rangle$ and $\mathcal{A}_k^+$ replaced by $\left|\phi^-\right\rangle$ and $\mathcal{A}_k^-$, we obtain

$$\left|\left|\langle\phi^+|\phi_k\rangle\right|^2\right\rangle|0\rangle \to \left|\left|\langle\phi^+|\phi_k\rangle\right|^2\right\rangle\left|\left|\langle\phi^-|\phi_k\rangle\right|^2\right\rangle \tag{21}$$

in Reg F, F', where the quantum states in Reg A, B, B', C, C', E, E' are not written out explicitly for simplicity, because they remain unchanged in the following steps.

Step 10 Calculate $\left|\langle\phi^+|\phi_k\rangle\right|^2$ minus $\left|\langle\phi^-|\phi_k\rangle\right|^2$ and encode the result in Reg G, using the quantum adder described in Appendix 1:

$$\left|\left|\langle\phi^+|\phi_k\rangle\right|^2\right\rangle\left|\left|\langle\phi^-|\phi_k\rangle\right|^2\right\rangle|0\rangle \to \left|\left|\langle\phi^+|\phi_k\rangle\right|^2\right\rangle\left|\left|\langle\phi^-|\phi_k\rangle\right|^2\right\rangle|y_k\rangle\,. \tag{22}$$

Step 11 Uncompute the ancillas using the inverse algorithm of *Step 2–Step 9*:

$$|k\rangle\left|\Psi_k^{\mathrm{ancilla}}\right\rangle|y_k\rangle \to |k\rangle\,|0\rangle\,|y_k\rangle\,. \tag{23}$$

4 Complexity analysis

Theorem 1 (QFTC) *Given an input* $\sum_k u_k |k\rangle$, *the required quantum state* $\sum_k u_k |k\rangle |y_k\rangle$ *can be prepared to digit accuracy* ϵ[1] *with fidelity* $1-\delta$[2] *using* $\mathcal{O}\big((\log N)^2/(\delta\epsilon)\big)$ *one- or two-qubit gates, and* $\mathcal{O}\big(1/(\delta\epsilon)\big)$ *calls of controlled-*O_x *and its inverse.*

Proof First, we consider the complexity involved in $\mathcal{A}_k^+$ (described in *Step 2–Step 6*). It contains Hadamard gates, controlled phase operators and swap gates which can be constructed using $\mathcal{O}\big((\log N)^2\big)$ one- or two-qubit gates and only one call of controlled-O_x.

The subsequent amplitude estimation block needs $\mathcal{O}(1/(\delta\varepsilon))$ applications of $Q_k^+ = -\mathcal{A}_k^+ S_0 (\mathcal{A}_k^+)^\dagger S_\chi$ to obtain accuracy ε with fidelity at least $1-\delta$ [7,24]. We then use the quantum multiply–adder and sine gate to obtain the value of $\left|\langle\phi^+|\phi_k\rangle\right|^2 = \frac{1}{4}(1+y_k^2)+y_k/2$ for different $|k\rangle$'s in the computational basis. Using a similar procedure to obtain $\left|\langle\phi^-|\phi_k\rangle\right|^2$, we obtain $y_k = \left|\langle\phi^+|\phi_k\rangle\right|^2 - \left|\langle\phi^-|\phi_k\rangle\right|^2$ finally. Since the derivative of $\sin x$ is always smaller than one, we set $\varepsilon = \Theta(\epsilon)$ in order to guarantee accuracy ϵ in y_k. As detailed in Appendix 1, the quantum multiply–adders and sine gates have complexity $\mathcal{O}(\text{polylog}(1/\varepsilon))$ which is smaller than $\mathcal{O}(1/\epsilon)$ in amplitude estimation. Therefore, the complexity of these gates can be omitted.

The total complexity of the proposed circuit will be $\mathcal{O}\big((\log N)^2/(\delta\epsilon)\big)$ one- or two-qubit gates, and $\mathcal{O}\big(1/(\delta\epsilon)\big)$ calls of controlled-O_x and its inverse. □

Throughout the proposed QFTC algorithm, $|k\rangle$ in Reg A is used to control the application of quantum operators acting on other registers, giving us the advantage of parallel calculating y_k for all k. Though values of y_k's cannot be obtained by a single measurement of $\sum_k |k\rangle |y_k\rangle$, they can be used in subsequent quantum computation once they are encoded in the computational basis.

The disadvantage of the QFTC algorithm to provide the value of $|y_k\rangle$ (as discussed in Sect. 5) compared to the corresponding classical algorithm lies in its accuracy. In the FFT, $\|\tilde{\boldsymbol{y}} - \boldsymbol{y}\| < \Theta(\log N) \times \epsilon$ [25,26]; in the QFTC, however, $\|\tilde{\boldsymbol{y}} - \boldsymbol{y}\| < \Theta(\sqrt{N}) \times \epsilon$. Precision at this level would be sufficient for example in Fourier transform spectroscopy when only a small set of frequencies dominate the behaviour of the vectors [27]. However, when high precision is needed, in order to achieve similar precision $\|\tilde{\boldsymbol{y}} - \boldsymbol{y}\| < \epsilon$ like the FFT, we will need $\sqrt{N}$ times the complexity in Theorem 1. Then we only have a quadratic, not exponential, speed-up in this case compared to the classical algorithm.

5 Application

One important family of operators is the circulant matrices which have found important applications in, for example, quantum walks on Moöbius strips [28], investigation on

[1] $|y_k - \tilde{y}_k| < \epsilon$, where $\tilde{y}_k$ is the truncated value of y_k with accuracy $\epsilon = 2^{-p_0}$.

[2] $\left|\left\langle \Psi^{\text{final}} \right| \left(\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle |\tilde{y}_k\rangle \right)\right| \geq 1-\delta$, where $\left|\Psi^{\text{final}}\right\rangle$ is the state obtained through the QFTC algorithm.

quantum supremacy [15], biochemical modelling [29], vibration analysis [30] and parallel diagnostic algorithm for super-computing [31].

Circulant matrices are defined as follows [32]:

$$C = \begin{pmatrix} c_0 & c_1 & \cdots & c_{N-1} \\ c_{N-1} & c_0 & \cdots & c_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ c_1 & c_2 & \cdots & c_0 \end{pmatrix}, \tag{24}$$

using an N-dimensional vector $\boldsymbol{c} = (c_0\, c_1\, \cdots\, c_{N-1})$. Such matrices are diagonalizable by the discrete Fourier transform (DFT), i.e.

$$C = F \Lambda F^{\dagger}, \tag{25}$$

where F is the Fourier matrix with $F_{kj} = \mathrm{e}^{2\pi i jk/N}/\sqrt{N}$, and Λ is a diagonal matrix of eigenvalues given by $\Lambda_k = \sqrt{N}\big(F(c_0\, c_1\, \cdots\, c_{N-1})^{\dagger}\big)_k \equiv \sqrt{N} F_k$. Note that the condition that C is Hermitian (in order to be a Hamiltonian) is equivalent to our assumption in Sect. 2 that the Fourier coefficients F_k are real. Since the eigenvalues of a circulant matrix are Fourier transform of its parameters, we are able to implement circulant quantum operators (non-unitary in general) using the conventional QFT through the manipulation of amplitudes, as detailed in Appendix 2.

This approach cannot be used directly for simulation of (non-sparse) circulant Hamiltonians, where we need to implement e^{-iCt} instead of C. Simulation of circulant Hamiltonians is equivalent to the implementation of continuous-time quantum walks on a weighted circulant graph [33]. Circulant matrices are adjacency matrices of circulant graphs, and c_j characterizes the probability for the walker to transfer from vertex ℓ to vertex $\ell - j$.

In order to simulate e^{-iCt}, we decompose it into $F\mathrm{e}^{-i\Lambda t}F^{\dagger}$, where $\mathrm{e}^{-i\Lambda t}$ can be simulated with the aid of the quantum circuit given in simulating diagonal Hamiltonians [34]. If the Fourier coefficients Λ_k are encoded in the computational basis, as performed by the QFTC algorithm, they can then be used to control the phase factor $\mathrm{e}^{-i\Lambda_k t}$ added to different eigenstates of the circulant matrix, for the purpose of implementing the diagonal Hamiltonian $\mathrm{e}^{-i\Lambda t}$.

In the following, we will demonstrate how the QFTC algorithm can be used to simulate Hamiltonians with a circulant matrix structure, as shown in Fig. 4:

Step 1 Perform the inverse QFT on $|s\rangle$:

$$|s\rangle = \sum_{k=0}^{N-1} s_k\, |k\rangle \rightarrow \sum_{k=0}^{N-1} \mathfrak{s}_k\, |k\rangle\,. \tag{26}$$

Step 2 Apply the QFTC algorithm (*Step 2–Step 11* in Sect. 3) for $\boldsymbol{c}$:

$$\sum_{k=0}^{N-1} \mathfrak{s}_k\, |k\rangle \rightarrow \sum_{k=0}^{N-1} \mathfrak{s}_k\, |k\rangle\, |F_k\rangle\,. \tag{27}$$

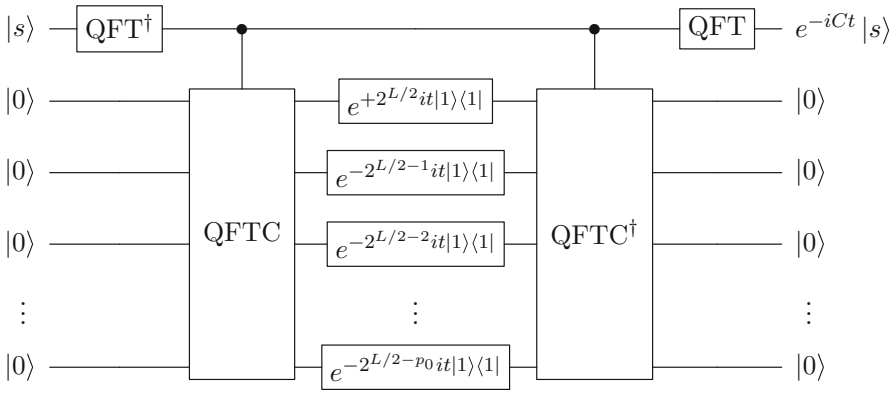

Fig. 4 Simulation of circulant Hamiltonians. $p_0 + 1$ is the number of digits of the resulting Fourier coefficients, and F_k was encoded in the form $f_0.f_1 f_2 \cdots f_{p_0}$ as the complemental code for a number between -1 and 1. Here we define QFTC $|k\rangle\,|0\rangle = |k\rangle\,|F_k\rangle$ (detailed in *Step 2–Step 11* in Sect. 3)

Step 3 Do controlled phase gate $\mathrm{e}^{+2^{L/2}it|1\rangle\langle 1|}$ on the first digit (qubit) of $|F_k\rangle$ and $\mathrm{e}^{-2^{L/2-p+1}it|1\rangle\langle 1|}$ on the pth digit (qubit) of $|F_k\rangle$ for all $p > 1$:

$$\sum_{k=0}^{N-1} \mathfrak{s}_k\,|k\rangle\,|F_k\rangle \to \sum_{k=0}^{N-1} \mathfrak{s}_k \mathrm{e}^{-i\Lambda_k t}\,|k\rangle\,|F_k\rangle\,. \tag{28}$$

Step 4 Undo the QFTC for every $|k\rangle$:

$$\sum_{k=0}^{N-1} \mathfrak{s}_k \mathrm{e}^{-i\Lambda_k t}\,|k\rangle\,|F_k\rangle \to \sum_{k=0}^{N-1} \mathfrak{s}_k \mathrm{e}^{-i\Lambda_k t}\,|k\rangle\,. \tag{29}$$

Step 5 Perform the QFT:

$$\sum_{k=0}^{N-1} \mathfrak{s}_k \mathrm{e}^{-i\Lambda_k t}\,|k\rangle \to \mathrm{e}^{-iCt}\,|s\rangle\,. \tag{30}$$

Theorem 2 (Simulation of Circulant Hamiltonians) *The simulation of a circulant Hamiltonian e^{-iCt} can be performed within error δ using $\mathcal{O}\big(\sqrt{N}t\,(\log N)^2/\delta^{3/2}\big)$ one- or two-qubit gates, as well as $\mathcal{O}\big(\sqrt{N}t/\delta^{3/2}\big)$ calls of controlled-O_x and its inverse, where $\boldsymbol{x} = \boldsymbol{c}$ is a unit vector in $\mathbb{C}^N$ and C is Hermitian.*[3]

Proof The error present in the Hamiltonian simulation is fully determined by the precision of the QFTC algorithm. According to the above QFTC complexity analysis, we need $\mathcal{O}\big((\log N)^2/(\delta\varepsilon)\big)$ one- or two-qubit gates, as well as $\mathcal{O}\big(1/(\delta\varepsilon)\big)$ calls of

[3] $\|\mathrm{e}^{-iCt} - \widetilde{\mathrm{e}^{-iCt}}\| \le \delta$, where $\widetilde{\mathrm{e}^{-iCt}}$ represents the operator that is actually performed by this algorithm.

controlled-O_x and its inverse, to achieve accuracy ε in F_k. The fidelity achieved for the Hamiltonian simulation, as defined by the squared modulus of inner product, is

$$(1-\delta)^2 \left|\langle e^{-i\tilde{C}t}\,|s\rangle\,, e^{-iCt}\,|s\rangle\rangle\right| = (1-\delta)^2 \left|\sum_{k=0}^{N-1} e^{i(\tilde{\Lambda}_k-\Lambda_k)t}\,|\mathfrak{s}_k|^2\right| > 1-\mathcal{O}((\sqrt{N}t\varepsilon)^2+\delta), \tag{31}$$

where the last inequality is derived using

$$\left|e^{i\gamma_1}+|\Gamma|\,e^{i\gamma_2}\right| = \left(1+|\Gamma|^2+2\,|\Gamma|\cos(\gamma_1-\gamma_2)\right)^{1/2} > (1+|\Gamma|)\left|\cos\frac{\gamma_1-\gamma_2}{2}\right|, \tag{32}$$

and $\tilde{\Lambda}_k$ are the estimated (truncated) eigenvalues calculated via the QFTC algorithm. For a fixed δ in the QFTC algorithm, if we choose $\varepsilon = \sqrt{\delta}/(\sqrt{N}t)$, the fidelity will be $1-\mathcal{O}(\delta)$. We then need $\mathcal{O}\big((\log N)^2/(\delta\varepsilon)\big) = \mathcal{O}\big(\sqrt{N}t\,(\log N)^2/\delta^{3/2}\big)$ one- or two-qubit gates, as well as $\mathcal{O}\big(\sqrt{N}t/\delta^{3/2}\big)$ calls of controlled-O_x and its inverse. □

The complexity in simulation of circulant Hamiltonians would depend linearly on the value of $\sqrt{|c_0|^2+\cdots+|c_{N-1}|^2}$ which was assumed to be 1. This value is always smaller (and normally much smaller) than the spectral norm of the circulant matrix C, which is often used to characterize the complexity in the simulation of dense Hamiltonians [35].

6 Conclusion

In this paper, we proposed a new QFTC algorithm, an efficient quantum scheme to encode the results of the discrete Fourier transform in the computational basis. This algorithm allows us to overcome a main shortcoming of the conventional quantum Fourier transform—the inability to perform operations controlled by the Fourier coefficients. In short, the QFTC utilizes swap tests to obtain a function of the Fourier coefficients in the amplitudes, with individual coefficients then extracted via amplitude estimation and quantum arithmetic.

Secondly, a detailed complexity analysis of the QFTC algorithm was performed, finding it requires $\mathcal{O}\big((\log N)^2/(\delta\epsilon)\big)$ calls of one- or two-qubit gates, as well as $\mathcal{O}\big(1/(\delta\epsilon)\big)$ calls of controlled-O_x and its inverse, in order to achieve fidelity $1-\delta$ and precision ϵ. Note that the overall complexity depends polylogarithmically on N, similarly to the conventional QFT, and we require only controlled phase gates and Hadamard gates. The inverse proportionality with the desired accuracy, ϵ, occurs due to the application of amplitude estimation within the algorithm.

Finally, we detailed an application of the QFTC algorithm in the simulation of circulant Hamiltonians, which requires $\mathcal{O}\big(\sqrt{N}t\,(\log N)^2/\delta^{3/2}\big)$ one- or two-qubit gates, as well as $\mathcal{O}\big(\sqrt{N}t/\delta^{3/2}\big)$ calls of controlled-O_x and its inverse to achieve fidelity $1-\delta$. This paves the way for a quantum circuit implementation of continuous-time quantum walks on circulant graphs, with potential applications in a wide array of disciplines. Further applications of the QFTC algorithm are expected.

Acknowledgements The authors would like to thank Ashley Montanaro for constructive comments and Jeremy O'Brien, Jonathan Matthews, Xiaogang Qiang, Lyle Noakes, Chuheng Zhang and Hanwen Zha for helpful discussions.

Appendix 1: Quantum arithmetic

Addition and multiplication are basic elements of arithmetic in classical computer. There have been several proposals on how to build quantum adders and multipliers [36–39], constructed predominately using CNOT gates and Toffoli gates. Draper's addition quantum circuits, however, utilize the quantum Fourier transformation (QFT) [40]. QFT-based multiplication and related quantum arithmetic have also been proposed [41–44]. In this appendix, for completeness, we outline the construction of the quantum arithmetic gates required for the QFTC algorithm in detail.

We show here, using QFT-based circuits and fixed-point number representation, all elementary quantum arithmetic gates used to construct the QFTC circuit (including adders, multipliers and cosine gates) have $\mathcal{O}(\text{poly}(n))$ complexity, where n is the number of qubits (number of digits) representing the number. With accuracy ϵ, this results in $\mathcal{O}(\text{polylog}(1/\epsilon))$ complexity.

QFT multiply–adder

We begin by describing a quantum multiply–adder for real inputs a and b between 0 and 1. Let $|a\rangle = |a_1\rangle\,|a_2\rangle\cdots|a_m\rangle$ represent the fixed-point number $a = 0.a_1a_2\cdots a_m$ (same for b). Using this representation, the quantum multiply–adder (QMA), as shown in Fig. 5a, can realize the following transformation,

$$\Pi^{\pm}_{m,n}\,|a\rangle\,|b\rangle\,|c\rangle = |a\rangle\,|b\rangle\,|c \pm a\times b\rangle\,, \tag{33}$$

where m and n denote the number of digits of a and b, respectively.

In quantum multiply–adders, the outputs, unlike the inputs, can be negative and we use the complemental code $c^{(C)} = c_0.c_1c_2\cdots c_{m+n} \in [0, 2)$ to represent the output $c \in (-1, 1)$ and $c = c^{(C)}$ if c is non-negative and $c = c^{(C)} - 2$ if c is negative. $|c\rangle$ is composed of $|c_0\rangle\,|c_1\rangle\cdots|c_{m+n}\rangle$. Note that this quantum multiply–adder also applies to any fixed-point-represented numbers by cleverly choosing the appropriate positions of the fractional points.

The quantum multiply–adder can be decomposed into the following form, as shown in Fig. 5b:

$$\Pi^{\pm}_{m,n} = (\mathbb{I}\otimes\mathbb{I}\otimes\text{QFT}^{\dagger})\times\pi^{\pm}_{m,n}\times(\mathbb{I}\otimes\mathbb{I}\otimes\text{QFT}), \tag{34}$$

where $\pi^{\pm}_{m,n}$ represents an intermediate quantum multiply–adder,

$$\pi^{\pm}_{m,n}\,|a\rangle\,|b\rangle\,|\phi(c)\rangle = |a\rangle\,|b\rangle\,|\phi(c \pm a\times b)\rangle \tag{35}$$

with $|\phi(c)\rangle := \text{QFT}\,|c\rangle$ and $|\phi_k(c)\rangle = \frac{1}{\sqrt{2}}(|0\rangle + \mathrm{e}^{2\pi i c\times 2^{m+n-k}}\,|1\rangle), k = 1, 2, \cdots, m+n+1$.

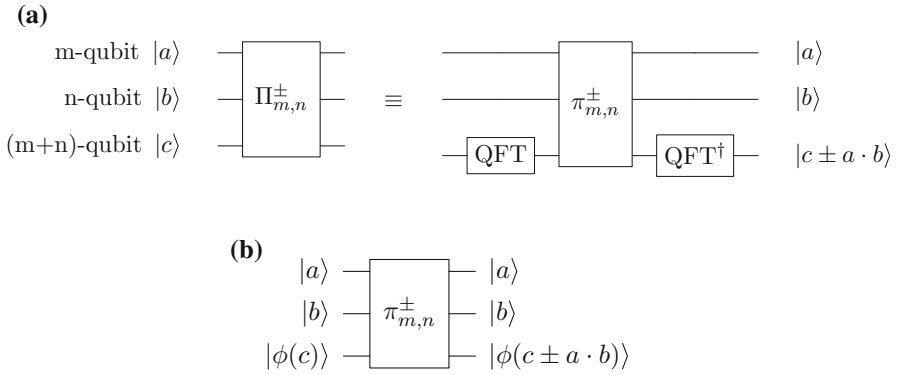

Fig. 5 Quantum circuit of the multiply–adder, **a** quantum multiply–adder, **b** intermediate multiply–adder

Figure 6 shows a detailed quantum circuit construction of $\pi^{\pm}_{m,n}$, using the QFT adders $2^{-l}\Sigma^{\pm}_{m,n}$, which act as follows:

$$\boxed{2^{-l}\Sigma^{\pm}_{m,n}}\,|b\rangle\,|\phi(c)\rangle = |b\rangle\left|\phi(c \pm 2^{-l}b)\right\rangle. \tag{36}$$

The QFT adders are constructed via controlled phase operations, as shown in Fig. 6c.

After applying the QFT adder $2^{-m}\Sigma^{\pm}_{m,n}$ (controlled by $|a_m\rangle$) in Fig. 6, we obtain

$$|\phi(c)\rangle \longrightarrow \left|\phi(c \pm a_m 2^{-m}b)\right\rangle. \tag{37}$$

Proceeding in a similar fashion, it can be seen that the final output state of the intermediate multiply–adder is

$$\left|\phi(c + a_m 2^{-m}b + \cdots + a_1 2^{-1}b)\right\rangle = |\phi(c \pm a \times b)\rangle\,. \tag{38}$$

To illustrate how the circuit works, take for example the evolution of $\phi_{m+n-l}(c)$ after $R^{\pm}_1, \ldots, R^{\pm}_n$:

$$|0\rangle + \mathrm{e}^{2\pi i c \times 2^l}\,|1\rangle \longrightarrow |0\rangle + \mathrm{e}^{2\pi i c \times 2^l \pm b}\,|1\rangle\,. \tag{39}$$

We then have

$$|\phi_k(c)\rangle \rightarrow \left|\phi_k(c \pm 2^{-l}b)\right\rangle.$$

It is clear from Fig. 6c that the QFT adder uses $\mathcal{O}\big((m+n)n\big)$ one- or two-qubit gates. Hence, the total complexity of the intermediate QFT multiply–adders is $\mathcal{O}\big((m+n)mn\big)$. Thus, with QFT scaling $\mathcal{O}\big((m+n)^2\big)$, the total complexity of the quantum multiply–adder $\Pi^{\pm}_{m,n}$ is $\max\{\mathcal{O}(mn^2), \mathcal{O}(nm^2)\}$.

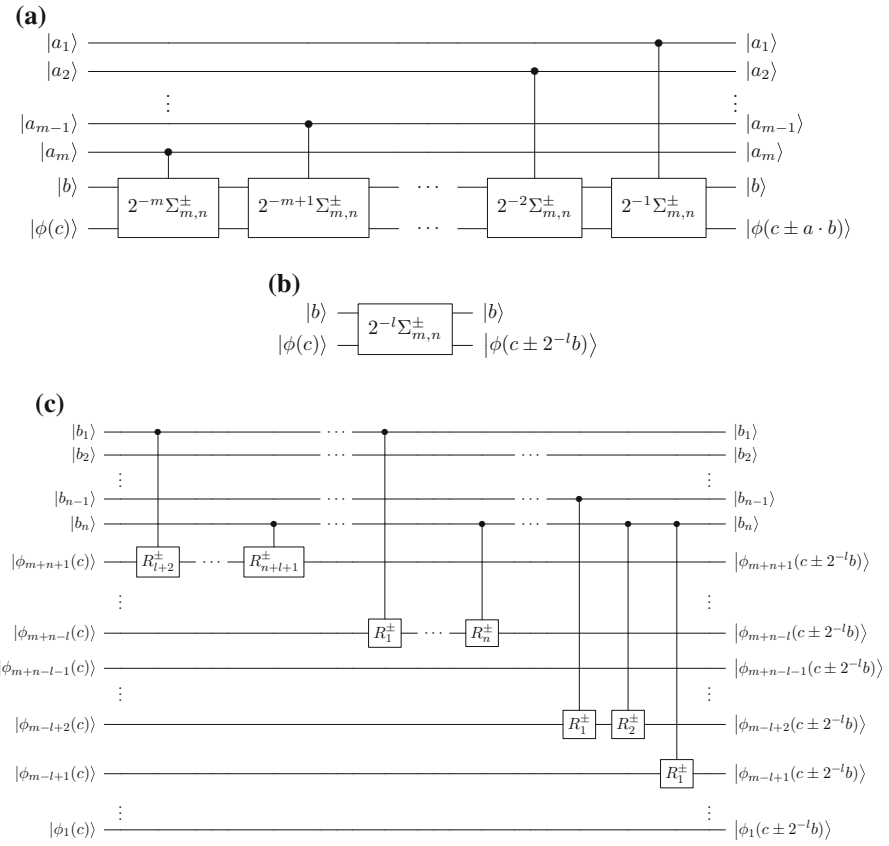

Fig. 6 Quantum circuit of $\pi^{\pm}_{m,n}$, (**a**), $\pi^{\pm}_{m,n}$ gate (**b**), QFT adder, (**c**) detailed quantum circuit construction of the QFT adder $2^{-l}\Sigma^{\pm}_{m,n}$, $R^{\pm}_k = |0\rangle\langle 0| + \mathrm{e}^{\pm 2\pi i/2^k}|1\rangle\langle 1|$

Note that if we choose $l = 0$ in $2^{-l}\Sigma^{\pm}_{m,n}$ and perform a QFT and an inverse QFT before and after the application of the QFT adder in Eq. 36, we have a quantum adder

$$|b\rangle|c\rangle \rightarrow |b\rangle|c \pm b\rangle\,. \tag{40}$$

We can also add (or subtract) two numbers without having to destroy their original values encoded in the computational basis, i.e.

$$|b\rangle|c\rangle|0\rangle \rightarrow |b\rangle|c\rangle|b\rangle \rightarrow |b\rangle|c\rangle|b \pm c\rangle \tag{41}$$

by using Eq. 40 twice.

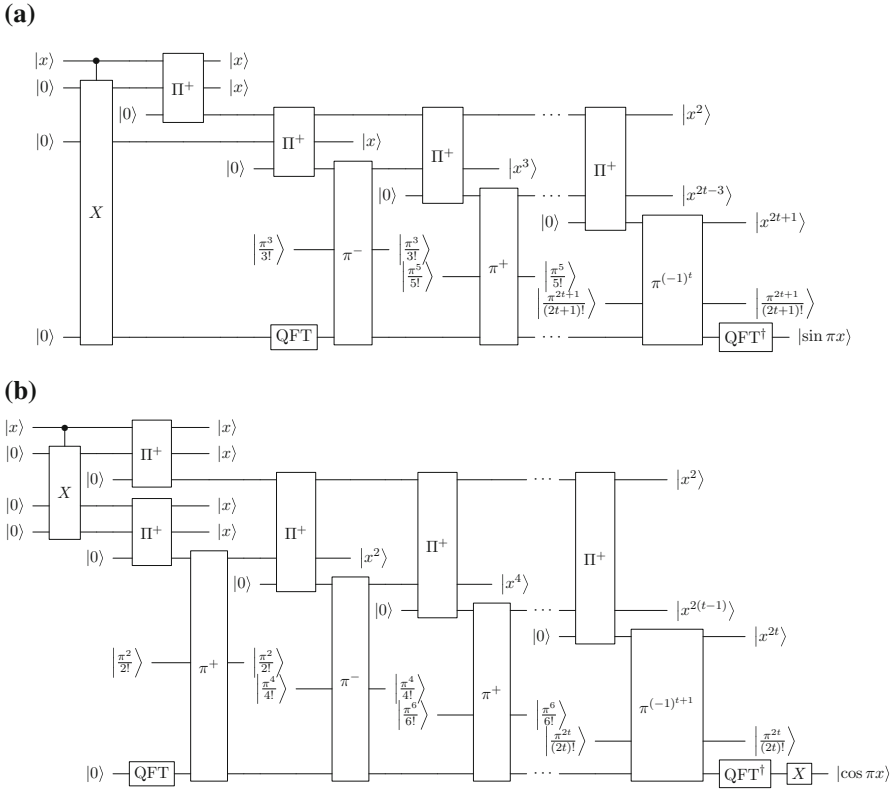

Fig. 7 Quantum circuits of the sine and cosine gates ($|0\rangle$ represents a number of qubits in above circuits where the numbers are omitted). Pauli-X gates are used to transform $|0\rangle$ into $|x\rangle$, and the subscript for all the quantum multiply–adders in above circuits is (p', p'), **a** sine gate, **b** cosine gate

Quantum sine and cosine gate

By implementing the Taylor series using the quantum multiply–adder, we are able to build a quantum sine (and cosine) gate. Suppose $x = 0.x_1x_2\cdots x_n$ and $x \in [0, 1)$. We aim to build a sine gate calculating the value of $\sin \pi x$, performing $|x\rangle\,|0^n\rangle\,|0^m\rangle \rightarrow |x\rangle\,|\sin \pi x\rangle\,\left|\Psi^{\text{ancilla}}\right\rangle$.

We now consider the error in the truncated Taylor series. First, the error introduced by imprecision in the n-digit representation of x is $\mathcal{O}(2^{-n})$, since the derivative of $\sin \pi x$ is bounded. The Taylor series of $\sin \pi x$ at around $x = 0$ is

$$\sin \pi x = \pi x - \frac{(\pi x)^3}{3!} + \frac{(\pi x)^5}{5!} - \cdots + (-1)^t \frac{(\pi x)^{2t+1}}{(2t+1)!} + \frac{(-1)^{t+1}\cos \pi z}{(2t+3)!}(\pi x)^{(2t+3)}. \tag{42}$$

The remainder term for the kth term in the expansion is $\frac{f^{(k+1)}(z)}{(k+1)!}x^{k+1}$, where $z \in (0, x)$, according to Taylor's Theorem [45]. As a result, in Eq. (42), the reminder term (error) is $\frac{(-1)^{t+1}\cos \pi z}{(2t+3)!}(\pi x)^{(2t+3)}$ and is obviously bounded by $\mathcal{O}(2^{-n})$ for $t = n$.

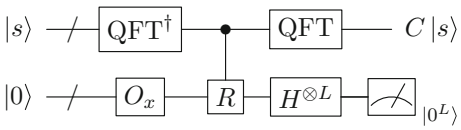

Fig. 8 Implementation of circulant matrices. Here $R\,|k\rangle\,|j\rangle = \mathrm{e}^{2\pi ikj/N}\,|k\rangle\,|j\rangle$

In the sine gate, the $t+1$ terms $\left\{\pi x, \frac{(\pi x)^3}{3!}, \cdots, (-1)^t \frac{(\pi x)^{2t+1}}{(2t+1)!}\right\}$ are first calculated and then added (or subtracted) together. Suppose each of the $t+1$ terms has an error within 2^{-p}. Taking $p = n + \lceil \log n \rceil = \mathcal{O}(n)$, the error introduced by adding and subtracting will be $\mathcal{O}(t \times 2^{-p}) = \mathcal{O}(2^{-n})$. Suppose all multiply–adders have p' digits inputs. When errors in y_1, y_2 are within $2^{-(\ell+1)}$ and $y_1, y_2 \le 1 - 2^{-(\ell+1)}$, $(y_1 + 2^{-(\ell+1)})(y_2 + 2^{-(\ell+1)}) = y_1 y_2 + 2^{-\ell}(y_1 + y_2)/2 + 2^{-2\ell-2} \le y_1 y_2 + 2^{-\ell}$. It means that by applying the multiply–adders $2t$ times, the error will be 2^{2t} times larger. Thus, we can choose a $p' = \mathcal{O}(p + 2t) = \mathcal{O}(n)$ which guarantees accuracy 2^{-p} in all the powers of x and also all the $t+1$ terms in the Taylor series.

We conclude that we can choose $t = \mathcal{O}(n)$ and $p' = \mathcal{O}(n)$ so that the total accuracy of the sine gate is bounded by 2^{-n}. Figure 7 shows the quantum circuit for the sine and cosine gate. The complexity of the quantum sine gate can be calculated based on the scaling of quantum multiply–adders which equals to $\mathcal{O}(p'^3)$. The total complexity of the quantum sine gate is $\mathcal{O}(tp'^3) = \mathcal{O}(n^4)$ for accuracy 2^{-n}. To put it in another way, $\mathcal{O}(\mathrm{polylog}(1/\epsilon))$ one- or two-qubit gates are required to achieve accuracy ϵ.

Appendix 2: Implementing circulant operators

Consider an arbitrary state $|s\rangle$. We wish to obtain $C\,|s\rangle$, where C is an arbitrary circulant matrix. Below, we present a possible algorithm for implementing a circulant matrix quantum operator (see Fig. 8).

Step 1 Perform the inverse QFT on $|s\rangle$:

$$\sum_{k=0}^{N-1} s_k\,|k\rangle \to \sum_{k=0}^{N-1} \mathfrak{s}_k\,|k\rangle\,. \tag{43}$$

Step 2 Add another register prepared to $\sum_{j=0}^{N-1} c_j\,|j\rangle$ using O_x ($\boldsymbol{x} = \boldsymbol{c}$ in Eq. 6):

$$\sum_{k=0}^{N-1} \mathfrak{s}_k\,|k\rangle \to \sum_{j,k=0}^{N-1} \mathfrak{s}_k c_j\,|k\rangle\,|j\rangle\,. \tag{44}$$

Step 3 Apply the controlled phase gate so that $|k\rangle\,|j\rangle \to \mathrm{e}^{2\pi ikj/N}\,|k\rangle\,|j\rangle$:

$$\sum_{j,k=0}^{N-1} \mathfrak{s}_k c_j\,|k\rangle\,|j\rangle \to \sum_{j,k=0}^{N-1} \mathfrak{s}_k c_j \mathrm{e}^{2\pi ijk/N}\,|k\rangle\,|j\rangle\,. \tag{45}$$

Step 4 Apply Hadamard gates to $|j\rangle$:

$$\sum_{j,k=0}^{N-1} \mathfrak{s}_k c_j e^{2\pi i jk/N} |k\rangle |j\rangle \rightarrow \sum_{j,k=0}^{N-1} \mathfrak{s}_k |k\rangle \left(F_k \left|0^L\right\rangle + \sqrt{1-F_k^2} \left|0^\perp\right\rangle \right), \quad (46)$$

where $\left|0^\perp\right\rangle$ represents any states perpendicular to $\left|0^L\right\rangle$.

Step 5 By post-selecting the ancillary qubit state $\left|0^L\right\rangle$, the quantum state in the first register collapses to

$$\frac{1}{\sqrt{\sum_k |F_k \mathfrak{s}_k|^2}} \sum_{k=0}^{N-1} F_k \mathfrak{s}_k |k\rangle . \quad (47)$$

Step 6 Perform the QFT:

$$\mathrm{QFT} \sum_{k=0}^{N-1} \mathfrak{s}_k F_k |k\rangle \propto C |s\rangle . \quad (48)$$

Note that the post-selection probability of obtaining the correct state in *Step 5* is

$$p = \sum_{k=0}^{N-1} |\mathfrak{s}_k F_k|^2 , \quad (49)$$

and p equals to $1/N$ when C is unitary. Therefore, using amplitude amplification [7], $\mathcal{O}((\log N)^2/\sqrt{p})$ one- or two-qubit gates, as well as $\mathcal{O}(1/\sqrt{p})$ calls of O_x, O_s and their inverses, are needed to implement a circulant matrix operation C, where $O_s \left|0^L\right\rangle = \sum_{k=0}^{N-1} s_k |k\rangle$.

References

1. Shor, P.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. **26**, 1484 (1997)
2. Deutsch, D.: Quantum theory, the Church-Turing principle and the universal quantum computer. Proc. R. Soc. Lond. A Math. Phys. Eng. Sci. **400**, 97 (1985). doi:10.1098/rspa.1985.0070
3. Bergland, G.: A guided tour of the fast Fourier transform. IEEE Spectr. **6**, 41 (1969)
4. Cleve, R., Watrous, J.: Fast parallel circuits for the quantum Fourier transform. In: 41st Annual Symposium on Foundations of Computer Science Proceedings, pp. 526–536 (2000)
5. Kitaev, A.Y.: Quantum measurements and the Abelian stabilizer problem. arXiv:quant-ph/9511026 (1995)
6. Brassard, G., Høyer, P., Tapp, A.: Quantum counting. In: Larsen, K.G., Skyum, S., Winskel, G. (eds.) Automata, languages and programming No. 1443 in Lecture Notes in Computer Science, pp. 820–831. Springer, Berlin (1998)
7. Brassard, G., Hoyer, P., Mosca, M., Tapp, A.: Quantum amplitude amplification and estimation. Contemp. Math. **305**, 53–74 (2002)
8. Benenti, G., Strini, G.: Quantum simulation of the single-particle Schrödinger equation. Am. J. Phys. **76**, 657 (2008)
9. Kassal, I., Jordan, S.P., Love, P.J., Mohseni, M., Aspuru-Guzik, A.: Polynomial-time quantum algorithm for the simulation of chemical dynamics. Proc. Natl. Acad. Sci. **105**(48), 18681 (2008)

10. Szkopek, T., Roychowdhury, V., Yablonovitch, E., Abrams, D.S.: Eigenvalue estimation of differential operators with a quantum algorithm. Phys. Rev. A **72**, 062318 (2005)
11. Hales, L., Hallgren, S.: An improved quantum Fourier transform algorithm and applications. In: 41st Annual Symposium on Foundations of Computer Science Proceedings, pp. 515–525 (2000)
12. Schützhold, R.: Pattern recognition on a quantum computer. Phys. Rev. A **67**, 062311 (2003)
13. van Dam, W., Hallgren, S., Ip, L.: Quantum algorithms for some hidden shift problems. SIAM J. Comput. **36**, 763 (2006)
14. Jordan, S.P.: Fast quantum algorithm for numerical gradient estimation. Phys. Rev. Lett. **95**, 050501 (2005)
15. Qiang, X., Loke, T., Montanaro, A., Aungskunsiri, K., Zhou, X., O'Brien, J.L., Wang, J.B., Matthews, J.C.F.: Efficient quantum walk on a quantum processor. arXiv:1510.08657 [quant-ph] (2015)
16. Harrow, A.W., Hassidim, A., Lloyd, S.: Quantum algorithm for linear systems of equations. Phys. Rev. Lett. **103**, 150502 (2009)
17. Lloyd, S., Mohseni, M., Rebentrost, P.: Quantum algorithms for supervised and unsupervised machine learning. arXiv:1307.0411 (2013)
18. Rebentrost, P., Mohseni, M., Lloyd, S.: Quantum support vector machine for big data classification. Phys. Rev. Lett. **113**(13), 130503 (2014)
19. Giovannetti, V., Lloyd, S., Maccone, L.: Architectures for a quantum random access memory. Phys. Rev. A **78**(5), 052310 (2008)
20. Grover, L., Rudolph, T.: Creating superpositions that correspond to efficiently integrable probability distributions. arXiv:quant-ph/0208112 (2002)
21. Kaye, P., Mosca, M.: Quantum networks for generating arbitrary quantum states. arXiv:quant-ph/0407102 (2004)
22. Soklakov, A.N., Schack, R.: Efficient state preparation for a register of quantum bits. Phys. Rev. A **73**, 012307 (2006)
23. Buhrman, H., Cleve, R., Watrous, J., de Wolf, R.: Quantum fingerprinting. Phys. Rev. Lett. **87**, 167902 (2001)
24. Nielsen, M.A., Chuang, I.L.: Quantum computation and quantum information. Cambridge University Press, Cambridge (2010)
25. Ramos, G.U.: Roundoff error analysis of the fast Fourier transform. Math. Comput. **25**(116), 757–768 (1971)
26. Tasche, M., Zeuner, H.: Worst and average case roundoff error analysis for FFT. BIT Numer. Math. **41**(3), 563–581 (2001)
27. Bell, R.: Introductory Fourier transform spectroscopy. Elsevier, Amsterdam (2012)
28. Delanty, M., Steel, M.: Discretely-observable continuous time quantum walks on Moöbius strips and other exotic structures in 3D integrated photonics. Phys. Rev. A **86**, 043821 (2012)
29. Yoneda, T., Sung, Y.M., Lim, J.M., Kim, D., Osuka, A.: PdII complexes of [44]- and [46] Decaphyrins: the largest Hückel aromatic and antiaromatic, and Möbius aromatic macrocycles. Angew. Chem. Int. Ed. Engl. **53**, 13169 (2014)
30. Olson, B.J., Shaw, S.W., Shi, C., Pierre, C., Parker, R.G.: Circulant matrices and their application to vibration analysis. Appl. Mech. Rev. **66**, 040803 (2014)
31. Cheng, B., Fan, J., Jia, X., Jia, J.: Parallel construction of independent spanning trees and an application in diagnosis on mobius cubes. J. Supercomput. **65**, 1279 (2013)
32. Golub, G.H., Van Loan, C.F.: Matrix computations, vol. 3. JHU Press, Baltimore (2012)
33. Mülken, O., Blumen, A.: Continuous-time quantum walks: models for coherent transport on complex networks. Phys. Rep. **502**, 37 (2011)
34. Childs, A.M.: Quantum information processing in continuous time. Ph.D. thesis, Massachusetts Institute of Technology (2004)
35. Childs, A.M., Kothari, R.: Limitations on the simulation of non-sparse hamiltonians. arXiv preprint arXiv:0908.4398 (2009)
36. Cuccaro, S.A., Draper, T.G., Kutin, S.A., Moulton, D.P.: A new quantum ripple-carry addition circuit. arXiv:quant-ph/0410184 (2004)
37. Draper, T.G., Kutin, S.A., Rains, E.M., Svore, K.M.: A logarithmic-depth quantum carry-lookahead adder. Quantum Info. Comput. **6**, 351–369 (2006)
38. Álvarez-Sánchez, J.J., Álvarez-Bravo, J.V., Nieto, L.M.: A quantum architecture for multiplying signed integers. J. Phys. Conf. Ser. **128**, 012013 (2008)

39. Vedral, V., Barenco, A., Ekert, A.: Quantum networks for elementary arithmetic operations. Phys. Rev. A **54**, 147 (1996)
40. Draper, T.G.: Addition on a quantum computer. arXiv:quant-ph/0008033 (2000)
41. Ruiz-Perez, L., Garcia-Escartin, J.C.: Quantum arithmetic with the quantum Fourier transform. arXiv:1411.5949 [quant-ph] (2014)
42. Maynard, C.M., Pius, E.: A quantum multiply-accumulator. Quantum Inf. Process. **13**(5), 1127 (2013). doi:10.1007/s11128-013-0715-5
43. Maynard, C.M., Pius, E.: Integer arithmetic with hybrid quantum-classical circuits. arXiv:1304.4069 (2013)
44. Pavlidis, A., Gizopoulos, D.: Fast quantum modular exponentiation architecture for Shor's factoring algorithm. Quantum Info. Comput. **14**(7&8), 649–682 (2014)
45. Kline, M.: Calculus: an intuitive and physical approach. Courier Corporation, North Chelmsford (1998)