Author for correspondence:

J. B. Wang
e-mail: jingbo.wang@uwa.edu.au

Efficient quantum circuits for dense circulant and circulant-like operators

S. S. Zhou[1,2,3] and J. B. Wang[4]

[1]Kuang Yaming Honors School, Nanjing University, Nanjing 210093, People's Republic of China

[2]Department of Physics, and [3]Department of Applied Physics, Yale Quantum Institute, Yale University, New Haven, CT 06520, USA

[4]School of Physics, The University of Western Australia, Perth, Western Australia 6009, Australia

JBW, 0000-0001-7544-0084

Circulant matrices are an important family of operators, which have a wide range of applications in science and engineering-related fields. They are, in general, non-sparse and non-unitary. In this paper, we present efficient quantum circuits to implement circulant operators using fewer resources and with lower complexity than existing methods. Moreover, our quantum circuits can be readily extended to the implementation of Toeplitz, Hankel and block circulant matrices. Efficient quantum algorithms to implement the inverses and products of circulant operators are also provided, and an example application in solving the equation of motion for cyclic systems is discussed.

1. Introduction

Quantum computation exploits the intrinsic nature of quantum systems in a way that promises to solve problems otherwise intractable on conventional computers. As the most widely used model of quantum computation, a quantum circuit provides a complete description of a specified quantum algorithm, whose computational complexity is determined by the number of quantum gates required. In general, the number of two-level gates (i.e. unitary matrices acting non-trivially on two-dimensional subspaces, which are universal for computation) needed to decompose an arbitrary unitary in N dimensions scales as $O(N^2)$. There are many known N-dimensional matrices that cannot be decomposed as a product of fewer than $N - 1$ two-level gates [1], and thus cannot be implemented efficiently on a quantum computer. An essential research focus in quantum computation is to explore which kinds of linear operations (either unitary or non-unitary) can be efficiently implemented

using $O(\mathrm{poly}(\log N))$ number of elementary quantum gates (i.e. one- or two-level unitary matrices) and measurements.

Significant breakthroughs in the area include the development of efficient quantum algorithms for Hamiltonian simulation, which is central to the studies of chemical and biological processes [2–8]. Recently, Berry, Childs and Kothari presented an algorithm for sparse Hamiltonian simulation achieving near-linear scaling with the sparsity and sublogarithmic scaling with the inverse of the error [8]. Additionally, using the Hamiltonian simulation algorithm as an essential ingredient, Harrow *et al.* [9] showed that for a sparse and well-conditioned matrix A, there is an efficient algorithm (known as the HHL algorithm) that provides a quantum state proportional to the solution $\sum_j x_j |j\rangle$ of the linear system of equations $Ax = b$.

However, as proven by Childs & Kothari [10], it is impossible to perform a generic simulation of an arbitrary dense Hamiltonian H in $\mathbb{C}^{N \times N}$ in time $O(\mathrm{poly}(\|H\|, \log N))$, where $\|H\|$ is the spectral norm, but possible for certain non-trivial classes of Hamiltonians. It is then natural to ask under what conditions we can extend the sparse Hamiltonian simulation algorithm and the HHL algorithm to the realm of dense matrices. In this paper, we use the 'unitary decomposition' approach developed by Berry *et al.* [7] to implement dense circulant Hamiltonians in time $O(\mathrm{poly}(\|H\|, \log N))$. Combining this with the HHL algorithm, we can also efficiently implement the inverse of dense circulant matrices and thus solve systems of circulant matrix linear equations.

Furthermore, we provide an efficient algorithm to implement circulant matrices C directly, by decomposing them into a linear combination of unitary matrices. We then apply the same technique to implement block circulant matrices, Toeplitz and Hankel matrices, which have significant applications in physics, mathematics and engineering [11–21]. For example, we can simulate classical random walks on circulant, Toeplitz and Hankel graphs [22,23]. In fact, any arbitrary matrix can be decomposed into a product of Toeplitz matrices [24]. If the number of Toeplitz matrices required is in the order of $O(\mathrm{poly}(\log N))$, we can have an efficient quantum circuit.

This paper is organized as follows. In §2, we present an algorithm to implement circulant matrices, followed by discussions on block circulant matrices, Toeplitz and Hankel matrices in §3. In §§4 and 5, we provide efficient methods to simulate circulant Hamiltonians and to implement the inverse of circulant matrices. In §6, we describe a technique to efficiently implement products of circulant matrices. In the last section, we provide an example application in solving the equation of motion for vibrating systems with cyclic symmetry.

2. Implementation of circulant matrices

A circulant matrix has each row right-rotated by one element with respect to the previous row, defined as

$$C = \begin{pmatrix} c_0 & c_1 & \cdots & c_{N-1} \\ c_{N-1} & c_0 & \cdots & c_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ c_1 & c_2 & \cdots & c_0 \end{pmatrix}, \tag{2.1}$$

using an N-dimensional vector $c = (c_0\ c_1\ \cdots\ c_{N-1})$ [25]. In this paper, we will assume c_j to be non-negative for all j, which is often the case in practical applications. We also assume that the spectral norm (the largest eigenvalue) $\|C\| = \sum_{j=0}^{N-1} c_j$ of the circulant matrix C is equal to 1 for simplicity.

Note that C can be decomposed into a linear combination of efficiently realizable unitary matrices as follows:

$$C = \begin{pmatrix} c_0 & c_1 & \cdots & c_{N-1} \\ c_{N-1} & c_0 & \cdots & c_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ c_1 & c_2 & \cdots & c_0 \end{pmatrix} = c_0 \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} + c_1 \begin{pmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & \vdots \\ \vdots & \vdots & \ddots & 1 \\ 1 & 0 & \cdots & 0 \end{pmatrix} + \cdots = \sum_{j=0}^{N-1} c_j V_j, \tag{2.2}$$

where $V_j = \sum_{k=0}^{N-1} |(k-j) \bmod N\rangle\langle k|$. Such a linear combination of unitary matrices can be dealt with by the unitary decomposition approach introduced by Berry *et al.* [7]. For completeness, we restate their method as lemma 2.1 given below.
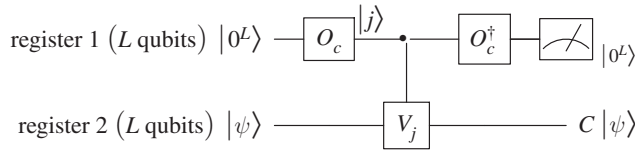
Figure 1. Quantum circuit to implement a circulant matrix.

Lemma 2.1. *Let $M = \sum_{j=0}^{J} \alpha_j W_j$ be a linear combination of unitaries W_j with $\alpha_j \geq 0$ for all j and $\sum_{j=0}^{J} \alpha_j = 1$. Let O_α be any operator that satisfies $O_\alpha |0^m\rangle = \sum_{j=0}^{J} \sqrt{\alpha_j}|j\rangle$, where m is the number of qubits used to represent $|j\rangle$, and $\mathrm{select}(W) = \sum_{j=0}^{J} |j\rangle\langle j| \otimes W_j$. Then*

$$(O_\alpha^\dagger \otimes I)\,\mathrm{select}(W)(O_\alpha \otimes I)|0^m\rangle|\psi\rangle = |0^m\rangle M|\psi\rangle + |\Psi^\perp\rangle, \tag{2.3}$$

where $(|0^m\rangle\langle 0^m| \otimes I)|\Psi^\perp\rangle = 0$.

Unless stated otherwise, we assume that $N = 2^L$, where L is an integer. If N is not a power of two, we will need to embed the system into a larger Hilbert space whose dimension is a power of two. On the other hand, it is also convenient to simply discretize practical problems using powers of two. Lemma 2.1 can be directly applied to implement the circulant matrix C, as shown in figure 1, by taking $M = C$, $\alpha_j = c_j$, $W_j = V_j$, $J = 2^L$ and $m = L$. Since $\mathrm{select}(V)|j\rangle|k\rangle = |j\rangle|(k - j) \mod N\rangle$, it can be implemented using quantum adders [26,27], which requires $O(\log N)$ one- or two-qubit gates. Note that when N is not a power of two, it may take additional $O(\log N)$ ancillary qubits to implement the 'mod N' operation in $\mathrm{select}(V)$, for example, by first subtracting N from $k - j$ and then using the sign qubit to control the 'mod N' operation.

A measurement result of $|0^L\rangle$ in the first register generates the required state $C|\psi\rangle$ in the second register. The probability of this measurement outcome is $O(\|C|\psi\rangle\|^2)$. With the help of amplitude amplification [28], this can be further improved, requiring only $O(1/\|C|\psi\rangle\|)$ rounds of application of $(O_c^\dagger \otimes I)\,\mathrm{select}(V)(O_c \otimes I)$. The amplitude amplification procedure also requires the same number of applications of O_ψ, where $O_\psi|0^L\rangle = |\psi\rangle$, and its inverse in order to reflect quantum states about the initial state $|0^L\rangle|\psi\rangle$. If O_ψ is unknown, amplitude amplification is not applicable and we will need to repeat the measuring process in figure 1 $O(1/\|C|\psi\rangle\|^2)$ times, during which $O(1/\|C|\psi\rangle\|^2)$ copies of $|\psi\rangle$ are required. It is worth noting that with the assumption $c_j \geq 0$, C is unitary if and only if $C = V_j$. In other words, a non-trivial circulant matrix is non-unitary and therefore, the oblivious amplitude amplification procedure [29] cannot be applied.

Provided with the oracle O_c satisfying $O_c|0^L\rangle = \sum_{j=0}^{N-1} \sqrt{c_j}|j\rangle$, theorem 2.2 follows directly from the above discussions. O_c can be efficiently implemented for certain efficiently computable vectors $\boldsymbol{c}$ [30–32]. Another way to construct states like $\sum_{j=0}^{N-1} \sqrt{c_j}|j\rangle$ is via qRAM, which uses $O(N)$ hardware resources but only $O(\log N)$ operations to access them [33,34].

Theorem 2.2 (Implementation of circulant matrices). *There exists an algorithm creating the quantum state $C|\psi\rangle$ for an arbitrary quantum state $|\psi\rangle = \sum_{k=0}^{N-1} \psi_k|k\rangle$, using $O(1/\|C|\psi\rangle\|)$ calls of O_c, O_ψ and their inverses, as well as $O(\log N/\|C|\psi\rangle\|)$ additional one- or two-qubit gates.*

The complexity in theorem 2.2 is inversely proportional to the square root of $p = \|C|\psi\rangle\|^2$, which depends on the quantum state to be acted upon. Specifically, $|C|\psi\rangle|^2 = \langle\psi\|C^\dagger C\|\psi\rangle = \langle\psi|F\Lambda^\dagger F^\dagger F\Lambda F^\dagger|\psi\rangle = \langle\psi|F\Lambda^\dagger \Lambda F^\dagger|\psi\rangle$. Here we use the diagonal form of C [25], $C = F\Lambda F^\dagger$, where F is the Fourier matrix with $F_{kj} = e^{2\pi ijk/N}/\sqrt{N}$ and Λ is a diagonal matrix of eigenvalues given by $\Lambda_k = \sum_{j=0}^{N-1} c_j\, e^{2\pi ijk/N}$. Since the spectral norm $\|C\|$ of the circulant matrix C is equal to one, we have $p = \langle\psi|F\Lambda^\dagger \Lambda F^\dagger|\psi\rangle \geq 1/\kappa^2$, where κ is the condition number, defined as the ratio between C's largest and smallest (absolute value of) eigenvalues [9]. Therefore, our algorithm is bound to perform well when $\kappa = O(\mathrm{poly}(\log N))$. In the ideal case where $\kappa = 1$ and $p = 1$, the vector $\boldsymbol{c}$ is a unit basis in which only one element is equal to one and the others are zero.

Even when κ is large, our algorithm is still efficient when the input quantum state after Fourier transform is in the subspace whose corresponding eigenvalues are large. For example, when $\Lambda_k = \cos(2\pi k/N)$ we have $\kappa = \infty$ when $N > 2$. $p = \langle\phi|\Lambda^\dagger \Lambda|\phi\rangle = \sum_{k=0}^{N-1} \cos(2\pi k/N)^2|\phi_k|^2 \geq \sum_{k\notin[N/8,3N/8]\cup[5N/8,7N/8]} \frac{1}{2}|\phi_k|^2$ in which $|\phi\rangle := F^\dagger|\psi\rangle = \sum_{k=0}^{N-1} \phi_k|k\rangle$. The success rate is therefore

lower-bounded by a constant as long as the input quantum state is restricted in a subspace such that $\phi_k = 0$ when $k \in [N/8, 3N/8] \cup [5N/8, 7N/8]$.

3. Circulant-like matrices

3.1. Block circulant matrices

Some block circulant matrices with special structures can also be implemented efficiently in a similar manner. We assume the blocks are N'-dimensional matrices and $L' = \log N'$ in the following discussions.

Firstly, when each block is a unitary operator up to a constant factor (i.e. $\mathbf{C}_j = c_j \mathbf{U}_j$), we have a unitary block (UB) matrix:

$$C_{UB} = \begin{pmatrix} \mathbf{C}_0 & \mathbf{C}_1 & \cdots & \mathbf{C}_{N-1} \\ \mathbf{C}_{N-1} & \mathbf{C}_0 & \cdots & \mathbf{C}_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_1 & \mathbf{C}_2 & \cdots & \mathbf{C}_0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \otimes \mathbf{C}_0 + \begin{pmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & \vdots \\ \vdots & \vdots & \ddots & 1 \\ 1 & 0 & \cdots & 0 \end{pmatrix} \otimes \mathbf{C}_1 + \cdots$$

$$= \sum_{j=0}^{N-1} V_j \otimes \mathbf{C}_j = \sum_{j=0}^{N-1} c_j V_j \otimes \mathbf{U}_j. \tag{3.1}$$

If the set of blocks $\{\mathbf{U}_j\}_{j=0}^{N-1}$ can be efficiently implemented, then by simply replacing select$(V) = \sum_{j=0}^{N-1} |j\rangle\langle j| \otimes V_j$ with $\sum_{j=0}^{N-1} |j\rangle\langle j|(V_j \otimes \mathbf{U}_j)$, we can efficiently implement the block circulant matrices C_{UB} using the same algorithm discussed in §2 as illustrated in figure 2a.

Specifically, when the set of blocks $\{\mathbf{U}_j\}_{j=0}^{N-1}$ are one-dimensional, we can implement complex-valued circulant matrices with efficiently computable phase. For example, for $\mathbf{U}_j = (e^{i\theta j})$, $j = 0, 1, \ldots, N-1$, circulant matrices with the parameter vector $\boldsymbol{c} = (c_0, e^{i\theta} c_1, \ldots, e^{i(N-1)\theta} c_{N-1})$ can be implemented efficiently. Moreover, if $\theta = \pi$, $\boldsymbol{c} = (c_0, -c_1, \ldots, (-1)^{N-1} c_{N-1})$ corresponding to the circulant matrix with negative elements on odd-numbered sites is efficiently implementable.

Another important family is block circulant matrices with circulant blocks (CB), which has found a wide range of applications in algorithms, mathematics, etc. [18–21]. It is defined as follows:

$$C_{CB} = \begin{pmatrix} \mathbf{C}_0 & \mathbf{C}_1 & \cdots & \mathbf{C}_{N-1} \\ \mathbf{C}_{N-1} & \mathbf{C}_0 & \cdots & \mathbf{C}_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_1 & \mathbf{C}_2 & \cdots & \mathbf{C}_0 \end{pmatrix}, \tag{3.2}$$

where $\mathbf{C}_j$ is a circulant matrix specified by a N'-dimensional vector $\boldsymbol{c}_j = (c_{j0} \ c_{j1} \ \cdots \ c_{j(N'-1)})$. C_{CB} is a $N \times N'$-dimensional matrix determined by $N \times N'$ parameters $\{c_{jj'}\}_{\substack{j=0,\ldots,N-1 \\ j'=0,\ldots,N'-1}}$. It can be decomposed as follows:

$$C_{CB} = \sum_{j=0}^{N-1} \sum_{j'=0}^{N'-1} c_{jj'} V_j \otimes V_{j'}. \tag{3.3}$$

Given an oracle $O_{c'}$ satisfying $O_{c'}|0^{L+L'}\rangle = \sum_{j=0}^{N-1} \sum_{j=0}^{N'-1} c_{jj'} |j\rangle |j'\rangle$, we can implement C_{CB} using the quantum circuit shown in figure 2b, which adopts a combination of two quantum subtractors.

3.2. Toeplitz and Hankel matrices

A Toeplitz matrix is a matrix in which each descending diagonal from left to right is constant, which can be written explicitly as

$$T = \begin{pmatrix} t_0 & t_{-1} & \cdots & t_{-(N-2)} & t_{-(N-1)} \\ t_1 & t_0 & \cdots & t_{-(N-3)} & t_{-(N-2)} \\ t_2 & t_1 & \cdots & t_{-(N-4)} & t_{-(N-3)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ t_{N-1} & t_{N-2} & \cdots & t_1 & t_0 \end{pmatrix}, \tag{3.4}$$

specified by $2N - 1$ parameters. We focus on the situation where $t_j \geq 0$ for all j as in §2. Clearly, when $t_{-(N-i)} = t_i$ for all i, T is a circulant matrix. Although a Toeplitz matrix is not circulant in general, any
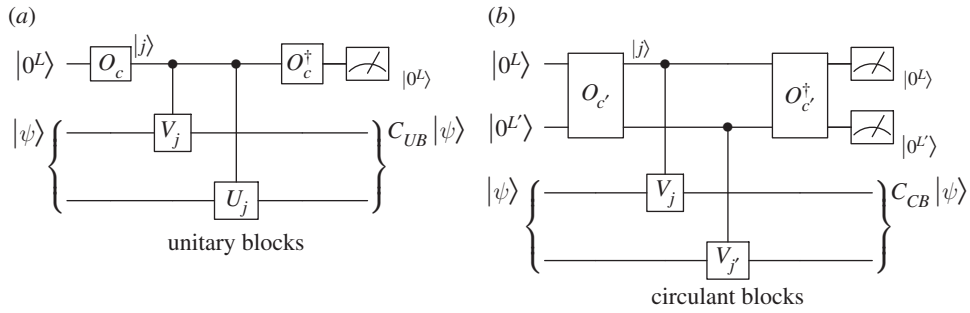
Figure 2. The quantum circuit to implement block circulant matrices with special structures.
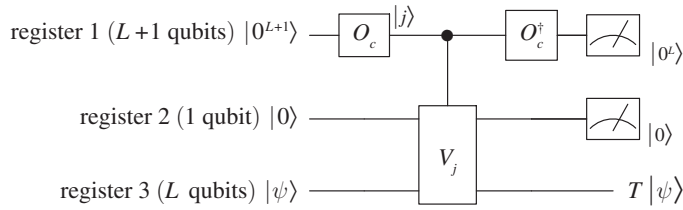

Figure 3. The quantum circuit to implement a Toeplitz matrix. In this figure, $O_c|0^{L+1}\rangle = \sum_{j=0}^{2N-1} c_j|j\rangle$, where $c = (t_0 \, t_{-1} \cdots t_{-(N-1)} \, 0 \, t_{N-1} \cdots t_1)$.

Toeplitz matrix T can be embedded in a circulant matrix [15,35], defined by

$$C_T = \begin{pmatrix} T & B_T \\ B_T & T \end{pmatrix}, \tag{3.5}$$

where B_T is another Toeplitz matrix defined by

$$B_T = \begin{pmatrix} 0 & t_{N-1} & \cdots & t_2 & t_1 \\ t_{-(N-1)} & 0 & \cdots & t_3 & t_2 \\ t_{-(N-2)} & t_{-(N-1)} & \cdots & t_4 & t_3 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ t_{-1} & t_{-2} & \cdots & t_{-(N-1)} & 0 \end{pmatrix}. \tag{3.6}$$

As a result, we use this embedding to implement Toeplitz matrices because

$$\begin{pmatrix} T & B_T \\ B_T & T \end{pmatrix} \begin{pmatrix} \psi \\ 0 \end{pmatrix} = \begin{pmatrix} T\psi \\ B_T\psi \end{pmatrix}. \tag{3.7}$$

Therefore, by implementing C_T, we obtain a quantum state proportional to $|0\rangle T|\psi\rangle + |1\rangle B_T|\psi\rangle$. Then we do a quantum measurement on the single qubit (in the second register in figure 3) to obtain the quantum state $T|\psi\rangle$. The success rate is $\|T|\psi\rangle\|^2$ according to theorem 2.2 under the normalization condition that $\sum_{j=-(N-1)}^{N-1} t_j = \sum_{j=0}^{N-1} c_j = 1$. With the help of amplitude amplification, only $O(1/\|T|\psi\rangle\|)$ applications of the circuit in figure 3 are required.

A Hankel matrix is a square matrix in which each ascending skew-diagonal from left to right is constant, which can be written explicitly as

$$H = \begin{pmatrix} h_{N-1} & h_{N-2} & \cdots & h_1 & h_0 \\ h_{N-2} & h_{N-3} & \cdots & h_0 & h_{-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ h_1 & h_0 & \cdots & h_{-(N-3)} & h_{-(N-2)} \\ h_0 & h_{-1} & \cdots & h_{-(N-2)} & h_{-(N-1)} \end{pmatrix}, \tag{3.8}$$

specified by $2N - 1$ non-negative parameters. A permutation matrix $P = \sigma_x^{\otimes L}$ transforms a Hankel matrix into a Toeplitz matrix. It can be easily verified that $T = HP$ and $H = TP$, in which $t_j = h_j$ for all j. Note

that when N is not a power of two, we need to be careful with the embedding when mapping a circulant matrix into a Hankel matrix. The subspace $\mathrm{span}\{|0\rangle, \ldots, |N-1\rangle\}$ in the implementation of circulant matrices corresponds to the subspace $\mathrm{span}\{|2^L - N\rangle, \ldots, |2^L - 1\rangle\}$ in the implementation of Hankel matrices.

Therefore by inserting the permutation P before the implementation of T, the circuit in figure 3 can be used to implement H, and the success rate is $\|H|\psi\rangle\|^2$ under the normalization condition that $\sum_{j=-(N-1)}^{N-1} h_j = \sum_{j=0}^{N-1} c_j = 1$. With the help of amplitude amplification, only $O(1/\|H|\psi\rangle\|)$ applications are required.

In comparison with existing algorithms, such as that described in [35], the above described quantum circuit provides a better way to realize circulant-like matrices, requiring fewer resources and with lower complexity. For example, only $2 \log N$ qubits are required to implement N-dimensional Toeplitz matrices, which is a significant improvement over the algorithm presented in [35] via sparse Hamiltonian simulations. More importantly, this is an exact method and its complexity does not depend on an error term. It is also not limited to sparse circulant matrices C as in [35]. Moreover, implementation of non-unitary matrices, such as circulant matrices, is not only of importance in quantum computing, but also a significant ingredient in quantum channel simulators [36,37], because the set of Kraus operators in the quantum channel $\rho \mapsto \sum_i K_i \rho K_i^\dagger$ is normally non-unitary [1]. The simplicity of our circuit increases its feasibility in experimental realizations.

4. Circulant Hamiltonians

Hamiltonian simulation is expected to be one of the most important undertakings for quantum computation. It is therefore important to explore the possibility of efficient implementation of circulant Hamiltonians because of their extensive applications. Particularly, the implementation of $\mathrm{e}^{-\mathrm{i}Ct}$ is equivalent to the implementation of continuous-time quantum walks on a weighted circulant graph [38,39]. Moreover, simulation of Hamiltonians is also an important part in the HHL algorithm to solve linear systems of equations [9].

A number of algorithms have been shown to be able to efficiently simulate sparse Hamiltonians [2–8], including the unitary decomposition approach [7]. We show that this approach can be extended to the simulation of dense circulant Hamiltonians. It is well known that circulant matrices are diagonalizable as $\mathrm{e}^{-\mathrm{i}Ct} = F \mathrm{e}^{-\mathrm{i}\Lambda t} F^\dagger$. In general, implementing an arbitrary diagonal unitary requires up to $O(N \log N)$ one- or two-qubit gates [40]. However, when $\{\Lambda_k\}_{k=0}^{N-1}$ can be efficiently computed, one can efficiently implement $\mathrm{e}^{-\mathrm{i}Ct}$ [23,41,42].

In this section, we will focus on the simulation of Hermitian circulant matrices, when $\mathrm{e}^{-\mathrm{i}Ct}$ is unitary. For completeness, we first summarize briefly the unitary decomposition approach in [7] and then discuss how it can be used to efficiently simulate dense circulant Hamiltonians. To simulate $U = \mathrm{e}^{-\mathrm{i}Ct}$, the evolution time t is divided into r segments with $U_r = \mathrm{e}^{-\mathrm{i}Ct/r}$, which can be approximated as $\tilde{U} = \sum_{k=0}^{K} 1/k!(-\mathrm{i}Ct/r)^k$ with error ϵ. It can be proven that if we choose $K = O(\log(r/\epsilon)/\log\log(r/\epsilon)) = O(\log(t/\epsilon)/\log\log(t/\epsilon))$, then $\|U_r - \tilde{U}\| \leq \epsilon/r$ and the total error is within ϵ.

Since $C = \sum_{j=0}^{N-1} c_j V_j$ as given by equation (2.2), we have

$$\tilde{U} = \sum_{k=0}^{K} \frac{(-\mathrm{i}Ct/r)^k}{k!} = \sum_{k=0}^{K} \sum_{j_1,\ldots,j_k=0}^{N-1} \frac{(-\mathrm{i}t/r)^k}{k!} c_{j_1} \cdots c_{j_k} V_{j_1} \cdots V_{j_k}. \tag{4.1}$$

Let $W_{(k,j_1,\ldots,j_k)} = (-\mathrm{i})^k V_{j_1} \cdots V_{j_k}$ and

$$O_\alpha |0^{K+KL}\rangle = \frac{1}{\sqrt{s}} \sum_{k=0}^{K} \sum_{j_1,\ldots,j_k=0}^{N-1} \sqrt{(t/r)^k/k! c_{j_1} \cdots c_{j_k}} |1^k 0^{K-k}\rangle |j_1\rangle \cdots |j_k\rangle |0^{(K-k)L}\rangle, \tag{4.2}$$

where $|1^k 0^{K-k}\rangle$ is the unary encoding of k. Here, s is the normalization coefficient and we choose $r = \lceil t/\ln 2 \rceil$ so that

$$s = \sum_{k=0}^{K} \sum_{j_1,\ldots,j_k=0}^{N-1} \frac{(t/r)^k}{k!} c_{j_1} \cdots c_{j_k} = \sum_{k=0}^{K} \frac{((c_0 + \cdots + c_{N-1})t/r)^k}{k!} \approx 2. \tag{4.3}$$
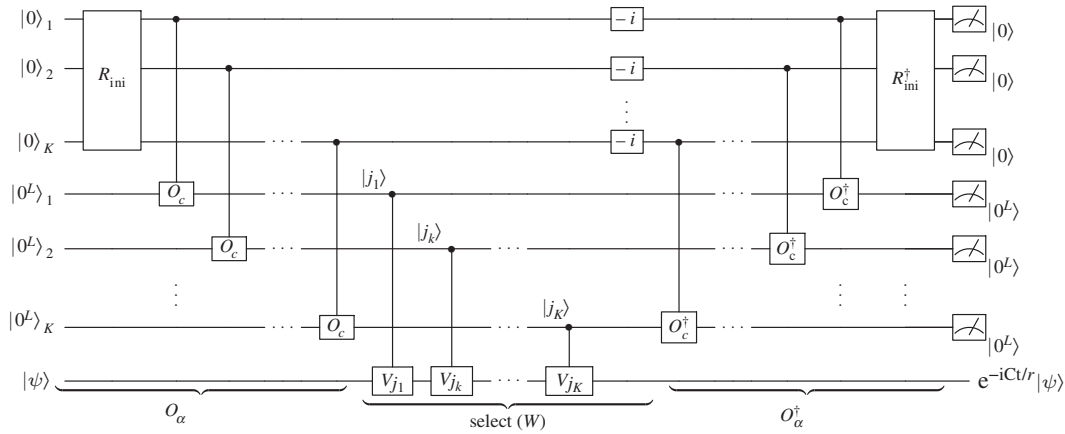
Figure 4. The quantum circuit to implement one segment of circulant Hamiltonians. Here $\boxed{R_{\mathrm{ini}}}\,|0^K\rangle = \sum_{k=0}^{K}\sqrt{(t/r)^k/k!}\,|1^k0^{K-k}\rangle$ and $\boxed{-i} = |0\rangle\langle 0| + (-i)|1\rangle\langle 1|$.

By taking $M = \tilde{U}$, $\alpha_j = \sqrt{(t/r)^k/k!c_{j_1}\cdots c_{j_k}}$, $W_j = W_{(k,j_1,\ldots,j_k)}$, $J = KN^K$ and $m = K + KL$ in lemma 2.1, we have

$$(O_\alpha^\dagger \otimes I)\mathrm{select}(W)(O_\alpha \otimes I)|0^{K+KL}\rangle|\psi\rangle = \frac{1}{s}|0^{K+KL}\rangle\tilde{U}|\psi\rangle + |\Psi^\perp\rangle, \tag{4.4}$$

where $(|0^{K+KL}\rangle\langle 0^{K+KL}| \otimes I)|\Psi^\perp\rangle = 0$. It has been shown in [7] that after one step of oblivious amplitude amplification procedure [29], $U_r = \mathrm{e}^{-\mathrm{i}Ct/r}$ can be simulated within error ϵ/r. The oblivious amplitude amplification procedure avoids the repeated preparations of $|\psi\rangle$ so that $\tilde{U}|\psi\rangle$ can be obtained using only one copy of $|\psi\rangle$, as shown in figure 4. The total complexity depends on the number of gates required to implement $\mathrm{select}(W)$ and O_α.

If C is not Hermitian (and $\tilde{U}$ is not at least approximately unitary), the oblivious amplitude amplification procedure [29] will not be applicable, and then we have to resort to the traditional amplitude amplification [28]. This will lead to a complexity depending exponentially on t because we have to run the amplitude amplification recursively, but the complexity will still depend logarithmically on N.

Theorem 4.1 (Simulation of circulant Hamiltonians). *There exists an algorithm performing* $\mathrm{e}^{-\mathrm{i}Ct}$ *on an arbitrary quantum state* $|\psi\rangle$ *within error* ϵ, *using* $O(t(\log(t/\epsilon)/\log\log(t/\epsilon)))$ *calls of controlled-O_c*[1] *and its inverse, as well as* $O(t(\log N)(\log(t/\epsilon)/\log\log(t/\epsilon)))$ *additional one- and two-qubit gates.*

Proof. We first consider the number of gates used to implement O_α in equation (4.2). It can be decomposed into two steps. The first step is to create the normalized version of the state $\sum_{k=0}^{K}\sqrt{(t/r)^k/k!}\,|1^k0^{K-k}\rangle$ from the initial state $|0^K\rangle$, which takes $O(K)$ consecutive one-qubit rotations on each qubit. We then apply K sets of controlled-O_c to transform $|0^L\rangle$ into $\sum_{j=0}^{N-1}\sqrt{c_j}|j\rangle$ when the control qubit is $|1\rangle$. We therefore need $O(K)$ calls of controlled-O_c and $O(K)$ additional one-qubit gates to implement O_α.

Next we focus on the implementation of

$$\mathrm{select}(W) = \sum_{(k,j_1,j_2,\cdots,j_k)}$$
$$|1^k0^{K-k}\rangle|j_1\rangle\cdots|j_k\rangle|0^{(K-k)L}\rangle\langle 1^k0^{K-k}|\langle j_1|\cdots\langle j_k|\langle 0^{(K-k)L}| \otimes (-\mathrm{i})^k V_{j_1}\cdots V_{j_k}, \tag{4.5}$$

which performs the transformation

$$|1^k0^{K-k}\rangle|j_1\rangle\cdots|j_k\rangle|0^{(K-k)L}\rangle|\psi\rangle \xrightarrow{\mathrm{select}(W)}$$
$$|1^k0^{K-k}\rangle|j_1\rangle\cdots|j_k\rangle|0^{(K-k)L}\rangle(-i)^k V_{j_1}\cdots V_{j_k}|\psi\rangle. \tag{4.6}$$

---

[1]By controlled-O_c, we mean the operation $|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes O_c$.

As $V_j|\ell\rangle = |(\ell - j) \mod N\rangle$, we can transform $|\psi\rangle$ into $V_{j_1}\cdots V_{j_k}|\psi\rangle$ by applying K quantum subtractors between $|j\rangle_{j=j_1,j_2,\ldots,j_k,0,\ldots}$ and $|\psi\rangle$. K phase gates on each of the first K qubits multiply the amplitude by $(-i)^k$. Therefore, select(W) can be decomposed into $O(K \log N)$ one or two-qubit gates.

In summary, $O(K)$ calls of controlled-O_c and its inverse as well as $O(K \log N)$ additional one-qubit gates are sufficient to implement one segment $e^{-iCt/r}$; and the total complexity to implement r segments will be $O(tK)$ calls of controlled-O_c and its inverse as well as $O(tK \log N)$ additional one-qubit gates, where $K = O(\log(t/\epsilon)/\log\log(t/\epsilon))$. ∎

Note that we assumed the spectral norm $\|C\| = 1$. To explicitly put it in the complexity in theorem 4.1, we can simply replace t by $\|C\|t$.

5. Inverse of circulant matrices

Following from §4, we now show that the HHL algorithm can be extended to solve systems of circulant matrix linear equations. We assume C to be Hermitian in this section in order for the phase estimation procedure to work.

Theorem 5.1 (Inverse of circulant matrices). *There exists an algorithm creating the quantum state $C^{-1}|\psi\rangle/\|C^{-1}|\psi\rangle\|$ within error ϵ given an arbitrary quantum state $|\psi\rangle$, using $\tilde{O}(\kappa^2/\epsilon)$ calls of controlled-O_c and its inverse, $O(\kappa)$ calls of O_ψ, as well as $\tilde{O}(\kappa^2 \log N/\epsilon)$ additional one- and two-qubit gates.*[2]

Proof. The basic procedure is the same as the HHL algorithm [9], except that C is a dense circulant matrix rather than sparse as required by the HHL algorithm, which is summarized below.

1. Apply the oracle O_ψ to create the input quantum state $|\psi\rangle$:

$$|0^L\rangle \xrightarrow{O_\psi} |\psi\rangle = \sum_{j=0}^{N-1} b_j|u_j\rangle,$$

   where $\{|u_j\rangle\}_{j=0}^{N-1}$ are the eigenvectors of C.
2. Run phase estimation of the unitary operator $e^{i2\pi C}$:

$$\sum_{j=0}^{N-1} b_j|u_j\rangle \to \sum_{j=0}^{N-1} b_j|u_j\rangle|\Lambda_j\rangle,$$

   where Λ_j are the eigenvalues of C and $\Lambda_j \leq 1$.
3. Perform a controlled-rotation on an ancillary qubit:

$$\sum_{j=0}^{N-1} b_j|u_j\rangle|\Lambda_j\rangle|0\rangle \to \sum_{j=0}^{N-1} b_j|u_j\rangle|\Lambda_j\rangle\left(\frac{1}{(\kappa\Lambda_j)|1\rangle} + \sqrt{1 - \frac{1}{(\kappa^2\Lambda_j^2)}}|0\rangle\right),$$

   where κ is the condition number defined in §2 to make sure that $1/(\kappa\Lambda_j) \leq 1$ for all j. The realization of this controlled-rotation requires the computation of Λ_j^{-1}'s [43].
4. Undo the phase estimation and then measure the ancillary qubit. Conditioned on getting 1, we have an output state $\propto \sum_{j=0}^{N-1} b_j/\Lambda_j|u_j\rangle$ and the success rate $p = \sum_{j=0}^{N-1} |b_j/\kappa\Lambda_j|^2 = \Omega(1/\kappa^2)$.

Error occurs in step 2 in Hamiltonian simulation and phase estimation. The complexity scales sublogarithmically with the inverse of error in Hamiltonian simulation as in theorem 4.1 and scales linearly with it in phase estimation [1]. The dominant source of error is phase estimation. Following from the error analysis in [9], a precision $O(\epsilon/\kappa)$ in phase estimation results in a final error ϵ. Taking the success rate $p = \Omega(1/\kappa^2)$ into consideration, the total complexity would be $\tilde{O}(\kappa^2/\epsilon)$, with the help of amplitude amplification [28]. ∎

For s-sparse Hamiltonians (with at most s non-zero entries in any row or column), the HHL algorithm scales as $\tilde{O}(s^2\kappa^2 \log N/\epsilon)$ [9]. In this work, we extended the HHL procedure to dense Hamiltonians with special structure and proved the scaling is independent of matrix sparsity. This simplification stems from the efficient implementation of select(V) which makes possible the decomposition of C into $O(N)$ terms without introducing $O(N)$ into the computational complexity.

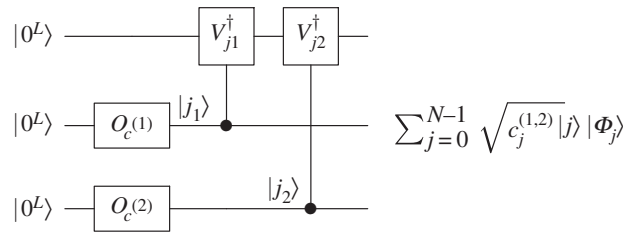[2]We use the symbol $\tilde{O}$ to suppress polylogarithmic factors.

Figure 5. The quantum circuit of $O_{c^{(1,2)}}$. Here $V_j^\dagger = \sum_{k=0}^{N-1} |(k+j) \mod N\rangle\langle k|$ and controlled-$V_j^\dagger$ is a quantum adder.

6. Products of circulant matrices

Products of circulant matrices are also circulant matrices, because a circulant matrix can be decomposed into a linear combination of $\{V_j\}_{j=0}^{N-1}$ that constitute a cyclic group of order N (we have $V_j V_k = V_{(j+k) \mod N}$). Suppose $C^{(1,2)} = C^{(1)}C^{(2)}$ is the product of two circulant matrices $C^{(1)}$ and $C^{(2)}$ which have a parameter vector $c^{(1,2)}$, where

$$c_j^{(1,2)} = \sum_{\substack{j_1,j_2 \\ j_1+j_2 \equiv j \mod N}} c_{j_1}^{(1)} c_{j_2}^{(2)}, \tag{6.1}$$

where $c^{(1)}$ and $c^{(2)}$ are each the parameters of $C^{(1)}$ and $C^{(2)}$. Clearly, when the spectral norm of $C^{(1)}$ and $C^{(2)}$ are one, the spectral norm of $C^{(1,2)}$ is also one. Classically, to calculate the parameters $c^{(1,2)}$ would take up $O(N)$ space. However, in the quantum case, we will show that $O_{c^{(1,2)}}$, encoding $c^{(1,2)}$, can be prepared using one $O_{c^{(1)}}$ and one $O_{c^{(2)}}$. It means that the oracle for a product of circulant matrices can be efficiently prepared when its factor circulants are efficiently implementable, as illustrated in figure 5.

Theorem 6.1 (Products of circulant matrices). *There exists an algorithm creating the oracle $O_{c^{(1,2)}}$, which satisfies*

$$O_{c^{(1,2)}}|0^{3L}\rangle = \sum_{j=0}^{N-1} \sqrt{c_j^{(1,2)}}|j\rangle|\Phi_j\rangle, \tag{6.2}$$

where $|\Phi_j\rangle$ is a unit quantum state dependent on j, using one $O_{c^{(1)}}$, one $O_{c^{(2)}}$ and $O(\log N)$ additional one- and two-qubit gates.

Proof. We need $2L$ ancillary qubits divided into two registers to construct the oracle for the product of two circulant matrices. We start by applying $O_{c^{(1)}}$ and $O_{c^{(2)}}$ on the last 2 registers, we obtain

$$|0^{3L}\rangle \to |0^L\rangle \left(\sum_{j_1=0}^{N-1} \sqrt{c_{j_1}^{(1)}}|j_1\rangle \right) \left(\sum_{j_2=0}^{N-1} \sqrt{c_{j_2}^{(2)}}|j_2\rangle \right). \tag{6.3}$$

In order to encode $c_j^{(1,2)}$ in the quantum amplitudes, we once again apply quantum adders to achieve our goals. By performing the following transformation:

$$|0\rangle|j_1\rangle|j_2\rangle \to |j\rangle|j_1\rangle|j_2\rangle, \tag{6.4}$$

where $j \equiv (j_1 + j_2) \mod N$. This can be achieved using two quantum adders, we obtain the state

$$\sum_{j=0}^{N-1} \sqrt{c_j^{(1,2)}}|j\rangle|\Phi_j\rangle, \tag{6.5}$$

because the amplitude of $|j\rangle$ is equal to $\sqrt{\sum_{\substack{j_1,j_2 \\ j_1+j_2 \equiv j \mod N}} (\sqrt{c_{j_1} c_{j_2}})^2} = \sqrt{c_j^{(1,2)}}$. ∎

This algorithm can be easily extended to implementing oracles for products of d circulants, in which d oracles of factor circulants and dL ancillary qubits are needed. Though the oracle described in theorem 6.1 may not be useful in all quantum algorithms, owing to the additional $|\Phi_j\rangle$ in equation (6.2), it is applicable in §§2 and 4 according to lemma 6.2 (the generalized form of lemma 2.1) described below. It implies that this technique could also be useful in other algorithms related to circulant matrices.

Lemma 6.2. *Let $M = \sum_{\alpha_j} \alpha_j W_j$ be a linear combination of unitaries W_j with $\alpha_j \geq 0$ for all j and $\sum_j \alpha_j = 1$. Let O_α be any operator that satisfies $O_\alpha |0^m\rangle = \sum_j \sqrt{\alpha_j}|j\rangle|\Phi_j\rangle$ (m is the number of qubits used to represent $|j\rangle|\Phi_j\rangle$) and $\mathrm{select}(W) = \sum_j |j\rangle\langle j| \otimes I \otimes W_j$. Then*

$$(O_\alpha^\dagger \otimes I)\,\mathrm{select}(W)(O_\alpha \otimes I)|0^m\rangle|\psi\rangle = |0^m\rangle M|\psi\rangle + |\Psi^\perp\rangle, \tag{6.6}$$

where $(|0^m\rangle\langle 0^m| \otimes I)|\Psi^\perp\rangle = 0$.

Proof.

$$(O_\alpha^\dagger \otimes I)\mathrm{select}(W)(O_\alpha \otimes I)|0^m\rangle|\psi\rangle = (O_\alpha^\dagger \otimes I)\mathrm{select}(W)\sum_j \sqrt{\alpha_j}|j\rangle|\Phi_j\rangle|\psi\rangle$$

$$= (O_\alpha^\dagger \otimes I)\sum_j \sqrt{\alpha_j}|j\rangle|\Phi_j\rangle W_j|\psi\rangle$$

$$(|0^m\rangle\langle 0^m|O_\alpha^\dagger \otimes I)\sum_j \sqrt{\alpha_j}|j\rangle|\Phi_j\rangle W_j|\psi\rangle = |0^m\rangle\sum_{j'}\sqrt{\alpha_{j'}}\langle j'|\langle \Phi_{j'}|\sum_j \sqrt{\alpha_j}|j\rangle|\Phi_j\rangle W_j|\psi\rangle$$

$$= |0^m\rangle\sum_j \alpha_j W_j|\psi\rangle = |0^m\rangle M|\psi\rangle.$$

■

7. Application: solving cyclic systems

Vibration analysis of mechanical structures with cyclic symmetry has been a subject of considerable studies in acoustics and mechanical engineering [14,17]. Here we provide an example where the above proposed quantum scheme can outperform classical algorithms in solving the equation of motion for vibrating and rotating systems with certain cyclic symmetry.

The equation of motion for a cyclically symmetric system consisting of N identical sectors, as shown in figure 6, can be written as

$$M\ddot{q} + D\dot{q} + Kq = f, \tag{7.1}$$

where q and f are N-dimensional vectors, denoting the displacement of and the external force acting on each individual sector, respectively. The mass, damping and stiffness matrices are all circulants, represented by $M = \mathrm{circ}(m_1, m_2, \ldots, m_N)$, $D = \mathrm{circ}(d_1, d_2, \ldots, d_N)$ and $K = \mathrm{circ}(s_1, s_2, \ldots, s_N)$.

Assume all sectors have the same mass ($M \propto I$) and there is zero damping ($D = 0$). If the system is under the so-called travelling wave engine order excitation, the equation of motion can be simplified as [14]:

$$\ddot{q} + Kq = f\,e^{in\Omega t}, \tag{7.2}$$

where the travelling wave is characterized by $f_j = f\,e^{i2\pi nj/N}$ for the external force vector f, n is the order of excitation and Ω is the angular frequency of the excitation. We search for solutions of the form $q = q_0\,e^{in\Omega t}$, which leads to

$$(K - n^2\Omega^2 I)q_0 = f. \tag{7.3}$$

Since $K - n^2\Omega^2 I$ is a circulant matrix, we can use theorem 5.1 to calculate

$$q_0 = (K - n^2\Omega^2 I)^{-1}f.$$

It is important to consider the conditions under which theorem 5.1 works.

1. $K - n^2\Omega^2 I$ *is Hermitian.* This is generally true for symmetric cyclic systems, where the coupling between q_j and q_{j+d} and the coupling between q_j and q_{j-d} are physically the same for any sector j and distance d.
2. $K - n^2\Omega^2 I$ *has non-negative (or non-positive) entries.* Although this is not in general true, theorem 5.1 will work under a slight modification. We observe that the off-diagonal elements of $K - n^2\Omega^2 I$ are always negative because the coupling force between two connecting sectors is always in the opposite direction to their relative motion.

   — If the diagonal elements of $K - n^2\Omega^2 I$ are also negative, then no modification to the proposed procedure is necessary.
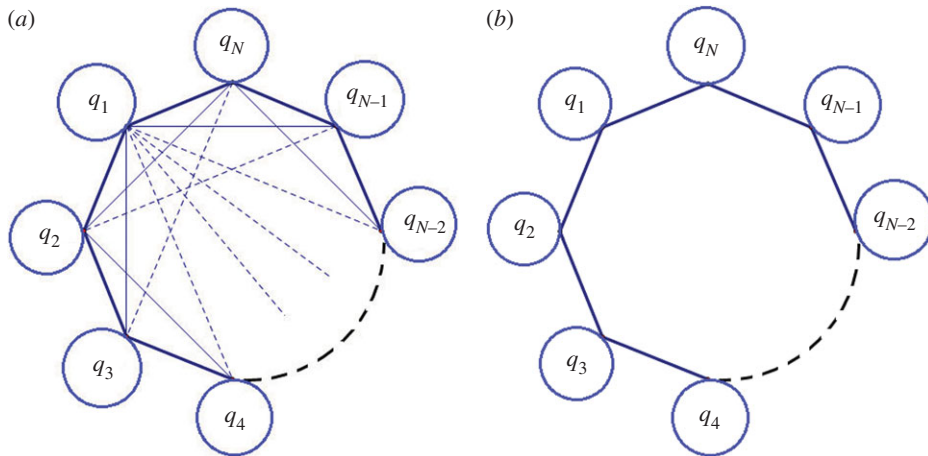
Figure 6. Topology diagram of an N-sector cyclic system. (*a*) A general cyclic system with coupling between any two sectors which can be solved using theorem 5.1. (*b*) A cyclic system with nearest-neighbour coupling which can be solved using the HHL algorithm [9].

— If the diagonal elements of $K - n^2\Omega^2I$ are positive, using the technique stated in §3.1, we can simply replace select(V) with $\text{Ref}_0 \cdot \text{select}(V)$, where $\text{Ref}_0 = |0^L\rangle\langle 0^L| - 2I$ is a reflection operator operating on the first register.

3. *The condition number κ of $K - n^2\Omega^2I$ is small.* This is true when the couplings among sectors are relatively weak—when $|K_0 - n^2\Omega^2| \gg K_1$ where K_0 characterizes the coupling between a sector and the exterior and K_1 characterizes the coupling among sectors.

4. *The corresponding oracle O_c of the circulant matrix $K - n^2\Omega^2I$ can be efficiently implemented.* It requires either there is a special structure of $K - n^2\Omega^2I$ or the information of $K - n^2\Omega^2I$ is stored in a qRAM in advance.

If all four conditions are satisfied, we have an exponential speed-up compared to classical computation. Note that the output q_0 is stored in quantum amplitudes, which cannot be read out directly. However, further computation steps can efficiently provide practically useful information about the system from the vector q_0, for example the expectation value $q_0^\dagger M q_0$ for some linear operator M or the similarity between two cyclic systems $\langle q_0'|q_0\rangle$ [9]. This type of speed-up is not achievable classically for it takes at least $O(N)$ steps to read out the value of q_0. It is also worth noting that the proposed algorithm, in contrast to previous quantum algorithms [3–7,9,35], works for dense matrices $K - n^2\Omega^2I$. It means that the cyclic systems need not be subject to nearest-neighbour coupling.

8. Conclusion

In this paper, we present efficient quantum algorithms for implementing circulant (as well as Toeplitz and Hankel) matrices and block circulant matrices with special structures, which are not necessarily sparse or unitary. These matrices have practically significant applications in physics, mathematics and engineering-related fields. The proposed algorithms provide exponential speed-up over classical algorithms, requiring fewer resources ($2 \log N$ qubits) and having lower complexity ($O(\log N/\|C|\psi\rangle\|)$) in comparison with existing quantum algorithms. Consequently, they perform better in quantum computing and are more feasible to experimental realization with current technology. Obstacles still exist, though, in the efficient realization of the oracles to generate the components of the circulant matrices.

Besides the implementation of circulant matrices, we discover that we can perform the HHL algorithm on circulant matrices to implement the inverse of circulant matrices, by adopting the Taylor series approach to efficiently simulate circulant Hamiltonians. Owing to the special structure of circulant matrices, we prove that they are one of the types of the dense matrices that can be efficiently simulated. Being able to implement the inverse of circulant matrices opens a door to solving a variety of real-world problems, for example, solving cyclic systems in vibration analysis. Finally, we show that it is possible to construct oracles for products of circulant matrices using the oracles for their factor circulants, a technique that will be useful in related algorithms.

References

1. Nielsen MA, Chuang IL. 2010 *Quantum computation and quantum information*. Cambridge, UK: Cambridge University Press.

2. Lloyd S. 1996 Universal quantum simulators. *Science* **273**, 1073. (doi:10.1126/science.273.5278.1073)

3. Berry DW, Ahokas G, Cleve R, Sanders BC. 2007 Efficient quantum algorithms for simulating sparse Hamiltonians. *Commun. Math. Phys.* **270**, 359–371. (doi:10.1007/s00220-006-0150-x)

4. Childs AM, Kothari R. 2010 Simulating sparse Hamiltonians with star decompositions. In *Theory of quantum computation, communication, and cryptography, TQC 2010* (eds W van Dam, VM Kendon, S Severini), pp. 94–103. Berlin, Heidelberg, Germany: Springer-Verlag.

5. Wiebe N, Berry DW, Høyer P, Sanders BC. 2011 Simulating quantum dynamics on a quantum computer. *J. Phys. A Math. Theoret.* **44**, 445308. (doi:10.1088/1751-8113/44/44/445308)

6. Poulin D, Qarry A, Somma R, Verstraete F. 2011 Quantum simulation of time-dependent Hamiltonians and the convenient illusion of Hilbert space. *Phys. Rev. Lett.* **106**, 170501. (doi:10.1103/PhysRevLett.106.170501)

7. Berry DW, Childs AM, Cleve R, Kothari R, Somma RD. 2015 Simulating Hamiltonian dynamics with a truncated Taylor series. *Phys. Rev. Lett.* **114**, 090502. (doi:10.1103/PhysRevLett.114.090502)

8. Berry DW, Childs AM, Kothari R. 2015 Hamiltonian simulation with nearly optimal dependence on all parameters. In *Proc. IEEE 56th Annual Symp. on Foundations of Computer Science*, pp. 792–809. IEEE. See http://www.ieee.org.

9. Harrow AW, Hassidim A, Lloyd S. 2009 Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **103**, 150502. (doi:10.1103/PhysRevLett.103.150502)

10. Childs AM, Kothari R. 2009 Limitations on the simulation of non-sparse Hamiltonians. See http://arxiv.org/abs/0908.4398.

11. Rietsch K. 2003 Totally positive Toeplitz matrices and quantum cohomology of partial flag varieties. *J. Am. Math. Soc.* **16**, 363–392. (doi:10.1090/S0894-0347-02-00412-5)

12. Haupt J, Bajwa WU, Raz G, Nowak R. 2010 Toeplitz compressed sensing matrices with applications to sparse channel estimation. *Inform. Theory IEEE Trans.* **56**, 5862–5875. (doi:10.1109/TIT.2010.2070191)

13. Noschese S, Pasquini L, Reichel L. 2013 Tridiagonal Toeplitz matrices: properties and novel applications. *Numer. Linear Algebra Appl.* **20**, 302–326. (doi:10.1002/nla.1811)

14. Olson BJ, Shaw SW, Shi C, Pierre C, Parker RG. 2014 Circulant matrices and their application to vibration analysis. *Appl. Mech. Rev.* **66**, 040803. (doi:10.1115/1.4027722)

15. Ng MK. 2004 *Iterative methods for Toeplitz systems*. New York, NY: Oxford University Press.

16. Peller V. 2012 *Hankel operators and their applications*. Berlin, Germany: Springer Science & Business Media.

17. Kaveh A, Rahami H. 2011 Block circulant matrices and applications in free vibration analysis of cyclically repetitive structures. *Acta Mech.* **217**, 51–62. (doi:10.1007/s00707-010-0382-x)

18. Rjasanow S. 1994 Effective algorithms with circulant-block matrices. *Linear Algebra Appl.* **202**, 55–69. (doi:10.1016/0024-3795(94)90184-8)

19. Tee GJ. 2005 Eigenvectors of block circulant and alternating circulant matrices. *Res. Lett. Info. Math. Sci*. **8**, 123–142.

20. Combescure M. 2009 Block-circulant matrices with circulant blocks, weil sums, and mutually unbiased bases. II. The prime power case. *J. Math. Phys.* **50**, 032104. (doi:10.1063/1.3078420)

21. Petrou M, Petrou C. 2010 *Image processing: the fundamentals*. New York, NY: John Wiley & Sons.

22. Delanty M, Steel MJ. 2012 Discretely-observable continuous time quantum walks on Moöbius strips and other exotic structures in 3D integrated photonics. *Phys. Rev. A* **86**, 043821. (doi:10.1103/PhysRevA.86.043821)

23. Qiang X, Loke T, Montanaro A, Aungskunsiri K, Zhou X, O'Brien JL, Wang J, Matthews JC. 2016 Efficient quantum walk on a quantum processor. *Nat. Commun.* **7**, 11511. (doi:10.1038/ncomms11511)

24. Ye K, Lim LH. 2016 Every matrix is a product of Toeplitz matrices. *Found. Comput. Math.* **16**, 577–598. (doi:10.1007/s10208-015-9254-z)

25. Golub GH, Van Loan CF. 2012 *Matrix computations*, vol. 3. Baltimore MD: John Hopkins University Press.

26. Cuccaro SA, Draper TG, Kutin SA, Moulton DP. 2004 A new quantum ripple-carry addition circuit. See http://arxiv.org/abs/quant-ph/0410184.

27. Draper TG, Kutin SA, Rains EM, Svore KM. 2004 A logarithmic-depth quantum carry-lookahead adder. See http://arxiv.org/abs/quant-ph/0406142.

28. Brassard G, Hoyer P, Mosca M, Tapp A. 2002 Quantum amplitude amplification and estimation. *Contemp. Math.* **305**, 53–74. (doi:10.1090/conm/305/05215)

29. Berry DW, Childs AM, Cleve R, Kothari R, Somma RD. 2014 Exponential improvement in precision for simulating sparse Hamiltonians. In *Proc. of the 46th Annual ACM Symp. on Theory of Computing*, pp. 283–292. ACM, New York, NY.

30. Grover L, Rudolph T. 2002 Creating superpositions that correspond to efficiently integrable probability distributions. See http://arxiv.org/abs/abs/quant-ph/0208112.

31. Kaye P, Mosca M. 2004 Quantum networks for generating arbitrary quantum states. See http://arxiv.org/abs/quant-ph/0407102.

32. Soklakov AN, Schack R. 2006 Efficient state preparation for a register of quantum bits. *Phys. Rev. A* **73**, 012307. (doi:10.1103/PhysRevA.73.012307)

33. Giovannetti V, Lloyd S, Maccone L. 2008 Architectures for a quantum random access memory. *Phys. Rev. A* **78**, 052310. (doi:10.1103/PhysRevA.78.052310)

34. Lloyd S, Mohseni M, Rebentrost P. 2013 Quantum algorithms for supervised and unsupervised machine learning. See http://arxiv.org/abs/1307.0411.

35. Mahasinghe A, Wang JB. 2016 Efficient quantum circuits for Toeplitz and Hankel matrices. *J. Phys. A Math. Theoret.* **49**, 275301. (doi:10.1088/1751-8113/49/27/275301)

36. Lu H *et al.* 2015 Universal digital photonic single-qubit quantum channel simulator. See http://arxiv.org/abs/1505.02879.

37. Wang D-S, Sanders BC. 2015 Quantum circuit design for accurate simulation of qudit channels. *New J. Phys.* **17**, 043004. (doi:10.1088/1367-2630/17/4/043004)

38. Manouchehri K, Wang JB. 2014 *Physical implementation of quantum walks*. Berlin, Germany: Springer.

39. Childs AM. 2010 On the relationship between continuous- and discrete-time quantum walk. *Commun. Math. Phys.* **294**, 581–603. (doi:10.1007/s00220-009-0930-1)

40. Bullock SS, Markov IL. 2004 Asymptotically optimal circuits for arbitrary n-qubit diagonal computations. *Q. Inform. Comput.* **4**, 27–47.

41. Childs AM. 2004 Quantum information processing in continuous time. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

42. Zhou S, Loke T, Izaac J, Wang J. 2017 Quantum fourier transform in computational basis. *Q. Inform. Process.* **16**, 82. (doi:10.1007/s11128-017-1515-0)

43. Cao Y, Papageorgiou A, Petras I, Traub J, Kais S. 2013 Quantum algorithm and circuit design solving the Poisson equation. *New J. Phys.* **15**, 013021. (doi:10.1088/1367-2630/15/1/013021)