Research Article

Vibration Analysis of Cyclic Symmetrical Systems by Quantum Algorithms

Anuradha Mahasinghe

Department of Mathematics, University of Colombo, Colombo 03, Sri Lanka

Correspondence should be addressed to Anuradha Mahasinghe; anuradhamahasinghe@gmail.com

Received 27 April 2019; Accepted 22 May 2019; Published 11 June 2019

Academic Editor: Xiangyu Meng

Copyright © 2019 Anuradha Mahasinghe. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Quantum computers have provided exponentially faster solutions to several physical and engineering problems over existing classical solutions. In this paper we present two quantum algorithms to analyze the forced vibration of a mechanical system with cyclic symmetry. Our main algorithm solves the equation of motion of an undamped and nonelastic rotating system with cyclic symmetry consisting of \( n \) sectors, by encoding the displacements of each sector at time \( t \) in a quantum state. The runtime of this algorithm is \( \text{polylog} n \) and \( \text{polylog} t \), thus exponentially faster than the analogous classical algorithm. Also we consider damped and elastic systems with cyclic symmetry and present another quantum algorithm to solve it in runtime \( \text{polylog} n \) and polynomial in \( t \).

1. Introduction

Several quantum algorithms were proven to run exponentially faster than the best-known classical algorithms. These algorithms were built to solve specific mathematical problems, for which the classical techniques seemed inadequate, with respect to resource cost and efficiency. Quantum algorithms designed for tasks such as searching databases [1], prime factoring [2], and phase estimation [3] are of fundamental importance in quantum computing. Many later developments in quantum algorithms over the circuit-gate framework are based on these primary algorithms.

A remarkable quantum algorithm designed by Harrow, Hassidim, and Loyd in 2009 [4], which is widely known as the HHL algorithm, was of particular importance in the field in the last decade. Based upon the more primary quantum eigenvalue estimation algorithm, its objective is to solve a sparse linear system of equations efficiently, which is unimaginable in the realm of classical algorithms. More precisely, the HHL algorithm is capable of solving a sparse linear system of \( n \) equations and \( n \) variables in a runtime and resource cost \( O(\text{polylog} n) \). It is thus exponentially faster than the most efficient classical algorithm for solving a linear system. Experimental verification of the HHL algorithm can be found in [5].

The HHL is one of the powerful quantum algorithms, with promising applications to different fields. Also it has been the key ingredient in several new quantum algorithms, yielding an exponential efficiency gain over best-known classical techniques. Examples include the quantum algorithm to solve the \( d \)-dimensional Poisson equation with complexity linear in \( d \) [6], one for solving a system of first-order differential equations [7] and another algorithm for curve fitting and calculating the effective resistance of electric networks [8]. It is interesting to see that most of these applications solve specific problems in the physical world, with straightforward applications to engineering. Therefore, it is an interesting task to find more physical and engineering problems, for which moderated versions of the HHL could be used to provide exponentially faster solutions over existing classical algorithms.

In this paper, we consider one of the important problems in mechanics and engineering: the forced vibration of a mechanical system with cyclic symmetry, for which we develop efficient quantum algorithms, based on the HHL approach. Vibration analysis of cyclic symmetrical systems
had been a subject of interest in different disciplines such as acoustics and mechanical engineering. It is customary to describe the dynamics of such systems using ordinary differential equations, in which the coefficients are distributed in a cyclic symmetrical way. In many situations, large-scale computational models are required to solve them. Solving them classically requires a large resource cost as well as a considerable time, since both scale rapidly with the number of components in the system. This is the context in which we develop our quantum algorithms.

The paper is organized as follows: In Section 2 we present the equation of motion for the forced vibration of a mechanical system with cyclic symmetry. In Section 3 we describe the relevant mathematical principles and transformations and express the solution in a particular form, which enables efficient quantum implementation. In Section 4 we present the corresponding quantum algorithms, describing why our efficiency gain is impossible with classical or other potential quantum methods. Section 5 contains our discussion of the methods.

2. Vibrating Systems with Cyclic Symmetry

A vibrating system with cyclic symmetry, which is also called a rotationally periodic system, is a system with \( n \) sectors, each of which contains an identical mechanical structure with identical coupling to forward nearest neighbors and to ground [9]. For simplicity, we consider a system in which each sector has a single degree of freedom. It can be considered as a cyclic chain of \( n \) identical and identically coupled oscillators with the dynamics of which are captured by the dimensionless transverse displacements.

The nature of the system imposes a cyclic structure on the factors effecting these displacements. For example, the acceleration of the system couples the masses to be arranged in a cyclic symmetrical way, and the product of the coupled masses and accelerations will be equal to the force applied to the system, if the other factors effecting the acceleration of the system were negligible. If the factors such as damping and elasticity were to be considered, they also act in a cyclic symmetrical way on each sector.

Thus, the main factor which distinguishes such a system from a general mechanical system is what is called cyclic symmetry. Mathematically, this is characterized by the notion of circulant matrices. A circulant matrix \( C \) is a square matrix in which the elements of each row of \( C \) are identical to those of the previous row but are moved one position to the right and wrapped around [10]. More specifically, the circulant matrix \( C \), which is symbolized by \( \text{circ}(c_1, c_2, \ldots, c_n) \) is the following square matrix.

\[
C = \begin{pmatrix}
c_1 & c_2 & \cdots & c_n \\
c_n & c_1 & \cdots & c_{n-1} \\
\vdots & \vdots & \ddots & \vdots \\
c_2 & c_3 & \cdots & c_1
\end{pmatrix}.
\] (1)

In the forthcoming subsection we discuss how the dynamics of vibrating systems are governed by circulant matrices.

2.1. Simple Vibrating Systems with Cyclic Symmetry. The motion of the simplest vibrating system with cyclic symmetry can be described by Newton’s second law. We will refer to this as the simple vibrating system with cyclic symmetry. In such a system, the accelerations of the \( n \) sectors are coupled by a circulant matrix which contains the individual masses of each sector, in which the product with the acceleration vector is equal to the coupled forcing vector, obeying Newton’s second law of motion. Therefore, the motion of a simple vibrating system with cyclic symmetry is described by the equation

\[
\ddot{M}u = f,
\] (2)

where

\[
M = \text{circ}(m_1, m_2, \ldots, m_n),
\] (3)

\( m_1, m_2, \ldots, m_n \) being the individual masses of the \( n \) sectors.

Let the forces acting on the coupled system be denoted by \( f_1, f_2, \ldots, f_n \). In a practical perspective, the masses will be distributed in a diagonal matrix. However, as an inspiration to the damped elastic vibration which we will discuss later, we may treat the matrix \( M \) as a circulant. The system represents much complex situation in which the masses are supposed to be arranged with circulant symmetry. It should be noted that each \( f_j \) is time-varied and hence a function of time. Thus, we consider the time-variation of the forcing vector described by

\[
f_j(t) = f_j(0)e(t),
\] (4)

where \( e(t) \) is a function of time and \( f_j(0) \) is the \( j \)th initial force applied on the coupled system. In most of the practical situations, \( e(t) \) is a trigonometric function, in particular either sine or cosine, or a combination of them. The function \( e(t) \) in our case is supposed to be a function that can be encoded and efficiently implemented in a quantum circuit. This assumption is due to the fact that forcing functions normally take sine or cosine forms, which can be implemented efficiently in a quantum circuit.

2.2. Damped Elastic Vibrating Systems with Cyclic Symmetry. Consider a simple vibrating system with cyclic symmetry, with additional effects due to damping and elasticity. Such a system has \( n \) masses, distributed in a circulant matrix, coupled with accelerations, as in a simple vibrating system with cyclic symmetry, with additional effects (accelerations) due to damping and stiffness. Adjacent masses are supposed to be elastically coupled through linear springs, each with nondimensional stiffness \( v \). An individual mass, together with the forward nearest neighbor elastic coupling, forms one fundamental sector. We suppose there are \( n \) such sectors in the overall system.

This vibrating system is modelled by a system of second-order ordinary differential equations, in which the mass, damping, and stiffness matrices are circulants denoted by

\[
M = \text{circ}(m_1, m_2, \ldots, m_n),
\]

\[
D = \text{circ}(d_1, d_2, \ldots, d_n),
\]

and

\[
K = \text{circ}(k_1, k_2, \ldots, k_n),
\]

respectively, and \( u \) is the displacement vector of the coupled system [9]. Thus, the dynamics of the this system is described by

\[
M\ddot{u} + D\dot{u} + Ku = f.
\] (5)
In the forthcoming section, we describe the approximated solutions for (2) and (5) in detail. Our quantum algorithms will be motivated by these solutions.

3. Solving Vibrating Systems

3.1. Diagonalization of Circulant Matrices. Due to the symmetry in the distribution of elements in a circulant matrix, it has interesting spectral properties. One may find a descriptive analysis of the spectral properties of circulants in [10]. Considering the circulant $C = circ(c_1, c_2, \ldots, c_n)$, we define the representer of $C$ as the complex polynomial,

$$ p(z) = c_1 + c_2 z + c_3 z^2 + \cdots + c_n z^{n-1}. \quad (6) $$

Then, it turns out that the eigenspectrum of $C$ is the list of complex numbers $p(1), p(\omega), p(\omega^2), \ldots, p(\omega^{n-1})$, where $\omega$ is the $n^{th}$ root of unity given by $\omega = \exp(2\pi i/n)$. Considering the special circulant matrix of dimensions $n \times n$, named the cyclic forward shift, denoted by $\Pi$, defined as,

$$ \Pi = circ(0, 1, 0, 0, \ldots, 0), \quad (7) $$

it is possible to find an alternative representation of a circulant matrix $C$, which restates the circulant as a linear combination of powers of the cyclic forward shift, with the same coefficients as in the representer of $C$, as follows:

$$ C = c_1 I + c_2 \Pi + c_3 \Pi^2 + \cdots + c_n \Pi^{n-1}. \quad (8) $$

The cyclic forward shift matrix is readily diagonalized by the Fourier matrix of order $n$. More precisely, it is possible to express the cyclic forward shift matrix as

$$ \Pi = F^T \Omega F, \quad (9) $$

where $F$ is the $n \times n$ Fourier matrix defined by

$$ F = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)(n-1)} \end{pmatrix}, \quad (10) $$

and

$$ \Omega = \text{diag}(1, \omega, \omega^2, \ldots, \omega^{n-1}). \quad (11) $$

Then it turns out that also the general circulant matrix $C$ is diagonalized by the Fourier matrix as follows:

$$ C = F^T \text{diag}(p(1), p(\omega), \ldots, p(\omega^{n-1})) F. \quad (12) $$

Here $p(1), p(\omega), \ldots, p(\omega^{n-1})$ are the eigenvalues of the circulant $C$. Therefore, the eigenvalues of a circulant are derivable simply by Fourier transforming a single row or a column.

3.2. Central Differences. The first- and second-order central differences of a variable $u$ are defined as follows:

$$ \dot{u} = \frac{u_{t+1} - u_{t-1}}{2h}, \quad (13) $$

$$ \ddot{u} = \frac{u_{t+1} - 2u_t + u_{t-1}}{h^2}. \quad (14) $$

In a domain with $p + 2$ points, with the assumption that the value of $u$ would be zero at the two extremes of the domain, that is,

$$ u_0 = u_{p+1} = 0, \quad (15) $$

and the central differences are expressible in terms of two tridiagonal matrices $L_1$ and $L_2$. We write down the discretized approximations of the derivatives using these matrices,

$$ \dot{u} = L_1 u, \quad (16) $$

$$ \ddot{u} = L_2 u, \quad (17) $$

for $u$ being a vector $(u(1), u(2), \ldots, u(p))^T$, while $\dot{u}$ and $\ddot{u}$ are defined similarly. The matrices $L_1$ and $L_2$ correspond to the first- and second-order central differences and then take the following form:

$$ L_1 = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 0 & 1 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 0 \end{pmatrix}, \quad (18) $$

$$ L_2 = \begin{pmatrix} -2 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -2 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -2 \end{pmatrix}. \quad (19) $$

Although the two tridiagonal matrices in (18) and (19) are not circulants, they exhibit a certain degree of symmetry, close to circulant. Such symmetry is studied in mathematics under Toeplitz operators. More precisely, a matrix is said to be Toeplitz, if its elements are constant along all diagonals parallel to the main diagonal. An immediate consequence of this definition is that, Toeplitz matrices are closed under matrix addition and scalar multiplication (we will use this fact in approximating the displacements of damped elastic vibrating systems with cyclic symmetry). Further, it follows from this definition that a circulant is indeed a Toeplitz. Also the two noncirculants $L_1$ and $L_2$ are Toeplitz.
3.3. Solution of a Simple Vibrating System with Cyclic Symmetry. First, we consider the simplest forced vibration in which the damping and elasticity effects can simply be ignored. Recall the motion of such a system is governed by (2), where \( M \) is the circulant matrix containing the individual masses; the system is in a coupled state in its present form. Since this condition would not support further analysis, our first step would be the decoupling the system, using diagonalization results of circulants described in Section 3.1.

Accordingly, it is straightforward to see that the diagonalized version of (2) takes the form

\[
F^T \Lambda_M F \ddot{u} = f, \tag{20}
\]

where \( \Lambda_M \) is the diagonal matrix consisting of the eigenvalues \( \lambda_1, \lambda_2, \ldots, \lambda_n \) of the mass circulant \( M \)

\[
\Lambda_M = \text{diag} (\lambda_1, \lambda_2, \ldots, \lambda_n). \tag{21}
\]

Left-multiplying both sides of this equation by \( F \),

\[
\Lambda_M F \ddot{u} = Ff. \tag{22}
\]

For simplicity, we assume the circulant \( M \) to be invertible. That is, all eigenvalues of \( M \) are non-zero, and the diagonal matrix \( \Lambda_M \) is hence invertible, in which the inverse consists of eigenvalue reciprocals of \( M \);

\[
\Lambda_M^{-1} = \text{diag} \left( \frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \ldots, \frac{1}{\lambda_n} \right). \tag{23}
\]

However, this assumption can be lifted, using \( M^* \), the Moore–Penrose pseudoinverse of \( \Lambda_M \), in case \( M \) is singular. That is equivalent to multiplying both sides of (20) by the pseudoinverse of \( M \), as it can be seen that the pseudoinverse of a circulant \( C \) with eigenvalues \( \xi_1, \xi_2, \ldots, \xi_n \) is given by

\[
C^* = F^T \Lambda^* F, \tag{24}
\]

where

\[
\Lambda^* = \text{diag} (\xi_1^+, \xi_2^+, \ldots, \xi_n^+), \tag{25}
\]

and

\[
\xi_j^+ = \begin{cases} 
\frac{1}{\xi_j}, & \text{if } \xi_j \neq 0 \\
0, & \text{otherwise.}
\end{cases} \tag{26}
\]

Thus, we arrive at the following (in which the eigenvalue reciprocal matrix \( \Lambda_M^{-1} \) would be replaced by \( \Lambda_M^* \) if \( M \) is singular):

\[
\ddot{u} = F^T \Lambda_M^{-1} F f. \tag{27}
\]

By the spectral properties of circulant matrices, it can be easily seen that the decoupled force of the \( j^{th} \) sector at time \( t \) (the \( j^{th} \) element of \( F f \)) is a function of \( j, \omega, t \) and \( f_1, f_2, \ldots, f_n \). Let this be denoted by \( G(j, \omega, f(t)) \). Since each entry of the Fourier transformed forcing vector is a linear combination of \( f_j \)'s, \( G(j, \omega, f(t)) \) can be alternatively expressed as follows:

\[
G(j, \omega, f(t)) = G(j, \omega, f_1(0), f_2(0), \ldots, f_n(0)) h(t). \tag{28}
\]

Since it is possible to derive an expression for the acceleration of each sector at a given time, now we discretize the domain with respect to time and approximate the displacements by central differences. Assuming the initial displacement of each sector is zero, we apply the central difference formula in (17).

Then the discretized version of the problem is the linear system:

\[
Ax = b, \tag{29}
\]

where

(i) \( A \) is the real symmetric \( 3 \times 3 \)–sparse square matrix of order \( np \), given by \( A = L_n \otimes L_3 \).

(ii) \( x \) is the \( np \times 1 \) vector consisting of \( n \) blocks, where each block has \( p \) entries. The \( k^{th} \) element of the \( j^{th} \) block gives \( u(j,k) \); the displacement of the \( j^{th} \) sector at time \( t = k \).

(iii) \( b \) is the \( np \times 1 \) vector consisting of \( n \) blocks, where each block has \( p \) entries. The \( k^{th} \) element of the \( j^{th} \) block gives \( G(j, \omega, f_1(0), f_2(0), \ldots, f_n(0)) h(k) \).

3.4. Solution of a Damped Elastic Vibrating System with Cyclic Symmetry. Recall that the forced vibration of an \( n \)–sector damped elastic vibrating system with cyclic symmetry is governed by the system of ordinary differential equations

\[
M \ddot{q} + D \dot{q} + K q = f, \tag{30}
\]

where \( M, D, \) and \( K \) are diagonal matrices corresponding to the mass, damping, and stiffness, and \( f \) is the forcing vector. Using the diagonalization of circulants we described previously, this can be restated as

\[
F^T \Lambda_M F \ddot{q} + F^T \Lambda_D F \dot{q} + F^T \Lambda_K q = F f, \tag{30}
\]

where \( \Lambda_M, \Lambda_D, \) and \( \Lambda_K \) are diagonal matrices consisting of the eigenvalues of \( M, D, \) and \( K \), respectively. Using the transformation \( q = Fu \) and left-multiplying by \( F \), we arrive at the decoupled system:

\[
\Lambda_M q + \Lambda_D \dot{q} + \Lambda_K q = F f. \tag{31}
\]

In other words, we have a system of \( n \) equations, in which the \( j^{th} \) equation is expressed as

\[
\lambda_j \ddot{q}_j + \mu_j \dot{q}_j + \nu_j q_j = (Ff)_j, \tag{32}
\]

where \( \lambda_j, \mu_j, \) and \( \nu_j \) denote the \( j^{th} \) eigenvalue of \( M, D, \) and \( K \) respectively, while the \( j^{th} \) term of the Fourier transformed forcing vector is denoted by \( (Ff)_j \). We keep the same notation as in the case of simple vibrating system:

\[
\lambda_j \ddot{q}_j + \mu_j \dot{q}_j + \nu_j q_j = G(j, \omega, f(t)) \tag{33}
\]


Discretizing the domain with \( p \) points, again we make use of the central differences given by (16) and (17). Then the numerical solution for the \( j^{th} \) system is given by

\[
(\lambda_j L_2 + \mu_j L_1 + \nu_j I) q_j = (Ff)_j,
\]

where \( q_j \) is the \( p \times 1 \) vector given by \((q_{j(1)}, q_{j(2)}, \cdots, q_{j(mp)})^T\), and \( I \) is the \( p \times p \) identity matrix.

Recall [4] describes an efficient quantum algorithm for this task; we make use a similar technique as a part of our algorithms in solving the vibration problem.

Thus, the original problem boils down to solve the linear system:

\[
\hat{A}\hat{x} = \hat{b}, \quad (37)
\]

where

(i) \( \hat{A} \) is the \( 3\times3 \) sparse block diagonal matrix of dimensions \( np \times np \), in which the \( j^{th} \) block is the Toeplitz matrix \( T_j \).

(ii) \( \hat{x} \) is the \( np \times 1 \) vector consisting of \( n \) blocks, where each block has \( p \) entries. The \( k^{th} \) element in the \( j^{th} \) block gives \( g(j, k) \), the Fourier transformed displacement of the \( j^{th} \) sector at time \( t = k \).

(iii) \( \hat{b} \) is the \( np \times 1 \) vector consisting of \( n \) blocks, where each block has \( p \) entries. The \( k^{th} \) element in the \( j^{th} \) block gives \( G(j, \omega, f(0), f_1(0), \cdots, f_m(0))h(k) \).

Thus, we observe that in both cases, the original problem can be transformed into a sparse linear system of equations. Recall [4] describes an efficient quantum algorithm for this task; we make use of a similar technique as a part of our algorithms in solving the vibration problem.

4. Quantum Algorithms

4.1. QFT and HHL. Quantum Fourier transform (QFT) is the key ingredient of many interesting quantum algorithms, including quantum prime factoring [2], quantum period finding [3], and the Simon’s problem [11]. It is also regarded the quantum version of discrete Fourier transform, which is classically achieved in engineering applications via fast Fourier transform (FFT) [12–14]. Although FFT has runtime \( O(n \log n) \), QFT has the significant advantage of having a runtime \( O(\log n^2) \) quantum gates, consisting of the Hadamard, controlled phase, and SWAP operators.

Quantum eigenvalue estimation, which is based on the QFT, is the main tool in the HHL algorithm. Given a unitary matrix \( U \) embedded in a quantum circuit, the quantum eigenvalue estimation generates the eigendecomposition of an input vector as the output. Since the HHL algorithm has been used as a part of our algorithms, we present a brief outline of the HHL steps.

The objective of the HHL algorithm is to solve a sparse linear system \( Ax = b \), where \( A \) is a Hermitian matrix (the condition of Hermitianity can be relaxed by embedding the matrix and its conjugate transpose in a block matrix). It encodes \( b \) in a quantum state \( |b\rangle \) and generates a quantum state \( |x\rangle \), which is proportional to the solution vector \( x \).

The key idea is the eigendecomposition of a Hermitian matrix. Considering this fact, \( |b\rangle \) can be expressed as a linear combination of the eigenvectors of \( A \):

\[
|b\rangle = \sum_l \beta_l |\xi_l\rangle. \quad (38)
\]

When this vector \( |b\rangle \) is made to input to the quantum eigenvalue estimation, the output is the state; \( \sum \beta_l |\xi_l\rangle |\xi_l\rangle \). The next step of the HHL is to append a qubit \( |0\rangle \) to this state and make \( \sum \beta_l |\xi_l\rangle |\xi_l\rangle |0\rangle \). Then a controlled rotation is applied on \( |\xi_l\rangle \), and the result is

\[
\sum \beta_l |\xi_l\rangle |\xi_l\rangle \left( \sqrt{1 - c^2} |0\rangle + \frac{c}{\xi_l} |1\rangle \right), \quad (39)
\]

where \( c \) is the normalizing constant, which is the ratio of the minimum and maximum eigenvalues of \( A \). The next step is to measure the last qubit in the standard basis, conditioned on seeing \( |1\rangle \). Then the system collapses to the state \( \sum (\beta_l |\xi_l\rangle |\xi_l\rangle |\xi_l\rangle) \).
Since the gates used in the quantum algorithm are unitary operations, it is possible to reverse them, that is, applying the conjugate transposes of the sequence of gates. This procedure is named uncomputing. The quantum eigenvalue estimation is a procedure in which all the steps are defined by unitary gates, making the uncomputing of eigenvalue estimation possible. After applying the relevant conjugate transposes, the system is in the state $\sum_{i}(b_i/|E_i|)|\psi_i\rangle$, which gives the desired outcome $|\chi\rangle$ of the algorithm.

Consider the Halmos dilation $U_S$ of the nonunitary matrix $S$,

$$U_S = \begin{pmatrix} S & \sqrt{I - SS^T} \\ \sqrt{I - S^T S} & -S \end{pmatrix}. \quad (40)$$

We apply this unitary dilation to $\Lambda_{M^{-1}}$, which can be constructed through a mapping, corresponding to the eigenvalues of $M$. Note that $U_{\Lambda_{M^{-1}}}$ is a 2–sparse matrix which can be considered a block matrix with four blocks, in which each block is an $n \times n$ diagonal matrix. Thus, it can be prepared and applied efficiently on a state. It can be easily seen that

$$\begin{pmatrix} F^\dagger & O \\ O & F^\dagger \end{pmatrix} \begin{pmatrix} F & O \\ O & F \end{pmatrix} \begin{pmatrix} f \\ 0 \end{pmatrix} = \begin{pmatrix} F^\dagger \Lambda_{M^{-1}} F f \\ \left( F^\dagger \left(I - \Lambda_{M^{-1}}^\dagger \Lambda_{M^{-1}}\right) F \right)f \end{pmatrix}, \quad (41)$$

where $\begin{pmatrix} f \\ 0 \end{pmatrix}$ is the $2n \times 1$ vector consisting of $n \times 1$ blocks $f$ and zero.

An efficient quantum circuit implementation for the unitary dilation can be done by making a Hermitian matrix $\Lambda_{M^{-1}}$ and its conjugate transpose are embedded in $\begin{pmatrix} O & U_{\Lambda_{M^{-1}}} \\ U_{\Lambda_{M^{-1}}}^\dagger & O \end{pmatrix}$. \quad (42)

It can be easily seen that $H_{\Lambda_{M^{-1}}}$ is a Hermitian 2–sparse matrix. Also it follows that $\|H_{\Lambda_{M^{-1}}}\| = 1$. Therefore, \quad (43)

$$e^{-iH_{\Lambda_{M^{-1}}} \theta} = \cos \theta I - i \sin \theta \Lambda_{M^{-1}}. \quad (43)$$

Letting $\theta = \pi/2$,

$$e^{-i(\pi/2)H_{\Lambda_{M^{-1}}}} = -i \sin \theta \Lambda_{M^{-1}}. \quad (44)$$

Therefore,

$$H_{\Lambda_{M^{-1}}} = i e^{-i(\pi/2)H_{\Lambda_{M^{-1}}}}. \quad (45)$$

Then it is easily verified that, given the input with three registers $|1\rangle|0\rangle|f\rangle$ to $e^{-i(\pi/2)H_{\Lambda_{M^{-1}}}}$, the output would be

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} F^\dagger \Lambda_{M^{-1}} F f \\ \left( F^\dagger \left(I - \Lambda_{M^{-1}}^\dagger \Lambda_{M^{-1}}\right) F \right)f \end{pmatrix}, \quad (46)$$

in which the desired state is derived upon measurement of the second register, conditioned on seeing $|0\rangle$.

The time-varied force applied on the system is an efficiently implementable function in a quantum circuit. This has to be plugged into the system after discretizing the time domain. This can be accomplished through the discretization of the function $e(t)$, encoded in a matrix $E$. Since $e(t)$ is usually sine or cosine, the corresponding operator $E$ is assumed to be unitary and efficient.

4.2. Simple Vibrating Systems with Cyclic Symmetry. Now we present the algorithmic steps in approximating the solution of $u(j, t)$ of the circulant system of ordinary differential equations $Mi = f$. Since a circulant matrix is determined by its eigenvalues, we assume the input which corresponds to the circulant $M$ to be in terms of its eigenvalues. By sparse Hamiltonian simulation, we construct $H_{\Lambda_{M^{-1}}}$ (in (42), (45)) which will be implemented efficiently. The second input is the original forcing vector, acting on the coupled components of the vibrating system, and it is assumed to be encoded in a quantum state. Whenever a qubit is measured in the algorithm, that is performed with respect to the standard basis.

Algorithm 1 requires $\log n$ qubits to encode $|f\rangle$, $O(\log p)$ ancillary qubits to perform the Hadamard transformations and to append at appropriate stages. To solve the linear system through sparse Hamiltonian simulation requires $O(\text{polylog } np)$ qubits. Recall that all the operations used in the algorithm are efficient, scaling down to $O(\text{polylog } np)$. A classical algorithm with analogous steps would require a space of polynomial $np$ to store the information, and also it would run with time complexity $O(\text{polylog } np)$. Therefore, the above algorithm has an exponential saving, compared to a classical algorithm designed for the same task (more on complexity and the selection of HHL for this task will be described in Section 4.4). In particular, if the classical algorithm is based upon approximating the displacement of each sector by finite differences, it would require a linear system solver based method, in which the HHL approach is exponentially advantageous over it. Also the classical fast Fourier transform (FFT) has complexity $n \log n$, while the QFT is again exponentially advantageous over the classical FFT, guaranteeing the exponential saving in our algorithm.

4.3. Damped Elastic Vibrating Systems with Cyclic Symmetry. Given the matrix $\tilde{A}$, Algorithm 2 requires almost the same resource cost as Algorithm 1 and runtime. If the entries in $\tilde{A}$ are assumed to be in the input then it would be the cost for solving the linear system. In particular, it will be polylog in $n$ and polynomial in $t$.

4.4. Achieving Quantum Speed-Up with HHL. It is worthwhile to discuss further the complexity of our algorithms, describing why these solutions are derivable with the same efficiency by neither classical nor other quantum algorithms. First we compare our methods with existing or possible counterpart classical algorithms. Recall the efficiency of HHL is polylog of $n$ for a sparse system with $n$ entries, which is scarcely imaginable in the classical realm, also the efficiency of non-HHL techniques we have used here are not achievable classically. We will explain this further by considering some
Mathematical Problems in Engineering

IN: $|f\rangle$, $H_{A_{M^{-1}}}$, $A = I_n \otimes L_2$

OUT: $u(j, t)$

(i) Append $|f\rangle$ to two qubits in state $|10\rangle$ and prepare the state $|10\rangle |f\rangle$.
(ii) Apply the sequence of operations $I_4 \otimes F_n$, $H_{A_{M^{-1}}}$ and $I_4 \otimes F_n$ respectively to the prepared state, and produce $|0\rangle \otimes (F^T A_{M^{-1}} F) |f\rangle |0\rangle + F^T (I - A_{M^{-1}} A_{M^{-1}}) F |f\rangle |1\rangle$.
(iii) Measure the first two qubits in the standard basis, conditioned on seeing $|00\rangle$. The outcome is proportional to $F^T A_{M^{-1}} F |f\rangle$.
(iv) Applying $H$ to log $p$ qubits in state $|0\rangle$, prepare $H^{\text{log} p} |0\rangle^{\text{log} p}$.
(v) Append $H^{\text{log} p} |0\rangle^{\text{log} p}$ to the outcome $F^T A_{M^{-1}} F |f\rangle$ of the measurement.
(vi) Apply $I_p \otimes E$ to the new state and prepare $|b\rangle$.
(vii) Run the eigenvalue estimation of $e^{|\lambda_i|}$, letting $|b\rangle$ be the input, and get the output $\sum \beta_i |\xi_i\rangle |\nu_i\rangle$.
(viii) Append a qubit $|0\rangle$ to the output to get $\sum \beta_i |\xi_i\rangle |\nu_i\rangle |0\rangle$.
(ix) Perform controlled rotation on $|\xi_i\rangle$.
(x) Measure the last qubit in the standard basis, conditioned on seeing 1. The outcome is proportional to $\sum (\beta_i |\xi_i\rangle |\nu_i\rangle |\nu_i\rangle |0\rangle$.
(xi) Uncompute the eigenvalue estimation to get $\sum (\beta_i |\xi_i\rangle |\nu_i\rangle)$, which is proportional to the state $|x\rangle$ encoding the solution.

Algorithm 1: Simple vibrating system.

IN: $|f\rangle$, $\tilde{A}$

OUT: $u(j, t)$

(i) Applying $H$ to log $p$ qubits in state $|0\rangle$, prepare $H^{\text{log} p} |0\rangle^{\text{log} p}$.
(ii) Apply $F$ to $f$.
(iii) Append $F(|f\rangle)$ to $H^{\text{log} p} |0\rangle^{\text{log} p}$.
(iv) Apply $I_p \otimes E$ to the new state and prepare $|b\rangle$.
(v) Run the eigenvalue estimation of $e^{|\lambda_i|}$, letting $|b\rangle$ be the input, and get the output $\sum \beta_i |\xi_i\rangle |\nu_i\rangle$.
(vi) Append a qubit $|0\rangle$ to the output to get $\sum \beta_i |\xi_i\rangle |\nu_i\rangle |0\rangle$.
(vii) Perform controlled rotation on $|\xi_i\rangle$.
(viii) Measure the last qubit in the standard basis, conditioned on seeing 1. The outcome is proportional to $\sum (\beta_i |\xi_i\rangle |\xi_i\rangle |\nu_i\rangle |\nu_i\rangle$. 
(ix) Uncompute the eigenvalue estimation to get $\sum (\beta_i |\xi_i\rangle |\nu_i\rangle)$, which is proportional to the state $|\bar{x}\rangle$.
(x) Apply the permutation $P$ on the outcome of the previous step.
(xi) Apply $I_p \otimes F_p^T$. This produces a state which is proportional to an $np \times 1$ vector consisting of $p$ blocks, each block has $n$ entries, where the $k$ th element in the $j$ th block gives $u(k, j)$.

Algorithm 2: Damped elastic vibrating system.

concrete, easily realizable examples and standard facts from the circuit-gate framework of quantum computation.

The major techniques we have used here are related to the properties of circulant matrices and unitary-Hermitian embedding of general complex matrices. For instance let us take a cyclic symmetrical system with three components, having masses 1, 2, and 3, respectively. Then the corresponding circulant is

$$ M = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 2 & 3 & 1 \end{pmatrix} $$

It is immediately verified that the circulant in (47) is invertible. Further, it can be seen that the eigenvalues of $M^{-1}$ are $1/6, -0.5 + 0.29i,$ and $-0.5 - 0.29i$. Therefore,

$$ \Lambda_{M^{-1}} = \begin{pmatrix} 1/6 & 0 & 0 \\ 0 & -0.5 + 0.29i & 0 \\ 0 & 0 & -0.5 - 0.29i \end{pmatrix} $$

It is clear that any classical implementation of the matrix in (48) on some classical vector requires a minimum of three registers. Accordingly, any classical implementation of this matrix on a classical state when there are $n$ components in the cyclic symmetrical system cannot be done unless we use at least $n$ registers. However, this can be implemented on a quantum state in complexity $O(\text{polylog} n)$ [16]. Thus, we implement the matrix-vector product in $O(\text{polylog} n)$, which cannot be achieved classically.

The exact matrix-vector product in our method is application of the matrix $\Lambda_{M^{-1}}$ on the already prepared quantum state $F|f\rangle$. Recall the time-varied force which governs the system in most real-world applications is a trigonometric function, in particular either sine or cosine, or a combination of them; the function $e(t)$ can be encoded and efficiently implemented in a quantum circuit. For instance, suppose the forces $Ff_0$ that act on the system represented by (47) are given by
\( E = \begin{pmatrix} \sin t & 0 & 0 \\ 0 & \cos t & 0 \\ 0 & 0 & \sin t \end{pmatrix}, \quad (49) \)

\( f_0 = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}. \)

It is possible to implement the operation \( E | f_0 \rangle \) in \( O(\text{polylog } n) \) using standard quantum circuits in literature [15, 16].

Recall we apply the matrix \( \Lambda P_M^{-1} \) to the already prepared quantum state \( F | f \rangle \) and it will undergo multiplication by \( F^\dagger \) consequently, it is important we discuss on this operation both classically and quantum mechanically. For instance, when we analyze the system given by (47) and (49), the corresponding two Fourier matrices in conjugate pairs are as follows:

\[ F = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 1 & 1 \\ 1 - \frac{1}{2} + \frac{\sqrt{3}}{2} & -\frac{1}{2} - \frac{\sqrt{3}}{2} & -\frac{1}{2} \end{pmatrix}, \quad (50) \]

\[ F^\dagger = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 1 & 1 \\ 1 - \frac{1}{2} - \frac{\sqrt{3}}{2} & -\frac{1}{2} + \frac{\sqrt{3}}{2} & -\frac{1}{2} \end{pmatrix}. \quad (51) \]

These operations are not implementable on a classical computer unless at least \( n \) registers are used. Considering the circulants of order \( n \), we use \( n \times n \) Fourier transform, for which the fastest possible classical methods are based on fast Fourier transforms (FFT) [17, 18]. Thus, the classical complexity boils down eventually at least to linear of \( n \) as the classical bound \( O(n \log n) \) cannot be overcome. Contrasting to this, the \( n \times n \) quantum Fourier transform or the \( QFT \), the quantum counterpart of \( FFT \), can be implemented in a quantum circuit in \( O(\text{polylog } n) \) [19].

The next instrumental non-HHL operation in our algorithms is the implementation of the Hermitian matrix, in which the Halmos dilation of a diagonal matrix is embedded. It is this Hermitian matrix that we claim to implement with an exponential efficiency gain over classical algorithms. Therefore, it is worthwhile to consider a concrete example in order to gain some intuition on its implementation in classical and quantum realms. Let us consider the system given by (47) and (49) again. The Halmos dilation of \( \Lambda P_M^{-1} \) in (48) is given by

\[ U_{\Lambda P_M^{-1}} = \begin{pmatrix} 0.167 & 0 & 0 & 0.986 & 0 & 0 \\ 0 & -0.5 + 0.29i & 0 & 0 & 0.816 & 0 \\ 0 & 0 & -0.5 - 0.29i & 0 & 0 & 0.816 \\ 0.986 & 0 & 0 & -0.167 & 0 & 0 \\ 0 & 0.816 & 0 & 0 & 0.5 - 0.29i & 0 \\ 0 & 0 & 0.816 & 0 & 0 & 0.5 + 0.29i \end{pmatrix}. \quad (52) \]

The Hermitian embedding of the matrix in (52) is given by

\[ H_{\Lambda P_M^{-1}} = \begin{pmatrix} 0.167 & 0 & 0 & 0 & 0 & 0.986 & 0 & 0 \\ 0 & -0.5 - 0.29i & 0 & 0 & 0.816 & 0 & 0 & 0 \\ 0 & 0 & -0.5 - 0.29i & 0 & 0 & 0.816 & 0 & 0 \\ 0.986 & 0 & 0 & -0.167 & 0 & 0 & 0 & 0.5 - 0.29i \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 + 0.29i \\ 0 & 0.816 & 0 & 0 & 0.816 & 0 & 0 & 0 \\ 0 & 0 & 0.816 & 0 & 0 & 0.816 & 0 & 0 \\ 0 & 0 & 0.816 & 0 & 0 & 0.816 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (53) \]
Clearly, any classical implementation of the matrix in (53) upon a classical vector can be done in complexity not less than \( 12 \). Considering an \( n \times n \) system, one may require \( O(\text{poly} n) \) resources and hence the same time complexity. It can be seen from (45) that our Hermitian matrix is expressible as the complex exponent of itself, with some global phase factor which can be ignored. This observation enables us using the **sparse Hamiltonian lemma**, a well-known remarkable result from quantum computation [15]. It guarantees the Hermitian matrix can be implemented on a quantum state in complexity \( O(\text{polylog} n) \). Obviously, this can never be done using a classical computer. Other operations such as tensor products with sparse matrices are standard efficient constructions in quantum computation in the circuit+gate framework.

Once the efficiency of non-HHL methods are discussed, it is worthwhile to look into the direct application of HHL in our methods. Our main application is solving system in (37), in which both matrix and known vector were already proven by us to be constructible in \( O(\text{polylog} n) \) time. What is left is solving a sparse linear system, for which we can use the HHL in \( O(\text{polylog} n) \) time. It is noteworthy that no classical computer can achieve this, as even the **conjugate gradient algorithm** [20]; the fastest classical method known so far for solving linear systems would not yield the efficiency of the HHL, as the latter is exponentially faster when coming to sparse systems.

It is worthwhile considering other available quantum techniques which might provide some insight on analyzing cyclic symmetrical systems. One recent advance is the **quantum principal component analysis** (QPCA) algorithm [21], which shares several common features with HHL, inclusive of being a quantum protocol that takes time polylog of the dimension such that its classical counterpart takes time polynomial in the dimension of the system. Despite the fact that several steps in our algorithms can still be applicable when the HHL is replaced by QPCA, the key ingredient in the latter is efficient exponentiation of a low-rank dense matrix. The interesting characteristic of cyclic symmetrical systems, in particular when considering the case of a simple vibration is the sparse linear system it eventually reduces down to, with the aid of finite differences. Recalling the HHL particularly makes use of sparsity, it suits better for our purpose over QPCA which considers rank of the matrix over sparsity.

Another possible quantum algorithm is **quantum neural networks** (QNN) [22] which has proven to be applicable in several fields [23]. This too has several similarities including the application of eigenvalue estimation as the key ingredient. However, the reduction of a cyclic symmetrical system by central differences to a sparse linear system is still of high significance, which can only be fully appreciated if either HHL or any other exponentially faster linear system solver was applied. Quantum-based learning algorithms such as **quantum reinforcement learning** (QRA) too have provided significantly faster results over classical algorithms [24], which could potentially be helpful in vibration analysis as well. Despite the probabilistic component, the HHL is a direct algorithm built over the circuit-gate framework of quantum computation, for which the complexity is explicitly expressible in terms of the size of the system. On the other hand, our HHL-based approaches have analytically guaranteed speed-up over their classical counterparts. This explains our preference for HHL over quantum learning algorithms.

The latest quantum computing paradigm which has shown much progress in recent years is **quantum annealing**, which is also used in the D-Wave machine, the first-ever commercially available quantum computer [25]. This is a hardware meta-heuristic which has been modelled upon the adiabatic framework of quantum computation, aimed at solving NP-hard problem instances efficiently. As for many other heuristics, the complexity of quantum annealing is still not explored completely, although several bounds have been suggested [26] and several numerical results are available in literature [27]. Indeed, many heuristics including certain learning algorithms were designed with the objective of solving NP problem instances in a reasonable time. It is important to mention that the equation of motion of a cyclic symmetrical system can be classically solved in polynomial time; hence our aim is to make it polylog. The same complexity reduction was achieved by HHL as well as its applications [6–8] which appeared consequently. Therefore, our selection of HHL as the base algorithm in this work is justifiable even in the context of quantum algorithms.

### 5. Discussion

Considering the real-world engineering applications of vibrating systems with cyclic symmetry, large-scale computational models which require significant resource cost and time must be built in order to analyze such systems in a classical setting. We proposed quantum algorithms for this task, which could earn an exponential saving compared to their classical counterparts. Thus developed quantum methods were based on the well-known HHL algorithm for solving linear systems. The speed-up suggested by the HHL-based methods for solving the equations of motion of vibrating systems with cyclic symmetry is hardly imaginable in a classical setting.

In order to make our algorithms more accurate, one might increase the dimension of discretized domain. Further, letting \( p \) very large does not significantly affect the efficiency of the algorithm. However, it will be more advantageous if the optimum \( p \) could be determined, considering the practical needs of accuracy and efficiency.

### Data Availability

No data were used to support this study.

### Conflicts of Interest

The author declares that they have no conflicts of interest.

### Acknowledgments

The author would like to thank Jingbo Wang and Thomas Loke for helpful discussions and comments on an earlier
version of this manuscript. This work was partly supported by University of Colombo research grants.

References


