Shor's Factoring Algorithm and Modular Exponentiation Operators

Robert L. Singleton Jr

Quantum Division, SavantX Research Center, Santa Fe, New Mexico, USA. E-mail: robert.singleton@savantx.com

Editors: ***Chris Fields* & *Danko D. Georgiev***

Article history: Submitted on June 20, 2023; Accepted on August 28, 2023; Published on September 18, 2023.

We provide a pedagogical presentation of Shor's factoring algorithm, which is a quantum algorithm for factoring very large numbers (of order of hundreds to thousands of bits) in polynomial time. In contrast, all known classical algorithms for the factoring problem take an exponential time to factor such large numbers. Shor's algorithm therefore has profound implication for public-key encryption such as RSA and Diffie–Hellman key exchange. We assume no prior knowledge of Shor's algorithm beyond a basic familiarity with the circuit model of quantum computing. Shor's algorithm contains a number of moving parts, and can be rather daunting at first. The literature is replete with derivations and expositions of Shor's algorithm, but most of them seem to be lacking in essential details, and none of them provide a pedagogical presentation. They require a thicket of appendices and assume a knowledge of quantum algorithms and classical mathematics with which the reader might not be familiar. We therefore start with first principle derivations of the quantum Fourier transform (QFT) and quantum phase estimation (QPE), which are the essential building blocks of Shor's algorithm. We then go on to develop the theory of modular exponentiation (ME) operators, one of the fundamental components of Shor's algorithm, and the place where most of the quantum resources are deployed. We also delve into the number theory that establishes the link between factorization and the period of the modular exponential function. We then apply the QPE algorithm to obtain Shor's factoring algorithm. We also discuss the post-quantum processing and the method of continued fractions, which is used to extract the exact period of the modular exponential function from the approximately measured phase angles of the ME operator. The manuscript then moves on to a series of examples. We first verify the formalism by factoring $N = 15$, the smallest number accessible to Shor's algorithm. We then proceed to factor larger integers, developing a systematic procedure that will find the ME operators for any semi-prime $N = p \times q$ (where q and p are prime). Finally, we factor the composite numbers $N = 21, 33, 35, 143, 247$ using the Qiskit simulator. It is observed that the ME operators are somewhat forgiving, and truncated approximate forms are able to extract factors just as well as the exact operators. This is because the method of continued fractions only requires an approximate phase value for its input, which suggests that implementing Shor's algorithm might not be as difficult as first suspected.

Quanta 2023; 12: 41–130.

1. Introduction

In this work, we present a pedagogical construction of Shor's factoring algorithm [1], which can factor exponentially large integers in polynomial time. All known classical algorithms for factorization require an exponential time because they work by brute force, essentially testing all (or most) numbers less than the number being factored. In contrast, Shor's algorithm exploits the massive parallelism inherent in quantum mechanics, so that all numbers can be tested simultaneously rather than sequentially, thereby providing for a polynomial factorization process. Since Shor's algorithm can factor massively large numbers very quickly, it has major implications for the security of encryption standards such as RSA [2] and Diffie–Hellman [3,4] key exchange, rendering these encryption methods severely compromised. As most internet communication is based on such public key encryption schemes, Shor's algorithm has profound implications for digital security.

Shor's algorithm rests upon two fundamental quantum algorithms, the quantum Fourier transform (QFT) and quantum phase estimation (QPE). The QFT, as the name suggests, implements the discrete Fourier transform on a gated quantum computer. Like the classical Fourier transform, it extracts frequency signals from an input source, except that the QFT works by manipulating quantum bits or qubits (two state quantum systems) on a gated quantum computer. The QPE algorithm, in contrast, finds the complex phases or the Eigenvalues of an arbitrary *unitary* linear operator. Shor's algorithm elegantly combines the QFT and QPE to construct a powerful quantum algorithm for factoring very large integers. More precisely, by employing a specific and well chosen unitary operator called the *modular exponentiation* (ME) operator, quantum parallelism allows the QPE to extract the factors of exponentially large numbers in polynomial time. One might say that the QPE algorithm is the workhorse of Shor's algorithm [5], and we shall spend most of our time on the associated ME operators. The mathematics behind Shor's algorithm is based on a simple but profound result from number theory, which maps the factoring problem onto another mathematical problem that finds the period of the *modular exponential function*. The period of this function is directly related to the factors of the number in question, and the QPE extracts this period using the method of continued fractions, thereby providing the sought after factors.

Historically, Shor's algorithm was motivated by Simon's algorithm [6], but we shall not discuss these (interesting) details. We assume no prior knowledge of Shor's algorithm beyond a basic familiarity with the circuit model of quantum computing. Shor's algorithm has a number of moving parts, and it is rather complex. It contains a pre-processing phase that happens on a classical computer, then the QPE is conducted on a quantum computer (using the associated ME operators), and finally there is a post-processing phase that takes place on a classical computer (employing the method of continued fractions). The literature is thick with expositions of Shor's algorithm, but most of them seem to be lacking in some essential respect, and none of them provide a satisfying pedagogical presentation. Consequently, this work is an attempt to derive Shor's algorithm from first principles in a self-contained manner assuming minimal familiarity with the requisite quantum computing machinery. In Sections 2 and 3, we therefore provide complete derivations the QFT and the QPE algorithms, respectively. As we have emphasized, these algorithms form the essential building blocks of Shor's algorithm. For the classical post-processing stage, one must utilize the theory of *continued fractions*, and in Section 4 we provide a brief introduction to the subject, proving a number of fundamental theorems. In Section 5, we are finally ready to address Shor's algorithm, which involves a rigorous construction of the appropriate modular exponentiation operator U. Shor's algorithm then follows by applying the QPE algorithm to this operator. After this, we move on to discuss the post-quantum processing in more detail, where we apply the theory of continued fractions from Section 4 to extract the *exact* period of the modular exponential function $f(x)$ from the *approximately* measured phase of the ME operator U.

In Section 6 we apply the formalism to factor $N = 15$, the smallest number accessible to Shor's algorithm, and we use this section to develop an all purpose factoring script. The difficulty in factoring a number with Shor's algorithm does not lie in the magnitude of the number itself, but in the size of the period r of the modular exponential function $f(x)$ [7]. In Section 7, we apply the formalism to the composites $N = 21, 33, 35, 143, 247$, which have periods ranging from $r = 2$ to $r = 36$. One might think that we have accomplished nothing, since knowing the exact ME operator is equivalent to knowing the period r of the function $f(x)$, and Shor's algorithm would therefore be unnecessary. However, it turns out that we do not require the *exact* ME operators! It is observed that the ME operators are somewhat forgiving, and truncated approximate forms are able to extract factors just as well as the exact operators. This is because the method of continued fractions only requires an approximate phase value for its input, which suggests that implementing Shor's algorithm might not be as difficult as first suspected. Finally, Section 8 provides some conclusions and closing remarks.

2. The Quantum Fourier Transform

2.1. General Definitions

In this section we formulate of the quantum Fourier transform (QFT), where our primary references are Refs. [8] and [9]. Given an M-vector of complex numbers $\psi = (\psi_0, \psi_1, \cdots, \psi_{M-1})$, the *discrete Fourier transform* $\tilde{\psi} = (\tilde{\psi}_0, \tilde{\psi}_1, \cdots, \tilde{\psi}_{M-1})$ is defined by

$$\tilde{\psi}_\ell \;=\; \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} e^{2\pi i\, \ell k/M}\, \psi_k \;, \tag{1}$$

and the *discrete inverse Fourier transform* is therefore given by

$$\psi_k \;=\; \frac{1}{\sqrt{M}} \sum_{\ell=0}^{M-1} e^{-2\pi i\, \ell k/M}\, \tilde{\psi}_\ell \;, \tag{2}$$

where the indices $\ell, k \in \{0, 1, \cdots, M-1\}$. We wish to implement the Fourier transform using an m-qubit quantum system, where $M = 2^m$ is the number of possible quantum states. The corresponding *quantum Fourier transform* will be a linear unitary operator on the m-qubit Hilbert space, denoted by QFT, whose action on the *computational basis* elements reproduces the classical transform (1),

$$QFT\,|\ell\rangle \;=\; \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} e^{2\pi i\, \ell k/M}\, |k\rangle \;. \tag{3}$$

Recall that a linear operator defined only on the basis states is sufficient to give the operator on any state in the Hilbert space. We can now express the QFT operator in a very useful basis-dependent form,

$$QFT = QFT \cdot \mathbb{1} = QFT \cdot \underbrace{\sum_{\ell=0}^{M-1} |\ell\rangle\langle\ell|}_{\mathbb{1}} = \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} \sum_{\ell=0}^{M-1} e^{2\pi i\, k\ell/M}\, |k\rangle\langle\ell| \;, \tag{4}$$

where we have used the decomposition of unity $\mathbb{1} = \sum_{\ell=0}^{M-1} |\ell\rangle\langle\ell|$. Since the quantum Fourier transform is unitary (an easy proof), that is to say $QFT \cdot QFT^\dagger = \mathbb{1}$, then the *inverse quantum Fourier transform* is given by

$$QFT^{-1} = QFT^\dagger = \frac{1}{\sqrt{M}} \sum_{\ell=0}^{M-1} \sum_{k=0}^{M-1} e^{-2\pi i\, k\ell/M}\, |\ell\rangle\langle k| \;. \tag{5}$$

Our aim in this Section is to construct a quantum circuit to implement the QFT operator and its inverse.

2.2. Qubit Ordering and the QFT Circuit

A quantum circuit has four distinct qubit ordering conventions of which we must be cognizant, as they are all present in the literature. We can order the qubits on the quantum circuit in two ways, and we can order the bits of a binary integer in two ways, thereby giving four possible conventions. We will be concerned with two such conventions. In computer science, one represents an m-bit integer k by m binary digits $k_r \in \{0, 1\}$ using the standard notation $k = k_{m-1} \cdots k_1 k_0$. The bit-ordering convention is that k_0 is the lowest-order bit, so that the value of the integer is given by $k = k_0\, 2^0 + k_1\, 2^1 + \cdots + k_{m-1}\, 2^{m-1}$. We must also label the qubits of the quantum circuit, which gives two more possible conventions. In OpenQASM/Qiskit [10], the upper qubit of an m-qubit circuit is labeled by 0, working its way down the circuit and ending with qubit $m-1$ at the bottom. For a general m-bit binary integer $k = k_{m-1} \cdots k_1, k_0$, the least significant bit k_0 is therefore placed at the *top* of the circuit, encoded by the computational basis state $|k_0\rangle$ of the upper qubit. The integers k therefore correspond to the computational basis states $|k\rangle \equiv |k_{m-1} \cdots k_0\rangle \equiv |k_0\rangle \otimes |k_1\rangle \otimes \cdots \otimes |k_{m-2}\rangle \otimes |k_{m-1}\rangle$. Note that the qubit ordering is opposite to the bit-string ordering. This will be one of our primary conventions. Alternatively, we can number the qubits on the circuit from 1 to m, and express binary numbers by m-bit strings of the form $k = k_1 k_2 \cdots k_{m-1} k_m$. In this convention, the lowest order bit k_m

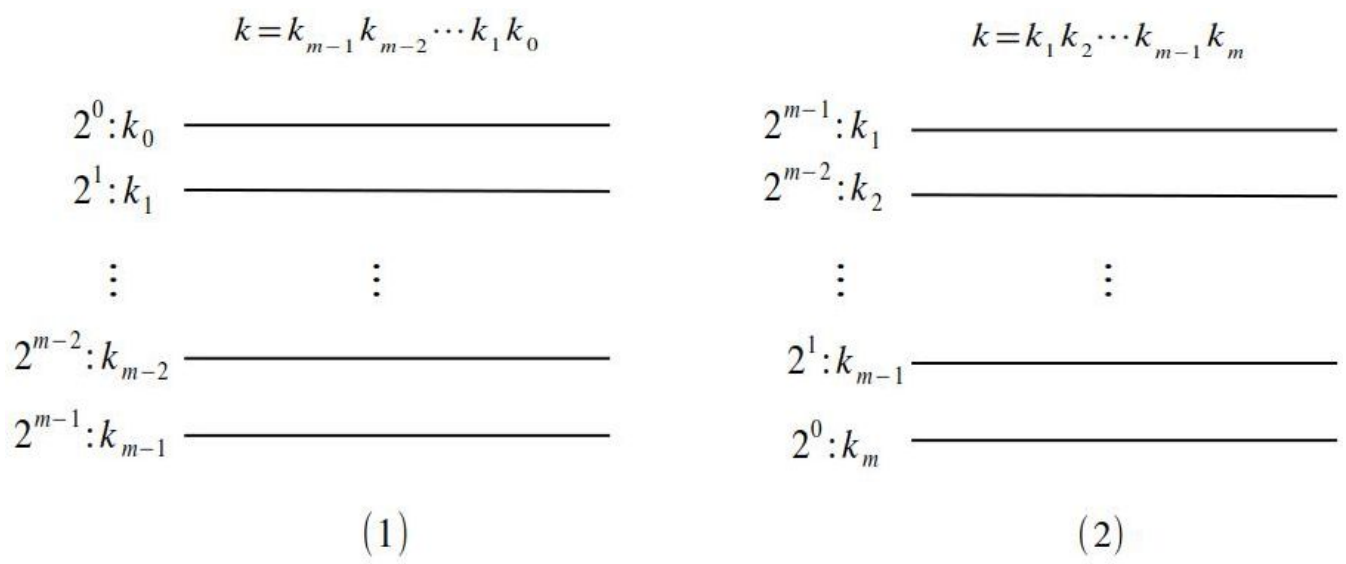


Figure 1: *An m-bit binary integer k can be encoded on a gated quantum computer in either of the conventions described in the text. Convention 1 is called the OpenQASM/Qiskit convention, while Convention 2 is called the Physics/Mathematics convention. Convention 1 labels the top qubit by 0, and works its way down to the last qubit labeled by m − 1. Binary integers are expressed with the standard bit encoding $k = k_{m-1}\cdots k_1 k_0$, which places the lowest order bit k_0 at the top of the circuit. Convention 2 numbers the qubits from 1 to m running from top to bottom, and the bit-ordering of integers is flipped to $k = k_1 k_2 \cdots k_m$, placing the lowest order bit k_m at the bottom of the circuit.*

is placed at the *bottom* of the circuit, and the value of the index integer is $k = k_m\, 2^0 + k_{m-1}\, 2^1 + \cdots + k_1\, 2^{m-1}$. The qubit ordering is the same as the bit-string ordering, and the integers k correspond to the computational basis states $|k\rangle \equiv |k_1 \cdots k_m\rangle \equiv |k_1\rangle \otimes |k_2\rangle \otimes \cdots \otimes |k_{m-1}\rangle \otimes |k_m\rangle$. This is the standard physics and mathematics convention. We shall use both conventions, which are illustrated in Fig. 1. Quantum circuits will be inverted horizontally between these two conventions, so it is important to keep track of which convention is in use.

2.2.1. Convention 2: Standard Physics/Mathematics

We first work through the details of the QFT for Convention 2, the physics and mathematics convention. We consider an m-qubit system in which the qubits are ordered from top to bottom, starting with qubit-1 in the upper position of the circuit and qubit-m at the bottom of the circuit. The quantum system has $M = 2^m$ distinct states that can be indexed by an integer $k \in \{0, 1, \cdots, M-1\}$. We can express this integer in the binary form,

$$\begin{aligned} k &= k_1 k_2 \cdots k_{m-1} k_m \\ &= 2^{m-1} k_1 + 2^{m-2} k_2 + \cdots + 2^1\, k_{m-1} + 2^0\, k_m\,, \end{aligned} \tag{6}$$

where k_m is the least significant bit. The computational basis elements are then defined by

$$\begin{aligned} |k\rangle &= |k_1 k_2 \cdots k_{m-1} k_m\rangle \\ &= |k_1\rangle \otimes |k_2\rangle \otimes \cdots \otimes |k_{m-1}\rangle \otimes |k_m\rangle \;\text{ where }\; k_r \in \{0, 1\}\,. \end{aligned} \tag{7}$$

For example, the state labeled by $k = 1$ is represented by

$$|1\rangle = |0 \cdots 01\rangle = |0\rangle \otimes \cdots \otimes |0\rangle \otimes |1\rangle\,, \tag{8}$$

with $k_m = 1$ and all other bits $k_r = 0$. This state will play a critical role in Shor's algorithm. Note that we are using a slightly ambiguous notation in which $|1\rangle$ is used in different senses on the left- and right-hand sides of equation (8). However, the meaning of the state $|1\rangle$ will be clear from context, so this should cause no problems. Furthermore, this bit convention implies the useful relation

$$\frac{k}{M} = \frac{k_1}{2^1} + \frac{k_2}{2^2} + \cdots + \frac{k_{m-1}}{2^{m-1}} + \frac{k_m}{2^m}\,. \tag{9}$$

Note that a sum over the index $k \in \{0, 1, \cdots, M-1\}$ can be converted into m sums over the binary components $k_r \in \{0, 1\}$ for $r \in \{1, 2, \cdots, m\}$,

$$\sum_{k=0}^{M-1} = \sum_{k_m \in \{0,1\}} \cdots \sum_{k_2 \in \{0,1\}} \sum_{k_1 \in \{0,1\}} . \tag{10}$$

This allows us to express the quantum Fourier transform (3) in the form

$$\begin{aligned} QFT\,|\ell\rangle &\equiv \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} e^{2\pi i\, \ell k/M}\, |k\rangle & (11)\\ &= \frac{1}{2^{m/2}} \sum_{k_m} \cdots \sum_{k_2} \sum_{k_1} e^{2\pi i\, \ell \left(2^{m-1} k_1 + 2^{m-2} k_2 + \cdots + 2^1\, k_{m-1} + 2^0\, k_m\right)/2^m}\, |k_1 k_2 \cdots k_{m-1} k_m\rangle \\ &= \frac{1}{2^{m/2}} \sum_{k_1=0,1} e^{2\pi i\, \ell k_1/2^1} |k_1\rangle \otimes \sum_{k_2=0,1} e^{2\pi i\, \ell k_2/2^2} |k_2\rangle \otimes \cdots \otimes \sum_{k_m=0,1} e^{2\pi i\, \ell k_m/2^m} |k_m\rangle \\ &= \frac{1}{2^{m/2}} \left(|0\rangle + e^{2\pi i\, \ell/2^1}\, |1\rangle\right)_1 \otimes \left(|0\rangle + e^{2\pi i\, \ell/2^2}\, |1\rangle\right)_2 \otimes \cdots \otimes \left(|0\rangle + e^{2\pi i\, \ell/2^{m-1}}\, |1\rangle\right)_{m-1} \otimes \left(|0\rangle + e^{2\pi i\, \ell/2^m}\, |1\rangle\right)_m . & (12) \end{aligned}$$

In our current convention, the quantum state indexed by the integer ℓ is given by

$$\begin{aligned} |\ell\rangle &= |\ell_1 \ell_2 \cdots \ell_{m-1} \ell_m\rangle \quad \text{with} \quad \ell_r \in \{0, 1\} \\ &= |\ell_1\rangle \otimes |\ell_2\rangle \otimes \cdots \otimes |\ell_{m-1}\rangle \otimes |\ell_m\rangle\,, \end{aligned} \tag{13}$$

where ℓ takes the binary form

$$\ell = \ell_1 \ell_2 \cdots \ell_{m-1} \ell_m = 2^{m-1} \ell_1 + 2^{m-2} \ell_2 + \cdots + 2^1\, \ell_{m-1} + 2^0\, \ell_m\,. \tag{14}$$

We shall also introduce the notion of *binary fractions* corresponding to non-negative m-bit phase angles,

$$\Omega \equiv 0.\ell_1 \ell_2 \cdots \ell_{m-1} \ell_m \equiv \frac{\ell_1}{2^1} + \frac{\ell_2}{2^2} + \cdots + \frac{\ell_{m-1}}{2^{m-1}} + \frac{\ell_m}{2^m}\,. \tag{15}$$

Note that the phase $\Omega = 0.\ell_1 \ell_2 \cdots \ell_m$ and the corresponding integer index $\ell = \ell_1 \ell_2 \cdots \ell_m$ are related by

$$\ell = M\Omega = 2^m\, \Omega\,, \tag{16}$$

an expression we shall employ throughout the sequel. We now rewrite the exponential terms in (12) as follows, working slowly through the algebra, starting with qubit-1:

$$\begin{aligned} 2\pi i\, \frac{\ell}{2^1} &= \frac{2\pi i}{2^1}\left[\left(2^{m-1} \ell_1 + 2^{m-2} \ell_2 + \cdots + 2^1\, \ell_{m-1}\right) + 2^0\, \ell_m\right] \\ &= 2\pi i\left[\left(2^{m-2} \ell_1 + 2^{m-3} \ell_2 + \cdots + 2^0\, \ell_{m-1}\right) + \Omega_1\right] & (17)\\ 2\pi i\, \frac{\ell}{2^2} &= \frac{2\pi i}{2^2}\left[\left(2^{m-1} \ell_1 + 2^{m-2} \ell_2 + \cdots 2^2 \ell_{m-2}\right) + 2^1\, \ell_{m-1} + 2^0\, \ell_m\right] \\ &= 2\pi i\left[\left(2^{m-3} \ell_1 + 2^{m-4} \ell_2 + \cdots 2^0 \ell_{m-2}\right) + \Omega_2\right] & (18)\\ &\cdots \\ 2\pi i\, \frac{\ell}{2^r} &= \frac{2\pi i}{2^r}\left[\left(2^{m-1} \ell_1 + 2^{m-2} \ell_2 + \cdots + 2^r\, \ell_{m-r}\right) + 2^{r-1}\, \ell_{m-r+1} + \cdots + 2^1\, \ell_{m-1} + 2^0\, \ell_m\right] \\ &= 2\pi i\left[\left(2^{m-r-1} \ell_1 + 2^{m-r-2} \ell_2 + \cdots + 2^0\, \ell_{m-r}\right) + \Omega_r\right] & (19)\\ &\cdots \\ 2\pi i\, \frac{\ell}{2^{m-1}} &= \frac{2\pi i}{2^{m-1}}\left[(2^{m-1} \ell_1) + 2^{m-2} \ell_2 + \cdots + 2^1\, \ell_{m-1} + 2^0\, \ell_m\right] \\ &= 2\pi i\left[(\ell_1) + \Omega_{m-1}\right] & (20) \end{aligned}$$

$$
\begin{aligned}
2\pi i\,\frac{\ell}{2^m} &= \frac{2\pi i}{2^m}\Big[2^{m-1}\ell_1 + 2^{m-2}\ell_2 + \cdots + 2^1\,\ell_{m-1} + 2^0\,\ell_m\Big] \\
&= 2\pi i\Big[\Omega_m\Big]\,, \qquad (21)
\end{aligned}
$$

where the *partial-phase angles* are defined by

$$
\begin{aligned}
\Omega_1 &\equiv \frac{\ell_m}{2^1} = 0.\ell_m \qquad (22)\\
\Omega_2 &\equiv \frac{\ell_{m-1}}{2^1} + \frac{\ell_m}{2^2} = 0.\ell_{m-1}\ell_m \qquad (23)\\
&\cdots \\
\Omega_r &\equiv \frac{\ell_{m-r+1}}{2^1} + \cdots + \frac{\ell_{m-1}}{2^{r-1}} + \frac{\ell_m}{2^r} = 0.\ell_{m-r+1}\cdots\ell_{m-1}\ell_m \qquad (24)\\
&\cdots \\
\Omega_{m-1} &\equiv \frac{\ell_2}{2^1} + \frac{\ell_3}{2^2} + \cdots + \frac{\ell_{m-2}}{2^{m-1}} + \frac{\ell_m}{2^{m-1}} = 0.\ell_2\cdots\ell_{m-1}\ell_m \qquad (25)\\
\Omega_m &\equiv \frac{\ell_1}{2^1} + \frac{\ell_2}{2^2} + \cdots + \frac{\ell_{m-1}}{2^{m-1}} + \frac{\ell_m}{2^m} = 0.\ell_1\cdots\ell_{m-1}\ell_m\,. \qquad (26)
\end{aligned}
$$

Note that integer multiples of $2\pi i$ in the parentheses of equations (17)–(21) do not contribute, as $e^{2\pi i\,n} = 1$ for any integer n, thereby permitting us to express the QFT operation only in terms of the partial phases

$$
\Omega_r = \sum_{k=1}^{r} \frac{\ell_{m-r+k}}{2^k} \quad \text{for } r \in \{1, 2, \cdots, m\}\,. \qquad (27)
$$

The quantum Fourier transform (12) of the ℓ-state can therefore be written in any one of three useful forms:

$$
\begin{aligned}
QFT\,|\ell\rangle &= \frac{1}{2^{m/2}}\Big(|0\rangle + e^{2\pi i\,\ell/2^1}\,|1\rangle\Big)_1 \otimes \Big(|0\rangle + e^{2\pi i\,\ell/2^2}\,|1\rangle\Big)_2 \otimes \cdots \otimes \qquad (28)\\
&\qquad \Big(|0\rangle + e^{2\pi i\,\ell/2^{m-1}}\,|1\rangle\Big)_{m-1} \otimes \Big(|0\rangle + e^{2\pi i\,\ell/2^m}\,|1\rangle\Big)_m \\
&= \frac{1}{2^{m/2}}\Big(|0\rangle + e^{2\pi i\,0.\ell_m}\,|1\rangle\Big)_1 \otimes \Big(|0\rangle + e^{2\pi i\,0.\ell_{m-1}\ell_m}\,|1\rangle\Big)_2 \otimes \cdots \otimes \qquad (29)\\
&\qquad \Big(|0\rangle + e^{2\pi i\,0.\ell_2\cdots\ell_{m-1}\ell_m}\,|1\rangle\Big)_{m-1} \otimes \Big(|0\rangle + e^{2\pi i\,0.\ell_1\ell_2\cdots\ell_{m-1}\ell_m}\,|1\rangle\Big)_m \\
&= \frac{1}{2^{m/2}}\Big(|0\rangle + e^{2\pi i\,\Omega_1}\,|1\rangle\Big)_1 \otimes \Big(|0\rangle + e^{2\pi i\,\Omega_2}\,|1\rangle\Big)_2 \otimes \cdots \otimes \\
&\qquad \Big(|0\rangle + e^{2\pi i\,\Omega_{m-1}}\,|1\rangle\Big)_{m-1} \otimes \Big(|0\rangle + e^{2\pi i\,\Omega_m}\,|1\rangle\Big)_m\,. \qquad (30)
\end{aligned}
$$

We next reverse the order of the qubits with a string of SWAP gates to form the state

$$
\begin{aligned}
|\psi_{\rm rev}\rangle &= \frac{1}{2^{m/2}}\Big(|0\rangle + e^{2\pi i\,\Omega_m}\,|1\rangle\Big)_1 \otimes \Big(|0\rangle + e^{2\pi i\,\Omega_{m-1}}\,|1\rangle\Big)_2 \otimes \cdots \otimes \qquad (31)\\
&\qquad \Big(|0\rangle + e^{2\pi i\,\Omega_2}\,|1\rangle\Big)_{m-1} \otimes \Big(|0\rangle + e^{2\pi i\,\Omega_1}\,|1\rangle\Big)_m\,.
\end{aligned}
$$

The state $|\psi_{\rm rev}\rangle$ can be represented quite readily by a quantum circuit. To see this, let us start with qubit-m of (31). Since $e^{2\pi i\,\Omega_1} = e^{2\pi i\,\ell_m/2} = (-1)^{\ell_m}$, and since ℓ_m takes the binary values 0 and 1, we have

$$
\left.
\begin{aligned}
\frac{1}{\sqrt{2}}\Big(|0\rangle + e^{2\pi i\,\Omega_1(\ell_m=0)}\,|1\rangle\Big)_m &= \frac{|0\rangle_m + |1\rangle_m}{\sqrt{2}} = H|0\rangle_m \\
\frac{1}{\sqrt{2}}\Big(|0\rangle + e^{2\pi i\,\Omega_1(\ell_m=1)}\,|1\rangle\Big)_m &= \frac{|0\rangle_m - |1\rangle_m}{\sqrt{2}} = H|1\rangle_m
\end{aligned}
\right\} = H|\ell_m\rangle \qquad (32)
$$

for $\ell_m \in \{0,1\}$, where the single-qubit Hadamard gate is defined by

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \text{with} \quad |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} . \tag{33}$$

We will often place subscripts on the single-qubit basis states such as $|\ell_m\rangle_m$ to explicitly indicate their qubit position in the quantum circuit. We will also denote the qubit upon which the Hadamard gate acts by a superscript, *e.g.* H^m explicitly states that H acts on the m-th qubit. Therefore, we can express (32) in the form

$$\frac{1}{\sqrt{2}}\Big(|0\rangle + e^{2\pi i\,\Omega_1}\,|1\rangle\Big)_m \;=\; H^m|\ell_m\rangle_m \;\text{ with }\; \ell_m \in \{0,1\}\,. \tag{34}$$

Moving on to the next qubit, $m-1$, we find

$$\frac{1}{\sqrt{2}}\Big(|0\rangle + e^{2\pi i\,\Omega_2}\,|1\rangle\Big)_{m-1} \;=\; \frac{1}{\sqrt{2}}\Big(|0\rangle + e^{2\pi i\,\ell_{m-1}/2^1}\cdot e^{2\pi i\,\ell_m/2^2}\,|1\rangle\Big)_{m-1}\,. \tag{35}$$

The first exponential $e^{2\pi i\,\ell_{m-1}/2} = e^{\pi i\,\ell_{m-1}} = (-1)^{\ell_{m-1}}$ gives a Hadamard gate acting on $|\ell_{m-1}\rangle_{m-1}$, and the second exponential $e^{2\pi i\,\ell_m/2^2}$ produces a controlled phase gate with angle $\theta = 2\pi/2^2$ (call it CR_2) acting on the target qubit $|\ell_{m-1}\rangle_{m-1}$ with the control qubit $|\ell_m\rangle_m$, so that

$$\frac{1}{\sqrt{2}}\Big(|0\rangle + e^{2\pi i\,\Omega_2}\,|1\rangle\Big)_{m-1} \;=\; H^{m-1}\cdot C^m R_2^{m-1}\,|\ell_{m-1}\rangle_{m-1} \;\text{ where }\; \ell_{m-1} \in \{0,1\}\,. \tag{36}$$

We have used superscripts on the controlled-R gate, writing $C^c R_2^t$ to explicitly indicate the control qubit c and the target qubit t. Finally, let us examine the 1-st qubit, where we find

$$\frac{1}{\sqrt{2}}\Big(|0\rangle + e^{2\pi i\,\Omega_m}\,|1\rangle\Big)_1 \;=\; \frac{1}{\sqrt{2}}\Big(|0\rangle + e^{2\pi i\,\ell_1/2}\cdot e^{2\pi i\,\ell_2/2^2}\cdots e^{2\pi i\,\ell_{m-1}/2^{m-1}}\cdot e^{2\pi i\,\ell_m/2^m}\,|1\rangle\Big)_1 \tag{37}$$

$$\;=\; H^1\cdot C^2R_2^1\cdots C^{m-1}R_{m-1}^1\cdot C^mR_m^1\,|\ell_1\rangle_1 \quad \text{with} \quad \ell_1 \in \{0,1\}\,, \tag{38}$$

where the single-qubit phase operator is defined by

$$R_n \;=\; \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^n} \end{bmatrix}\,. \tag{39}$$

In terms of the standard phase gate $P(\theta)$, we can express the phase rotation by

$$R_n = P(\theta_n) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta_n} \end{bmatrix} \quad \text{where} \quad \theta_n = \frac{2\pi}{2^n} = \frac{\pi}{2^{n-1}}\,. \tag{40}$$

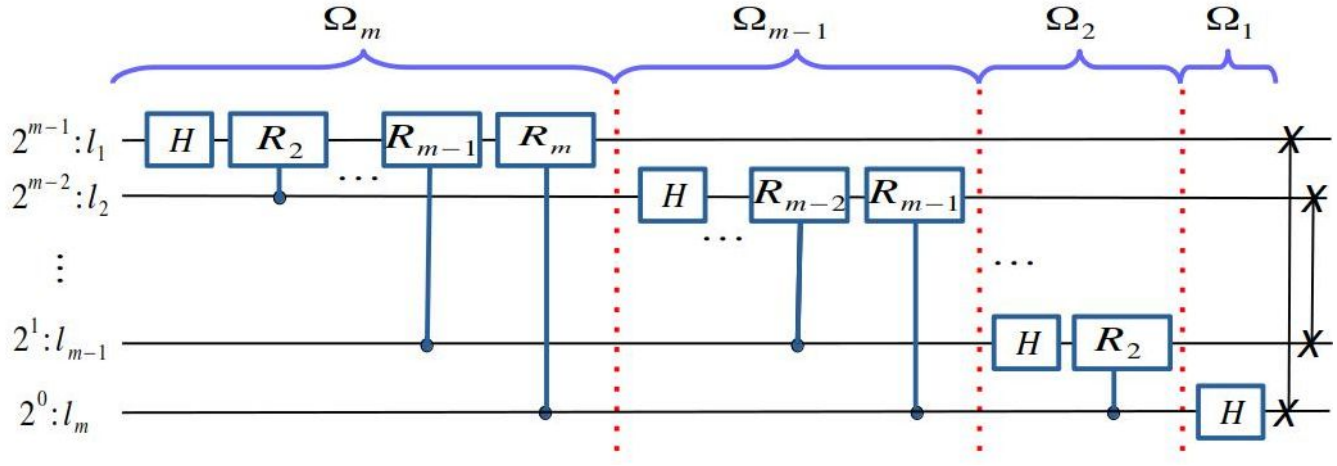


Figure 2: *Convention 2 for the QFT: The standard physics/mathematics convention.*

Figure 2 reproduces the corresponding QFT circuit. Note that SWAP gates are required at the end of the circuit to place the states back into their original order. The inverse $QFT^\dagger$ is given by reading the circuit backwards from right to left, starting with the SWAP gates, and inverting all phase angles (replacing R_n by $R_n^\dagger$). The circuit uses $1 + 2 + \cdots + m = \frac{1}{2}m(m+1) = O(m^2)$ distinct gates, plus $O(m/2)$ SWAP gates.

What would have happened if we had not used the SWAP gates to reverse the qubit order in state (31), but instead appealed directly to (30)? We would have found terms like $|\ell_m\rangle_1$, and the index of ℓ would not have paired properly with the associated qubit. By performing the SWAP operations, we only find states of the form $|\ell_k\rangle_k$. It should therefore cause no confusion if we henceforth drop the subscript on the basis states and simply write $|\ell_k\rangle$.

2.2.2. Convention 1: OpenQASM/Qiskit

We now look at the convention used by OpenQASM/Qiskit. Consider an m qubit system in which the circuit for the computational basis states start with qubit 0 in the upper position and qubit $m-1$ in the lower position. There are $M = 2^m$ quantum states in the system, and the index integer $k \in \{0, 1, \cdots, M-1\}$ can be expressed in a binary form where k_0 is the least significant bit,

$$\begin{aligned} k &= k_{m-1}k_{m-2}\cdots k_1 k_0 \\ &= 2^{m-1}k_{m-1} + 2^{m-2}k_{m-2} + \cdots + 2^1 k_1 + 2^0 k_0 \,. \end{aligned} \tag{41}$$

The integer k can then be used to label the computational basis states,

$$\begin{aligned} |k\rangle &= |k_{m-1}k_{m-2}\cdots k_1 k_0\rangle \\ &= |k_0\rangle \otimes |k_1\rangle \otimes \cdots \otimes |k_{m-2}\rangle \otimes |k_{m-1}\rangle \,. \end{aligned} \tag{42}$$

In this convention, the state labeled by $k = 1$ is given by

$$|1\rangle = |0\cdots 01\rangle = |1\rangle \otimes |0\rangle \otimes \cdots \otimes |0\rangle \,, \tag{43}$$

with $k_0 = 1$ and all other bits $k_r = 0$. As previously mentioned, this state will play a critical role in Shor's algorithm. We also record here the convenient relation

$$\frac{k}{M} = \frac{k_{m-1}}{2^1} + \frac{k_{m-2}}{2^2} + \cdots + \frac{k_1}{2^{m-1}} + \frac{k_0}{2^m} \,. \tag{44}$$

As before, we can replace a sum over the index integer k by m sums over the binary components $k_r \in \{0, 1\}$ for $r \in \{0, 1, \cdots, m-1\}$,

$$\sum_{k=0}^{M-1} = \sum_{k_{m-1}\in\{0,1\}} \cdots \sum_{k_1\in\{0,1\}} \sum_{k_0\in\{0,1\}} \,, \tag{45}$$

so that expression (3) for the quantum Fourier transform becomes

$$QFT\,|\ell\rangle \equiv \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} e^{2\pi i\,\ell k/M}\,|k\rangle \tag{46}$$

$$\begin{aligned} &= \frac{1}{2^{m/2}} \sum_{k_{m-1}} \cdots \sum_{k_1} \sum_{k_0} e^{2\pi i\,\ell\left(2^0 k_0 + 2^1 k_1 + \cdots + 2^{m-2}k_{m-2} + 2^{m-1}k_{m-1}\right)/2^m}\,|k_{m-1}k_{m-2}\cdots k_1 k_0\rangle \\ &= \frac{1}{2^{m/2}} \sum_{k_0=0,1} e^{2\pi i\,\ell k_0/2^m}|k_0\rangle \otimes \sum_{k_1=0,1} e^{2\pi i\,\ell k_1/2^{m-1}}|k_1\rangle \otimes \cdots \otimes \sum_{k_{m-1}=0,1} e^{2\pi i\,\ell k_{m-1}/2^1}|k_{m-1}\rangle \\ &= \frac{1}{2^{m/2}}\left(|0\rangle + e^{2\pi i\,\ell/2^m}\,|1\rangle\right)_0 \otimes \left(|0\rangle + e^{2\pi i\,\ell/2^{m-1}}\,|1\rangle\right)_1 \otimes \cdots \otimes \\ &\qquad \left(|0\rangle + e^{2\pi i\,\ell/2^2}\,|1\rangle\right)_{m-2} \otimes \left(|0\rangle + e^{2\pi i\,\ell/2^1}\,|1\rangle\right)_{m-1} \,. \end{aligned} \tag{47}$$

Next we use the form of the ℓ-state given by our OpenQASM convention,

$$\begin{aligned} |\ell\rangle &= |\ell_{m-1}\ell_{m-2}\cdots\ell_1\ell_0\,\rangle \quad \text{with} \quad \ell_r \in \{0,1\} \\ &= |\ell_0\rangle \otimes |\ell_1\rangle \otimes \cdots \otimes |\ell_{m-2}\rangle \otimes |\ell_{m-1}\rangle\,, \end{aligned} \tag{48}$$

where the m-bit integer ℓ takes the binary form

$$\begin{aligned} \ell &= \ell_{m-1}\ell_{m-2}\cdots\ell_1\ell_0 \\ &= 2^{m-1}\ell_{m-1} + 2^{m-2}\ell_{m-2} + \cdots + 2^1\,\ell_1 + 2^0\,\ell_0\,, \end{aligned} \tag{49}$$

while m-bit binary fractions can be expressed by

$$\Omega = 0.\ell_{m-1}\ell_{m-2}\cdots\ell_1\ell_0 \equiv \frac{\ell_{m-1}}{2^1} + \frac{\ell_{m-2}}{2^2} + \cdots + \frac{\ell_1}{2^{m-1}} + \frac{\ell_0}{2^m}\,. \tag{50}$$

Similarly to the previous case, the exponential terms in (47) can be written

$$\begin{aligned} 2\pi i\,\frac{\ell}{2^m} &= \frac{2\pi i}{2^m}\Big[2^{m-1}\ell_{m-1} + 2^{m-2}\ell_{m-2} + \cdots + 2^1\,\ell_1 + 2^0\,\ell_0\Big] \\ &= 2\pi i\Big[\Omega_0\Big] \end{aligned} \tag{51}$$

$$\begin{aligned} 2\pi i\,\frac{\ell}{2^{m-1}} &= \frac{2\pi i}{2^{m-1}}\Big[(2^{m-1}\ell_{m-1}) + 2^{m-2}\ell_{m-2} + \cdots + 2^1\,\ell_1 + 2^0\,\ell_0\Big] \\ &= 2\pi i\Big[(\ell_{m-1}) + \Omega_1\Big] \end{aligned} \tag{52}$$

$$\cdots$$

$$\begin{aligned} 2\pi i\,\frac{\ell}{2^{m-r}} &= \frac{2\pi i}{2^{m-r}}\Big[\Big(2^{m-1}\ell_{m-1} + 2^{m-2}\ell_{m-2} + \cdots + 2^{m-r}\,\ell_{m-r}\Big) + 2^{m-r-1}\,\ell_{m-r-1} \\ &\qquad\qquad + \cdots + 2^1\,\ell_1 + 2^0\,\ell_0\Big] \\ &= 2\pi i\Big[\Big(2^{r-1}\ell_{m-1} + 2^{r-2}\ell_{m-2} + \cdots + 2^0\,\ell_{m-r}\Big) + \Omega_r\Big] \end{aligned} \tag{53}$$

$$\cdots$$

$$\begin{aligned} 2\pi i\,\frac{\ell}{2^2} &= \frac{2\pi i}{2^2}\Big[\Big(2^{m-1}\ell_{m-1} + 2^{m-2}\ell_{m-2} + \cdots 2^2\ell_2\Big) + 2^1\,\ell_1 + 2^0\,\ell_0\Big] \\ &= 2\pi i\Big[\Big(2^{m-3}\ell_{m-1} + 2^{m-4}\ell_{m-2} + \cdots 2^0\ell_2\Big) + \Omega_{m-2}\Big] \end{aligned} \tag{54}$$

$$\begin{aligned} 2\pi i\,\frac{\ell}{2^1} &= \frac{2\pi i}{2^1}\Big[\Big(2^{m-1}\ell_{m-1} + 2^{m-2}\ell_{m-2} + \cdots + 2^1\,\ell_1\Big) + 2^0\,\ell_0\Big] \\ &= 2\pi i\Big[\Big(2^{m-2}\ell_{m-1} + 2^{m-3}\ell_{m-2} + \cdots + 2^0\,\ell_1\Big) + \Omega_{m-1}\Big]\,, \end{aligned} \tag{55}$$

where the partial phases are now defined by

$$\Omega_0 \equiv \frac{\ell_{m-1}}{2^1} + \frac{\ell_{m-2}}{2^2} + \cdots + \frac{\ell_1}{2^{m-1}} + \frac{\ell_0}{2^m} = 0.\ell_{m-1}\cdots\ell_1\ell_0 \tag{56}$$

$$\Omega_1 \equiv \frac{\ell_{m-2}}{2^1} + \frac{\ell_{m-3}}{2^2} + \cdots + \frac{\ell_1}{2^{m-2}} + \frac{\ell_0}{2^{m-1}} = 0.\ell_{m-2}\cdots\ell_1\ell_0 \tag{57}$$

$$\cdots$$

$$\Omega_r \equiv \frac{\ell_{m-r-1}}{2^1} + \frac{\ell_{m-r-2}}{2^2} + \cdots + \frac{\ell_1}{2^{m-r-1}} + \frac{\ell_0}{2^{m-r}} = 0.\ell_{m-r-1}\cdots\ell_1\ell_0 \tag{58}$$

$$\cdots$$

$$\Omega_{m-2} \equiv \frac{\ell_1}{2^1} + \frac{\ell_0}{2^2} = 0.\ell_1\ell_0 \tag{59}$$

$$\Omega_{m-1} \equiv \frac{\ell_0}{2^1} = 0.\ell_0\,. \tag{60}$$

The general phase takes the form

$$\Omega_r = \sum_{k=1}^{m-r} \frac{\ell_{m-r-k}}{2^k} \quad \text{for } r \in \{0, 1, \cdots, m-1\}\,, \tag{61}$$

and the quantum Fourier transform (47) can now be expressed in one of three equivalent ways:

$$QFT\,|\ell\rangle = \frac{1}{2^{m/2}}\Big(|0\rangle + e^{2\pi i\,\ell/2^m}\,|1\rangle\Big)_0 \otimes \Big(|0\rangle + e^{2\pi i\,\ell/2^{m-1}}\,|1\rangle\Big)_1 \otimes \cdots \otimes \Big(|0\rangle + e^{2\pi i\,\ell/2^2}\,|1\rangle\Big)_{m-2} \otimes \Big(|0\rangle + e^{2\pi i\,\ell/2^1}\,|1\rangle\Big)_{m-1} \tag{62}$$

$$= \frac{1}{2^{m/2}}\Big(|0\rangle + e^{2\pi i\,0.\ell_0}\,|1\rangle\Big)_0 \otimes \Big(|0\rangle + e^{2\pi i\,0.\ell_1\ell_0}\,|1\rangle\Big)_1 \otimes \cdots \otimes \Big(|0\rangle + e^{2\pi i\,0.\ell_{m-2}\cdots\ell_1\ell_0}\,|1\rangle\Big)_{m-2} \otimes \Big(|0\rangle + e^{2\pi i\,0.\ell_{m-1}\cdots\ell_1\ell_0}\,|1\rangle\Big)_{m-1} \tag{63}$$

$$= \frac{1}{2^{m/2}}\Big(|0\rangle + e^{2\pi i\,\Omega_0}\,|1\rangle\Big)_0 \otimes \Big(|0\rangle + e^{2\pi i\,\Omega_1}\,|1\rangle\Big)_1 \otimes \cdots \otimes \Big(|0\rangle + e^{2\pi i\,\Omega_{m-2}}\,|1\rangle\Big)_{m-2} \otimes \Big(|0\rangle + e^{2\pi i\,\Omega_{m-1}}\,|1\rangle\Big)_{m-1}\,. \tag{64}$$

As before, we invert the qubits with SWAP gates to form the state

$$|\psi_{\text{rev}}\rangle = \frac{1}{2^{m/2}}\Big(|0\rangle + e^{2\pi i\,\Omega_{m-1}}\,|1\rangle\Big)_0 \otimes \Big(|0\rangle + e^{2\pi i\,\Omega_{m-2}}\,|1\rangle\Big)_1 \otimes \cdots \otimes \Big(|0\rangle + e^{2\pi i\,\Omega_1}\,|1\rangle\Big)_{m-2} \otimes \Big(|0\rangle + e^{2\pi i\,\Omega_0}\,|1\rangle\Big)_{m-1}\,, \tag{65}$$

which can be expressed in terms of basic gates to give the following circuit. Starting with qubit-0 of (65), and using $e^{2\pi i\,\Omega_{m-1}} = e^{2\pi i\,\ell_0/2} = (-1)^{\ell_0}$, we find a state similar to the previous case,

$$\frac{1}{\sqrt{2}}\Big(|0\rangle + e^{2\pi i\,\Omega_{m-1}}\,|1\rangle\Big)_0 = H^0|\ell_0\rangle \quad \text{where } \ell_0 \in \{0, 1\}\,. \tag{66}$$

The other qubits give corresponding results, so we move on to the bottom of the circuit, expressing qubit $m-1$ by

$$\frac{1}{\sqrt{2}}\Big(|0\rangle + e^{2\pi i\,\Omega_0}\,|1\rangle\Big)_{m-1} = \frac{1}{\sqrt{2}}\Big(|0\rangle + e^{2\pi i\,\ell_{m-1}/2}\cdot e^{2\pi i\,\ell_{m-2}/2^2}\cdots e^{2\pi i\,\ell_1/2^{m-1}}\cdot e^{2\pi i\,\ell_0/2^m}\,|1\rangle\Big)_{m-1}$$
$$= H^{m-1}\cdot C^{m-2}R_2^{m-1}\cdots C^1R_{m-1}^{m-1}\cdot C^0R_m^{m-1}\,|\ell_{m-1}\rangle \tag{67}$$

with $\ell_{m-1} \in \{0, 1\}$, where the phase operators are defined by

$$R_n = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^n} \end{bmatrix} = P(\pi/2^{n-1})\,. \tag{68}$$

Figure 3 reproduces the commensurate QFT circuit in the OpenQASM/Qiskit convention. The inverse $QFT^\dagger$ is given by reading the circuit in reverse order from right to left, starting with the SWAP gates, and inverting the sign of the phase gates. Note that the QFT circuits are reversed between the two conventions.

3. Quantum Phase Estimation

We now turn to quantum phase estimation (QPE), which is the workhorse of Shor's algorithm. Our primary references for this section are Refs. [8] and [9]. We shall work through the calculations in both Conventions 1 and 2 as outlined in Section 2.2, corresponding to the OpenQASM and the standard physics conventions, respectively.

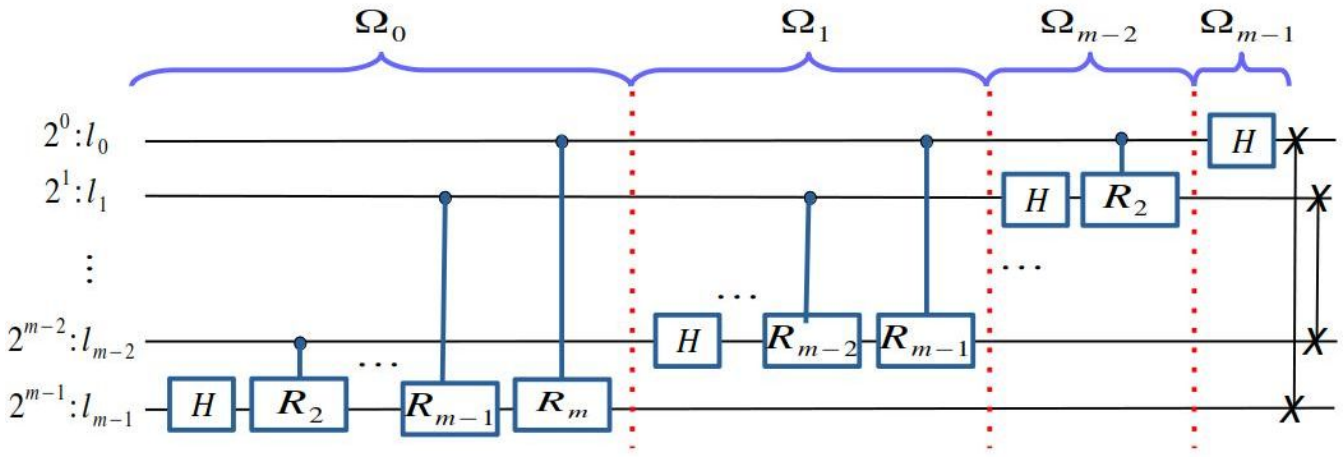


Figure 3: *Convention 1 for the QFT: OpenQASM/Qiskit.*

3.1. Convention 2: Standard Physics/Mathematics

We first examine Convention 2, the traditional physics and mathematics convention. Consider a linear unitary operator U with Eigenvalue $e^{2\pi i\theta}$ and Eigenstate $|u\rangle$, where θ is a real phase such that $0 \leq \theta < 1$:

$$U|u\rangle = e^{2\pi i\theta}|u\rangle \, . \tag{69}$$

Since U is unitary, its Eigenvalues have a norm of unity. We wish to build a QPE circuit that will output an approximate (m-bit binary) value for the phase angle θ. The circuit will consist of a *front-end* and a *back-end*. There are two registers in the QPE front-end circuit: (i) a control register consisting of m qubits and (ii) a work register containing n qubits. We store the Eigenstate $|u\rangle$ in the work register. To construct the front-end circuit, we first apply Hadamard gates to every control qubit, forming the state

$$|\psi_1\rangle = H|0\rangle \otimes \cdots \otimes H|0\rangle \otimes |u\rangle = \frac{1}{\sqrt{M}} \sum_{k-0}^{M-1} |k\rangle \otimes |u\rangle \, , \tag{70}$$

where $M = 2^m$ is the total number of computational basis states. Given the unitary operator U, we assume that we are able to build a family of m controlled-U^p gates for $p \in \{2^0, 2^1, \cdots, 2^{m-1}\}$ that operate on the work register containing the state $|u\rangle$. Note that the action of a CU^p operator for a single control qubit takes the form

$$CU^pH|0\rangle \otimes |u\rangle = \frac{1}{\sqrt{2}} CU^p \Big(|0\rangle + |1\rangle\Big) \otimes |u\rangle = \frac{1}{\sqrt{2}}|0\rangle \otimes |u\rangle + \frac{1}{\sqrt{2}}|1\rangle \otimes U^p|u\rangle \tag{71}$$

$$= \frac{1}{\sqrt{2}}\Big(|0\rangle \otimes |u\rangle + |1\rangle \otimes e^{2\pi i p\,\theta}|u\rangle\Big) = \frac{1}{\sqrt{2}}\Big(|0\rangle + e^{2\pi i p\,\theta}|1\rangle\Big) \otimes |u\rangle \, . \tag{72}$$

This is an example of *phase kickback*, in which the phase operation in the target register makes its way back into the control register. We now string these gates together to form the front-end of the circuit composed of the gates $C^nU^{2^n}$ for $n \in \{0, 1, \cdots, m-1\}$, as illustrated in Fig. 4, with the least significant power $p = 2^0$ attached to the least significant m-th target qubit (as Convention 2 dictates). We see that the output state of the front-end becomes

$$|\psi_2\rangle = \frac{1}{2^{m/2}}\Big(|0\rangle + e^{2\pi i\,2^{m-1}\theta}|1\rangle\Big)_1 \otimes \Big(|0\rangle + e^{2\pi i\,2^{m-2}\theta}|1\rangle\Big)_2 \otimes \cdots \otimes \tag{73}$$

$$\Big(|0\rangle + e^{2\pi i\,2^1\theta}|1\rangle\Big)_{m-1} \otimes \Big(|0\rangle + e^{2\pi i\,2^0\theta}|1\rangle\Big)_m \otimes |u\rangle$$

$$= \frac{1}{2^{m/2}} \sum_{k_1=0,1} e^{2\pi i\,\theta\,2^{m-1}k_1}|k_1\rangle \otimes \sum_{k_2=0,1} e^{2\pi i\,\theta\,2^{m-2}k_2}|k_2\rangle \otimes \cdots \otimes \sum_{k_m=0,1} e^{2\pi i\,\theta\,2^0k_m}|k_m\rangle \otimes |u\rangle$$

$$= \frac{1}{2^{m/2}} \sum_{k_1} \sum_{k_2} \cdots \sum_{k_m} e^{2\pi i\,\theta\left(2^{m-1}k_1 + 2^{m-2}k_2 + \cdots + 2^0k_m\right)}|k_1\rangle \otimes |k_2\rangle \otimes \cdots \otimes |k_m\rangle \otimes |u\rangle$$

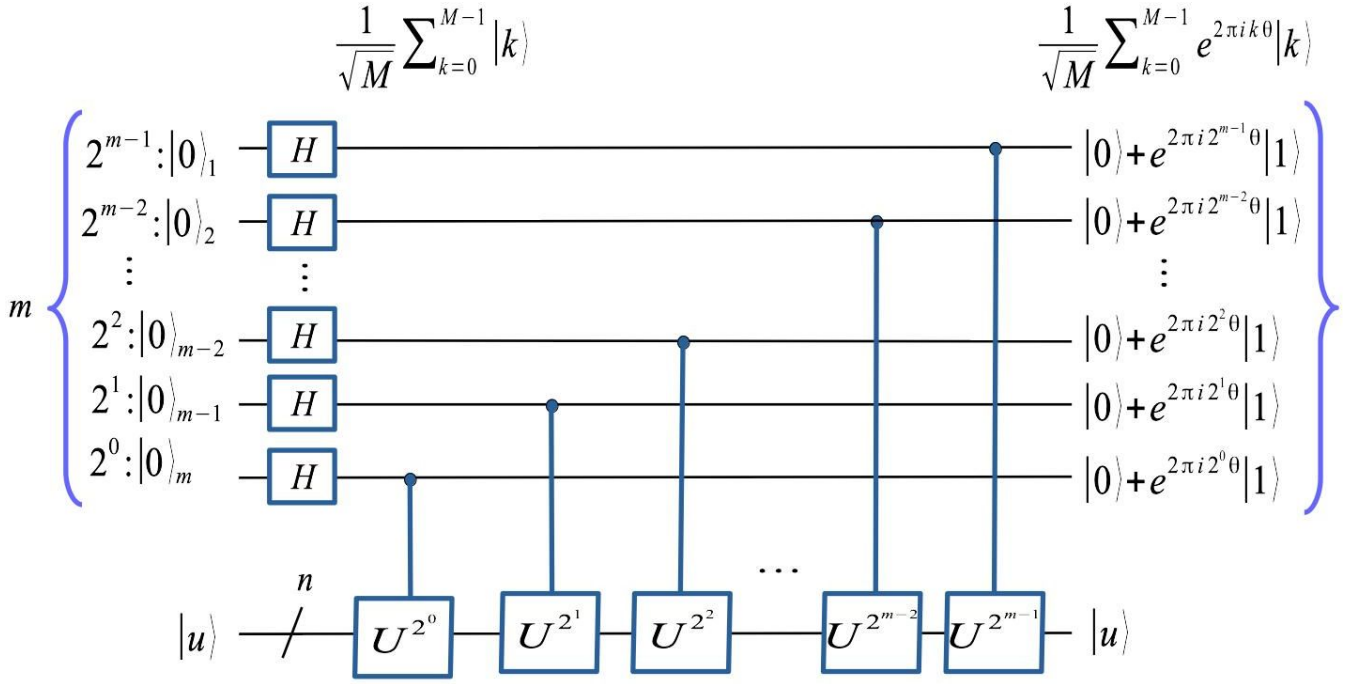


Figure 4: *QPE front-end: Convention 2 (physics and mathematics).*

$$= \frac{1}{\sqrt{M}}\sum_{k=0}^{M-1} e^{2\pi i k\,\theta}\,|k\rangle\otimes|u\rangle\,, \tag{74}$$

where $M = 2^m$ and $k = 2^{m-1}k_1 + 2^{m-2}k_2 + \cdots 2^1 k_{m-1} + 2^0 k_m$, with $k_r \in \{0, 1\}$.

This result is valid for a general phase angle θ.

For simplicity, let us first suppose that the binary form of the phase angle terminates after exactly m bits, so that

$$\theta = 0.\theta_1\,\theta_2\,\cdots\theta_{m-1}\,\theta_m \quad \text{where} \quad \theta_r \in \{0, 1\} \tag{75}$$

$$= \frac{\theta_1}{2} + \frac{\theta_2}{2^2} + \cdots + \frac{\theta_{m-1}}{2^{m-1}} + \frac{\theta_m}{2^m}\,. \tag{76}$$

We will shortly extend the argument to general phase angles that do not not terminate. We see that (76) implies that $M\theta$ can be written as a binary integer,

$$\ell_\theta \equiv M\theta = 2^m\theta = 2^{m-1}\,\theta_1 + 2^{m-2}\,\theta_2 + \cdots + 2^0\,\theta_m \tag{77}$$

$$= \theta_1\,\theta_2\,\cdots\theta_m \in \{0, 1, \cdots, M-1\}\,. \tag{78}$$

Upon using the relation $\theta = \ell_\theta/M$, we can now express the output state of the front-end as

$$|\psi_2\rangle = \frac{1}{\sqrt{M}}\sum_{k=0}^{M-1} e^{2\pi i\, k\ell_\theta/M}|k\rangle\otimes|u\rangle \tag{79}$$

$$= QFT\,|\ell_\theta\rangle\otimes|u\rangle\,, \tag{80}$$

where we have used the definition of the QFT operator (3). Therefore, the back-end of the QPE circuit will consist of an inverse QFT acting on the control register, as illustrated in Fig. 5. In Fig. 6 we expand the $QFT^\dagger$ circuit explicitly. In either case, the final output state is given by

$$|\psi_3\rangle = QFT^\dagger\,|\psi_2\rangle = |\ell_\theta\rangle\otimes|u\rangle\,, \tag{81}$$

where $\ell_\theta \equiv 2^m\theta \in \{0, 1, \cdots, M-1\}$. Some authors denote the state $|\ell_\theta\rangle$ by $|2^m\theta\rangle$. Upon measuring the control register, we will find $\ell_\theta = \theta_1\cdots\theta_m$ for $\theta_r \in \{0, 1\}$, and the corresponding phase is then given *exactly* by $\theta = \ell_\theta/2^m = 0.\theta_1\cdots\theta_m$, in agreement with (75). We see that the QPE circuit in Fig. 5 does indeed extract the correct phase.

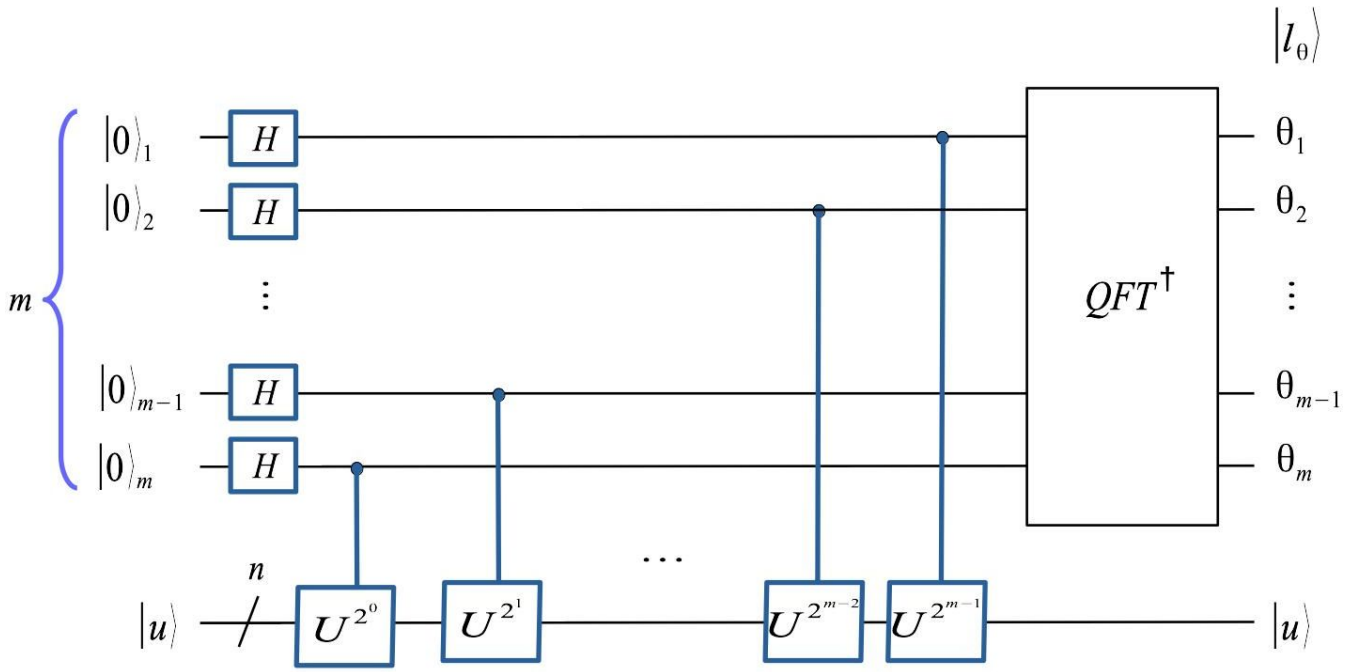


Figure 5: *QPE: Convention 2. The QPE circuit with the inverse quantum Fourier transform $QFT^\dagger$ represented abstractly by a box.*

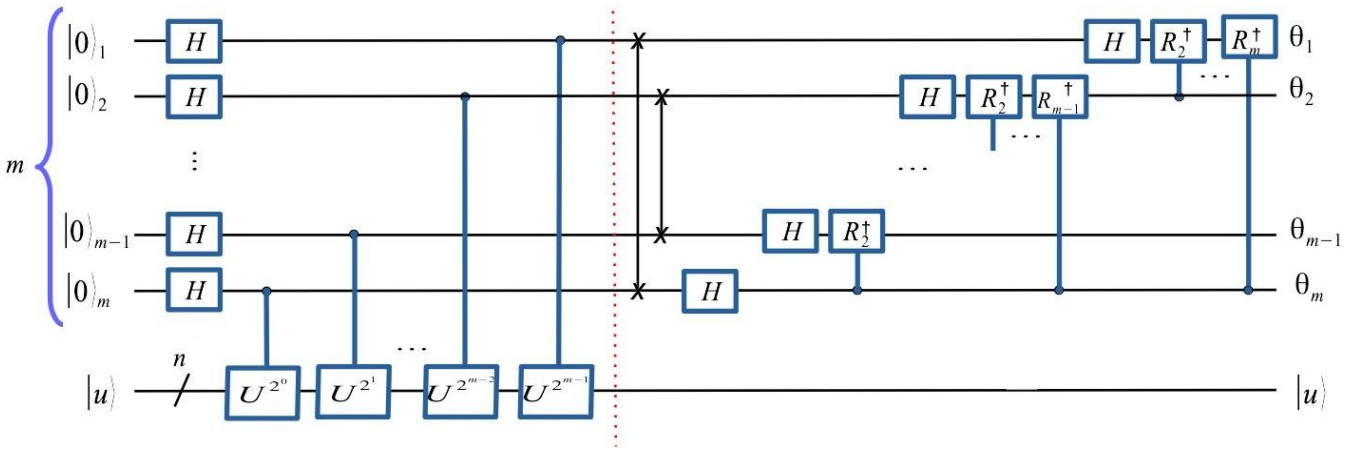


Figure 6: *The $QFT^\dagger$ circuit has been expanded to illustrate the full complexity of the QPE circuit.*

3.2. Convention 1: OpenQASM/Qiskit

We now examine the OpenQASM/Qiskit convention. As before, we string the CU^p operators together to form the front-end of the circuit, with the smallest power p being attached to the 0-th qubit (the upper and least significant qubit of the circuit). The corresponding front-end is illustrated in Fig. 7.
Similarly to (74), the output of the front-end can be expressed by

$$\begin{aligned} |\psi_2\rangle &= \frac{1}{2^{m/2}}\left(|0\rangle + e^{2\pi i\, 2^0\theta}|1\rangle\right)_0 \otimes \left(|0\rangle + e^{2\pi i\, 2^1\theta}|1\rangle\right)_1 \otimes \cdots \otimes \\ &\qquad \left(|0\rangle + e^{2\pi i\, 2^{m-2}\theta}|1\rangle\right)_{m-2} \otimes \left(|0\rangle + e^{2\pi i\, 2^{m-1}\theta}|1\rangle\right)_{m-1} \otimes |u\rangle \qquad (82) \\ &= \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} e^{2\pi i k\,\theta}|k\rangle \otimes |u\rangle\,, \qquad (83) \end{aligned}$$

where $M = 2^m$ and $k = 2^{m-1}k_{m-1} + 2^{m-2}k_{m-2} + \cdots 2^1 k_1 + 2^0 k_0$, with $k_r \in \{0, 1\}$. We first consider the simple case of

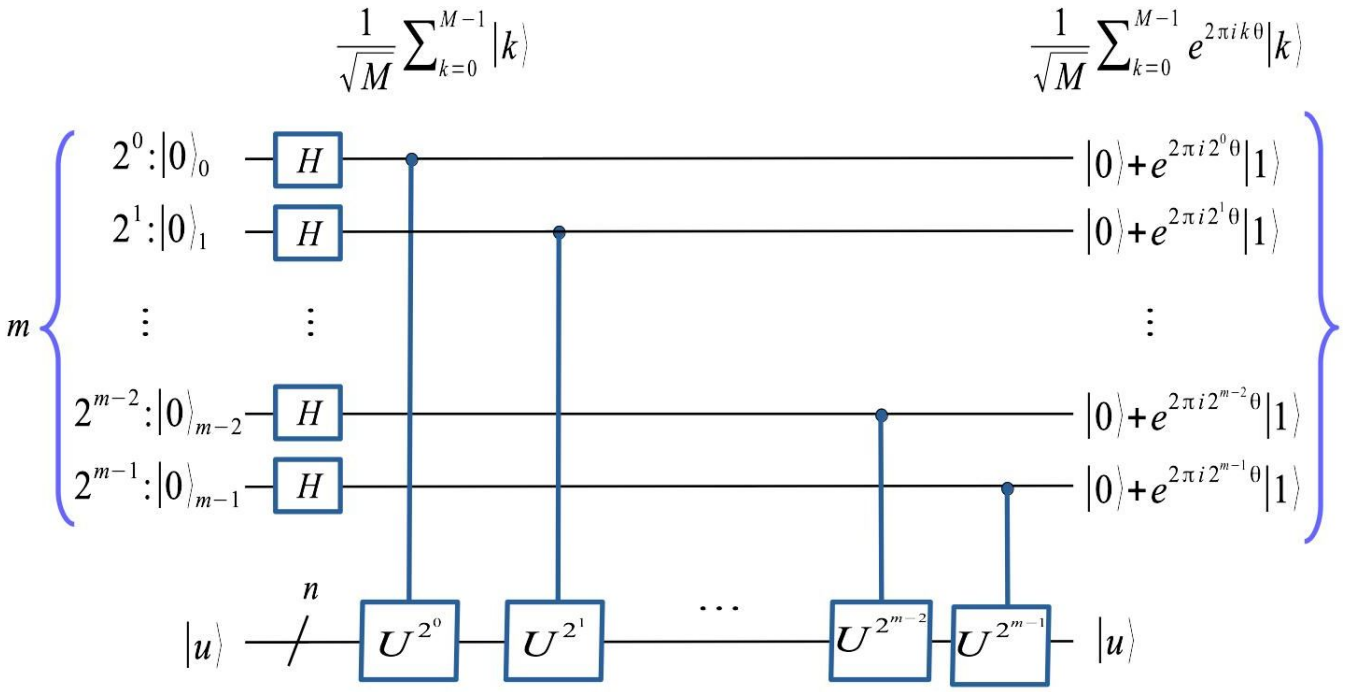


Figure 7: *QPE front-end: Convention 1 (OpenQASM/Qiskit).*

an m-bit phase angle,

$$\begin{aligned}\theta &= 0.\theta_{m-1}\,\theta_{m-2}\cdots\theta_1\,\theta_0 \quad \text{where } \theta_r\in\{0,1\}\\ &= \frac{\theta_{m-1}}{2^1}+\frac{\theta_{m-2}}{2^2}+\cdots+\frac{\theta_1}{2^{m-1}}+\frac{\theta_0}{2^m}\,.\end{aligned}\tag{84}$$

As before, this expression implies that $M\theta$ is a binary integer between 0 and $M-1$:

$$\ell_\theta\equiv M\theta=2^m\theta = 2^{m-1}\,\theta_{m-1}+2^{m-2}\,\theta_{m-2}+\cdots+2^1\,\theta_1+2^0\,\theta_0 \tag{85}$$

$$= \theta_{m-1}\,\theta_{m-2}\cdots\theta_1\,\theta_0\in\{0,1,\cdots,M-1\}\,. \tag{86}$$

Again, the relation $\theta=\ell_\theta/M$ and definition (3) of the QFT imply that the output of the front-end becomes

$$\begin{aligned}|\psi_2\rangle &= \frac{1}{\sqrt{M}}\sum_{k=0}^{M-1}e^{2\pi i k\,\theta}|k\rangle\otimes|u\rangle=\frac{1}{\sqrt{M}}\sum_{k=0}^{M-1}e^{2\pi i\,k\ell_\theta/M}|k\rangle\otimes|u\rangle\\ &= QFT\,|\ell_\theta\rangle\otimes|u\rangle\,.\end{aligned}\tag{87}$$

Upon taking the inverse Fourier transform, the final state of the QPE circuit is

$$|\psi_3\rangle=QFT^\dagger\,|\psi_2\rangle=|\ell_\theta\rangle\otimes|u\rangle\,. \tag{88}$$

The back-end of the QPE circuit therefore consists of an inverse QFT operator, as illustrated in Fig. 8. We also give the full circuit for the inverse QFT in Fig. 9. We see that a measurement of the control register gives the integer $\ell_\theta=\theta_{m-1}\cdots\theta_0$, which in turn provides the correct measured phase $\theta=\ell_\theta/2^m=0.\theta_{m-1}\cdots\theta_0$, in agreement with (84).

3.3. General Phase Angles

We now turn to the case in which the phase angle θ in (69) is a general real number between 0 and 1. The analysis here applies for both Conventions. Recall that the output of the front-end in Fig. 4 or Fig. 7 is

$$|\psi_2\rangle = \frac{1}{\sqrt{M}}\sum_{k=0}^{M-1}e^{2\pi i k\,\theta}\,|k\rangle\otimes|u\rangle\,, \tag{89}$$

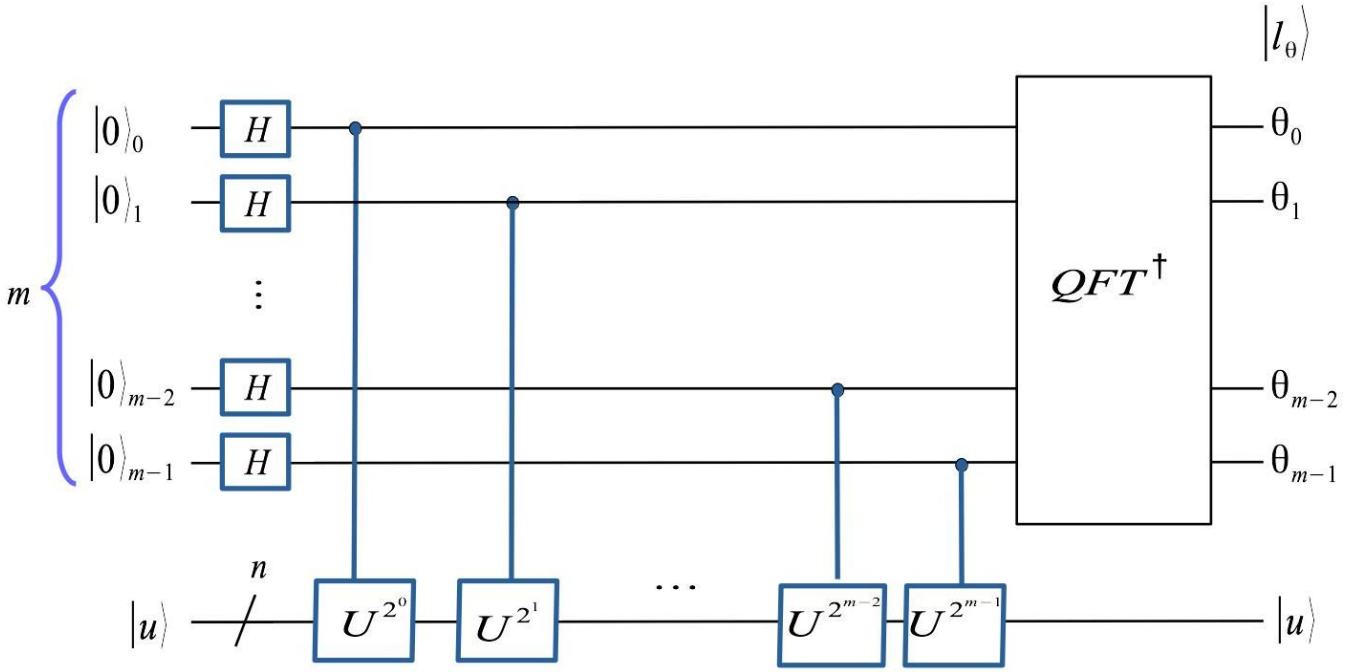


Figure 8: *QPE front- and back-end: Convention 1 (OpenQASM/Qiskit)*

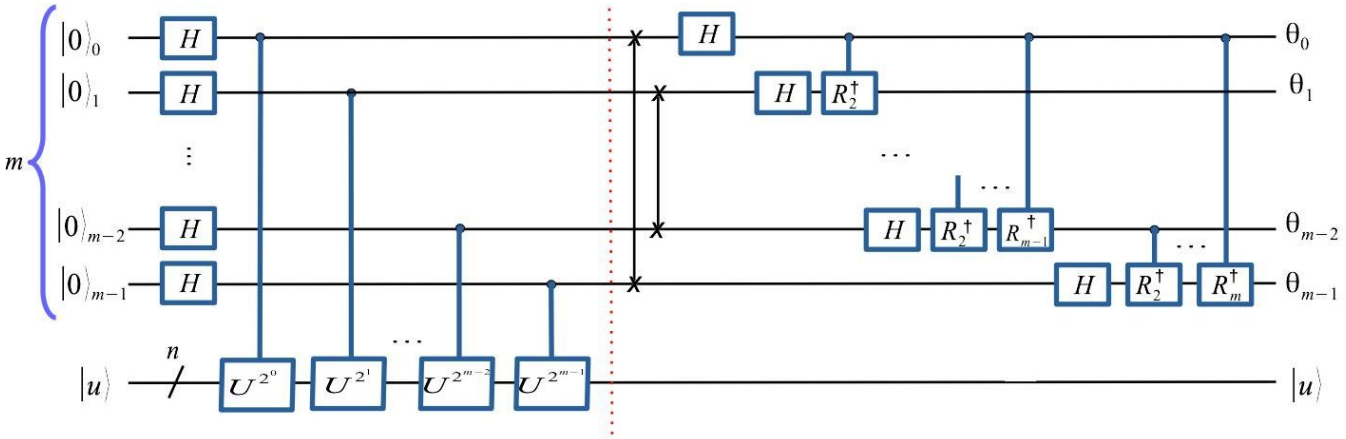


Figure 9: *QPE: Convention 1. Expanded $QFT^\dagger$.*

where this result holds for general phase angles θ. The output of the QPE circuit is then given by $|\psi_3\rangle = QFT^\dagger\,|\psi_2\rangle$, where the inverse of the QFT operator takes the form

$$QFT^\dagger = \frac{1}{\sqrt{M}} \sum_{\ell=0}^{M-1} \sum_{k=0}^{M-1} e^{-2\pi i\, k\ell/M}\, |\ell\rangle\langle k| \,. \tag{90}$$

We therefore obtain the final state

$$|\psi_3\rangle \;=\; QFT^\dagger|\psi_2\rangle = \frac{1}{M} \sum_{\ell=0}^{M-1} \sum_{k=0}^{M-1} e^{-2\pi i\, k\ell/M}\, e^{2\pi i k\,\theta}\, |\ell\rangle \otimes |u\rangle \tag{91}$$

$$= \frac{1}{M} \sum_{\ell=0}^{M-1} \sum_{k=0}^{M-1} [e^{2\pi i\,(\theta-\ell/M)}]^k\, |\ell\rangle \otimes |u\rangle = \frac{1}{M} \sum_{\ell=0}^{M-1} \frac{1-e^{2\pi i\,(\theta-\ell/M)M}}{1-e^{2\pi i\,(\theta-\ell/M)}}\, |\ell\rangle \otimes |u\rangle \,, \tag{92}$$

where we have performed an exact finite geometric sum. Since this is such an important result, we summarize it below:

$$|\psi_3\rangle = \sum_{\ell=0}^{M-1} A_\ell(\theta)\,|\ell\rangle \otimes |u\rangle \tag{93}$$

with amplitudes

$$A_\ell(\theta) \equiv \frac{1}{M}\,\frac{1-e^{2\pi i\,(\theta-\ell/M)M}}{1-e^{2\pi i\,(\theta-\ell/M)}} \tag{94}$$

$$= \frac{1}{M}\,\frac{1-e^{2\pi i\,(\theta-\theta_\ell)M}}{1-e^{2\pi i\,(\theta-\theta_\ell)}} = \frac{1}{M}\,\frac{1-e^{2\pi i\,(\ell_\theta-\ell)}}{1-e^{2\pi i\,(\ell_\theta-\ell)/M}}\,. \tag{95}$$

In (94) we have expressed the amplitude in terms of the fundamental quantities θ and ℓ, while (95) expresses the amplitude in terms of the m-bit measured phase $\theta_\ell \equiv \ell/M$ and the "mode number" $\ell_\theta \equiv \theta M$ (which might or might not be an integer). We now find that the probability of measuring the ℓ-th state is

$$P_\ell(\theta) = \left|A_\ell(\theta)\right|^2 = \frac{1}{M^2}\,\frac{\sin^2\left[\pi\left(\theta-\dfrac{\ell}{M}\right)M\right]}{\sin^2\left[\pi\left(\theta-\dfrac{\ell}{M}\right)\right]} \tag{96}$$

$$= \frac{1}{M^2}\,\frac{\sin^2\left[\pi(\theta-\theta_\ell)M\right]}{\sin^2\left[\pi(\theta-\theta_\ell)\right]} = \frac{1}{M^2}\,\frac{\sin^2\left[\pi(\ell_\theta-\ell)\right]}{\sin^2\left[\pi(\ell_\theta-\ell)/M\right]}\,. \tag{97}$$

The probability $P_\ell(\theta)$ is maximum for the state $\ell \in \{0, 1, \cdots, M-1\}$ for which $\delta = \theta - \ell/M = \theta - \theta_\ell$ is minimum. For large values of $M = 2^m$, the probability $P_\ell(\theta)$ is sharply peaked about θ, as illustrated in Fig. 10. We note that the value of $\ell_\theta \equiv M\theta$ need not be an m-bit integer (nor indeed, an integer at all). The point to be emphasized is that when ℓ_θ is not an integer, then the QPE circuit becomes probabilistic in nature. This will turn out to be a key feature of Shor's factoring circuit.

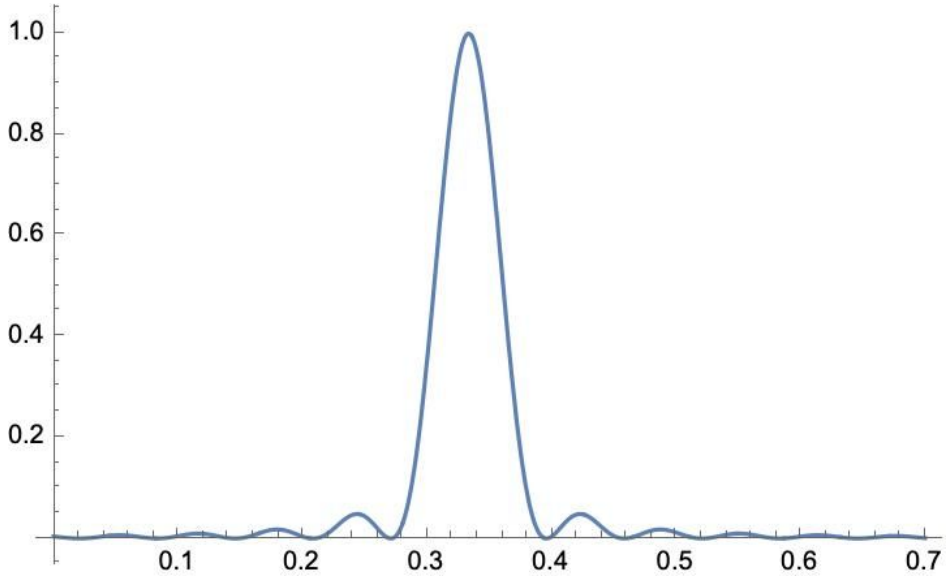


Figure 10: *The probability $P_\ell(\theta) = |A_\ell(\theta)|^2$ as a function of the measured phase angle $\theta_\ell \equiv \ell/M$ for $M = 2^4 = 16$, where $\theta = 1/3$. Note that $A_\ell(\theta) \approx 1$ for $\theta_\ell \approx \theta$.*

In fact, this analysis implicitly assumes that $\ell_\theta = M\theta$ in amplitude (94) is not an integer, as the expression for $A_\ell(\theta)$ then requires a more delicate treatment. To see this, let us write the amplitude in terms of the quantity ℓ_θ, so that

$$A_{\ell,\ell_\theta} \equiv A_\ell(\ell_\theta/M) = \frac{1}{M}\,\frac{1-e^{2\pi i\,(\ell_\theta-\ell)}}{1-e^{2\pi i\,(\ell_\theta-\ell)/M}}\,. \tag{98}$$

We have already considered the case in which ℓ_θ is a non-negative integer in the previous section, and consistency demands that amplitude (98), or equivalently (94), must collapses to the Kronecker-delta form

$$A_\ell(\theta) = \delta_{\ell,\ell_\theta} \tag{99}$$

when ℓ_θ becomes an integer. We will now show how the expression for $A_\ell(\theta)$ in (98) reduces to this simpler form. Note that the numerator in (98) vanishes for every integer ℓ_θ (and therefore for every value of $\ell_\theta - \ell$, as ℓ is itself an integer). Consequently, the only way one can obtain a non-zero probability is when the denominator also vanishes, so that we have the indeterminate form $0/0$. However, the denominator vanishes only for $\ell_\theta - \ell$ such that

$$(\ell_\theta - \ell)/M = n \;\text{ for any } n \in \mathbb{Z}\,, \tag{100}$$

or equivalently,

$$\ell = \ell_\theta - nM\,. \tag{101}$$

The index $\ell \in \{0, 1, \cdots, M-1\}$ is consequently out of range for every value of $n \in \mathbb{Z}$ except $n = 0$; therefore, $A_\ell(\theta) = 0$ except when

$$\ell = \ell_\theta \equiv M\theta\,. \tag{102}$$

The value of the amplitude for $\ell = \ell_\theta$ must of course be unity (up to an arbitrary phase), and indeed it is. As previously mentioned, when $\ell = \ell_\theta$ in (98), we obtain the indeterminate form $0/0$. We therefore replace $\ell_\theta - \ell$ by a small displacement ε and the take the limit $\varepsilon \to 0$, in which case the associated amplitude becomes

$$A = \lim_{\varepsilon\to 0} \frac{1}{M}\,\frac{1 - e^{2\pi i\,\varepsilon}}{1 - e^{2\pi i\,\varepsilon/M}} = 1\,, \tag{103}$$

as expected. This establishes expression (99). In this case, the final state (93) becomes the Eigenstate

$$|\psi_3\rangle = |\ell_\theta\rangle \otimes |u\rangle\,, \quad \text{where } \ell_\theta = M\theta \in \{0, 1, \cdots, M-1\}\,, \tag{104}$$

as we obtained in the previous section.

3.4. Generalized Input States

Suppose now that we choose the work state $|u\rangle$ to be a linear sum over all Eigenstates of U. Then the output $|\psi_2\rangle$ in equation (89) will become a corresponding sum over these Eigenstates. That is to say, for the Eigenstates

$$U|u_s\rangle = e^{2\pi i\,\theta_s}\,|u_s\rangle\,, \tag{105}$$

let us populate the work register with the linear combination of states

$$|u\rangle = \sum\nolimits_s a_s\,|u_s\rangle\,. \tag{106}$$

We now see that the output of the front-end (89) becomes

$$|\psi_2\rangle \;=\; \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} \sum\nolimits_s a_s\, e^{2\pi i k\,\theta_s}\,|k\rangle \otimes |u_s\rangle\,, \tag{107}$$

and that the control and work registers are now entangled. The output of QPE circuit is given by $|\psi_3\rangle = QFT^\dagger|\psi_2\rangle$, and using the form of $QFT^\dagger$ in (90), we find

$$|\psi_3\rangle \;=\; QFT^\dagger|\psi_2\rangle = \sum_{\ell=0}^{M-1} \sum\nolimits_s A_\ell(\theta_s)\,|\ell\rangle \otimes |u_s\rangle\,, \tag{108}$$

where the amplitudes $A_\ell(\theta_s)$ are defined by

$$A_\ell(\theta_s) \equiv \frac{a_s}{M}\,\frac{1-e^{2\pi i\,(\theta_s-\ell/M)M}}{1-e^{2\pi i\,(\theta_s-\ell/M)}} \tag{109}$$

$$= \frac{a_s}{M}\,\frac{1-e^{2\pi i\,(\theta_s-\theta_\ell)M}}{1-e^{2\pi i\,(\theta_s-\theta_\ell)}} = \frac{a_s}{M}\,\frac{1-e^{2\pi i\,(\ell_s-\ell)}}{1-e^{2\pi i\,(\ell_s-\ell)/M}} \tag{110}$$

when $\ell_s \equiv M\theta_s$ is not an integer. In contrast, for integer values $\ell_s \in \{0, 1, \cdots, M-1\}$, then expression (109) reduces to

$$A_\ell(\theta_s) \equiv a_s\,\delta_{\ell,\ell_s}\,. \tag{111}$$

For general values of θ_s, we therefore find that the probability of measuring a specific ℓ-s state is given by

$$P_\ell(\theta_s) = \left|A_\ell(\theta_s)\right|^2 = \frac{|a_s|^2}{M^2}\,\frac{\sin^2\left[\pi\left(\theta_s-\dfrac{\ell}{M}\right)M\right]}{\sin^2\left[\pi\left(\theta_s-\dfrac{\ell}{M}\right)\right]} \tag{112}$$

$$= \frac{|a_s|^2}{M^2}\,\frac{\sin^2\left[\pi(\theta_s-\theta_\ell)M\right]}{\sin^2\left[\pi(\theta_s-\theta_\ell)\right]} = \frac{|a_s|^2}{M^2}\,\frac{\sin^2\left[\pi(\ell_s-\ell)\right]}{\sin^2\left[\pi(\ell_s-\ell)/M\right]}\,. \tag{113}$$

As before, for large values of $M = 2^m$, the probability is sharply peaked about the control states for which $\ell \approx \ell_s \equiv M\theta_s$, while the amplitudes a_s determine the most likely values of θ_s.

When all of the angles θ_s become m-bit rational numbers, this analysis simplifies considerably. In this case, the parameters

$$\ell_s \equiv M\theta_s = 2^m\theta_s \tag{114}$$

are all integers, and the output of the front-end (107) can be written

$$|\psi_2\rangle = \frac{1}{\sqrt{M}}\sum_{k=0}^{M-1}\sum\nolimits_s a_s\,e^{2\pi i\,k\theta_s}\,|k\rangle\otimes|u_s\rangle \tag{115}$$

$$= \sum\nolimits_s a_s\,\frac{1}{\sqrt{M}}\sum_{k=0}^{M-1} e^{2\pi i\,k\ell_s/M}\,|k\rangle\otimes|u_s\rangle \tag{116}$$

$$= \sum\nolimits_s a_s\,QFT\;|\ell_s\rangle\otimes|u_s\rangle\,, \tag{117}$$

and therefore

$$|\psi_3\rangle = QFT^\dagger\,|\psi_2\rangle = \sum\nolimits_s a_s\,|\ell_s\rangle\otimes|u_s\rangle\,. \tag{118}$$

A measurement of the system now yields one of the states $|\ell_s\rangle\otimes|u_s\rangle$ with probability $P_s = |a_s|^2$. All other states have vanishing probability! This result can also be obtained directly from the general final state (108). In this case, when all phase angles θ_s becomes m-bit fractions, then the amplitudes (109) reduce to $A_\ell(\theta_s) = a_s\,\delta_{\ell,\ell_s}$, so that only the terms for which $\ell = \ell_s \equiv M\theta_s$ for some value of s will contribute, and the general expression (108) collapses to (118).

4. Continued Fractions

In an effort to render this work self-contained, in this section we take a mathematical digression to briefly introduce the theory of *continued fractions*, a topic with which many readers might not be entirely familiar. In Section 5.4, we will employ continued fractions in the post-quantum processing stage of Shor's algorithm to extract the *exact* phase $\phi_s = s/r$ of the modular exponentiation operator $U_{a,N}$ from the measured (and *approximate*) m-bit phase value $\tilde{\phi}$. As

we have emphasized, this will provide the requisite period r from which the factors of N can be inferred. Continued fractions, however, are interesting in their own right, and they provide a number of fascinating connections between the integers and the real numbers. In classical mathematics, for example, continued fractions were employed to find rational approximations to many irrational numbers of interest (something quite useful before the advent of modern computers and calculators). We introduce the subject by considering the following infinite continued fraction

$$x = 1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \ddots}}}}, \tag{119}$$

where the fraction x telescopes downward without end. Note that the denominator of the fractional piece after the initial 1 takes the same form as the continued fraction itself, so that we can express (119) by the equation

$$x = 1 + \frac{1}{x}. \tag{120}$$

At the risk of introducing a spurious (negative) solution, we multiply (120) by x to obtain the quadratic equation $x^2 = x + 1$. The positive solution to this equation is $x = (1 + \sqrt{5})/2$, which we recognize as the golden mean! Continued fractions are interesting indeed. In fact, any real number can be expressed as a continued fraction consisting of a sequence of well-chosen integers using a simple and easily executed algorithm.

We shall concentrate on continued fractions that terminate after a finite number of iterations, thereby producing a rational number. We define a *finite continued fraction* as a number of the form

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \ddots + \frac{1}{a_n}}}}, \tag{121}$$

where a_0 is an integer (positive or negative) and $a_1, a_2, \cdots, a_n$ are all positive integers. We will denote a continued fraction by enumerating its integer coefficients in square brackets, so that $x = [a_0; a_1, a_2, \cdots, a_n]$. It is traditional to offset the first integer a_0 with a semicolon. For our purposes, the most important attributes of continued fractions are their so-called *convergents*, whose definition is formalized below.

Definition 1. Suppose $x = [a_0; a_1, a_2, \cdots, a_n]$ is a continued fraction. Any continued fraction of the form $[a_0; a_1, a_2, \cdots, a_m]$ for $m \leq n$ is called a *convergent* of the original continued fraction for x.

For example,

$$[a_0; a_1, a_2, a_3] = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3}}} \tag{122}$$

is a convergent of the continued fraction $x = [a_0; a_1, a_2, a_3, \cdots, a_n]$. In fact, we can regard the convergents of a continued fraction of x as systematically improving rational approximations to x. As we have emphasized, any finite continued fraction gives a rational number. It turns out that the converse is also true, namely, that any rational number can be represented as a *finite* continued fraction with integer coefficients. There exists a simple and efficient (polynomial time) algorithm for determining the associated continued fraction of a rational number. The algorithm is best explained through an example, so let us consider the following rational approximation to π:

$$3.1415 \quad = \quad \frac{31415}{10000} = \frac{6283}{2000} = 3 + \frac{283}{2000} = 3 + \cfrac{1}{\cfrac{2000}{283}} \tag{123}$$

$$= 3 + \cfrac{1}{7 + \cfrac{19}{283}} = 3 + \cfrac{1}{7 + \cfrac{1}{\cfrac{283}{19}}} = 3 + \cfrac{1}{7 + \cfrac{1}{14 + \cfrac{17}{19}}} \tag{124}$$

$$= 3 + \cfrac{1}{7 + \cfrac{1}{14 + \cfrac{1}{\cfrac{19}{17}}}} = 3 + \cfrac{1}{7 + \cfrac{1}{14 + \cfrac{1}{1 + \cfrac{2}{17}}}} \tag{125}$$

$$= 3 + \cfrac{1}{7 + \cfrac{1}{14 + \cfrac{1}{1 + \cfrac{1}{\cfrac{17}{2}}}}} = 3 + \cfrac{1}{7 + \cfrac{1}{14 + \cfrac{1}{1 + \cfrac{1}{8 + \cfrac{1}{2}}}}} \quad , \tag{126}$$

and therefore $3.1415 = [3; 7, 14, 1, 8, 2]$. This algorithm consists of successive inversions of rational numbers, followed by splitting the inverted form into an integer plus a rational piece, and continuing this process again. For example, in line (123) the rational contribution $283/2000$ is inverted to form the equivalent number $1/(2000/283)$, and in the first line of (124), the denominator $2000/283 > 1$ is then split into its equivalent form $7 + 19/283$. We then invert the rational piece to give $1/(283/19)$, and we split $283/19 > 1$ into an equivalent form $14 + 17/19$. We continue in this fashion until the procedure terminates.

In a certain sense, continued fractions are a more natural representation of real numbers than their decimal counterparts. This is because the continued fraction representation of rational numbers always terminates after a finite number of iterations, *i.e.* $x \in \mathbb{Q}$ iff $x = [a_0; a_1, \cdots, a_n]$ for some finite sequence of integer coefficients a_ℓ. In contrast, decimal representations of rational numbers need not be finite, *e.g.* the rational number $2/3 = 0.666\cdots$ has an infinite number of digits, whereas the continued fraction expansion $2/3 = [0; 1, 2]$ has only two non-zero coefficients.

Note that we have required the coefficients of continued fractions to be integers. This is because the infinite continued fraction expansion of an irrational number is then unique. Furthermore, the continued fraction expansion of a rational number is almost unique. It turns out that there are only *two* possible continued fraction expansions for any given rational number, provided the coefficients a_ℓ are *integers* (we call this *semi-uniqueness*). To see this, suppose that $x = [a_0; a_1, \cdots, a_n]$ with $a_n > 1$ is a continued fraction expansion with integer coefficients. We can rewrite the last coefficient as $a_n = (a_n - 1) + 1/1$, and consequently we can also express the rational number by the continued fraction $x = [a_0; a_1, \cdots, a_n - 1, 1]$. This means that, without loss of generality, we may take a continued fraction representation of a rational number to have either an even or an odd number of terms, providing the coefficients are integers (and we shall use this fact in proving Theorem 3 below). We will, however, sometimes find it convenient to generalize the notion of a continued fraction to allow for rational (or even real) coefficients a_ℓ. But we pay a price for doing so, as we can no longer be assured of the semi-uniqueness of the associated continued fraction.

We now prove three essential theorems concerning continued fractions. The first two establish that the convergents of a continued fraction take the form p_n/q_n, where p_n and q_n are special sequences of relatively prime integers. The third theorem can be used to relate the period r to the sequence of denominators q_n, and as we shall see in the next section, it will be essential in extracting the period from the measured phase. Our primary references for this section are Refs. [8] and [9].

Theorem 1. Let $[a_0; a_1, \cdots, a_m]$ be a continued fraction, where the coefficients a_ℓ can be either rational numbers or integers. The convergents $x_n \equiv [a_0; a_1, \cdots, a_n]$ for $n \leq m$ are equal to the ratio $x_n = p_n/q_n$, where p_n and q_n are defined through the sequence

$$\begin{aligned} p_n &= a_n p_{n-1} + p_{n-2} \\ q_n &= a_n q_{n-1} + q_{n-2} \end{aligned} \tag{127}$$

for $2 \leq n \leq m$, with the seed values

$$p_0 = a_0 \qquad q_0 = 1 \tag{128}$$

$$p_1 = a_1 a_0 + 1 \quad q_1 = a_1 \, . \tag{129}$$

Furthermore, if the coefficients a_ℓ are positive integers, then p_n and q_n are also positive integers (and they strictly increase in magnitude).

Proof. The proof will be through induction on n. For $n = 0$, the convergent is just $x_0 \equiv [a_0]$, which corresponds to $x_0 = p_0/q_0$ for $p_0 = a_0$ and $q_0 = 1$, thereby validating (128). For $n = 1$, we have the convergent

$$x_1 \equiv [a_0; a_1] = a_0 + \frac{1}{a_1} = \frac{a_1 a_0 + 1}{a_1} \, , \tag{130}$$

so that $x_1 = p_1/q_1$ with $p_1 = a_1 a_0 + 1$ and $q_1 = a_1$, thereby validating (129). This takes care of the initial seeding. To provide a bit of intuition, let us explicitly verify the $n = 2$ case for $x_2 \equiv [a_0; a_1, a_2]$. From (127) we have

$$p_2 = a_2 p_1 + p_0 = a_2 a_1 a_0 + a_2 + a_0 \tag{131}$$

$$q_2 = a_2 q_1 + q_0 = a_2 a_1 + 1 \, , \tag{132}$$

and we can therefore express the convergent x_2 in the form:

$$x_2 \equiv [a_0; a_1, a_2] = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2}} \tag{133}$$

$$= a_0 + \frac{a_2}{a_2 a_1 + 1} = \frac{a_2 a_1 a_0 + a_0 + a_2}{a_2 a_1 + 1} = \frac{p_2}{q_2} \, , \tag{134}$$

thereby validating the theorem for $n = 2$. We now assume (127) holds for some $n \geq 3$ with $x_n = [a_0; a_1, a_2, \cdots, a_n]$, and we wish to prove that it continues to hold for $n + 1$. Note that any convergent $x_{n+1} = [a_0; a_1, \cdots, a_{n-1}, a_n, a_{n+1}]$ (which has $n + 1$ coefficients) may be expressed in the alternative form

$$x_{n+1} = \underbrace{[a_0; a_1, \cdots, a_{n-1}, a_n + 1/a_{n+1}]}_{n \text{ coefficients}} \, , \tag{135}$$

which contains only n coefficients, albeit rational coefficients. We can therefore apply the induction hypothesis to (135). To this end, let $\tilde{p}_\ell/\tilde{q}_\ell$ be the sequence of convergents associated with the second form of the continued fraction for x_{n+1}. The induction hypothesis now gives

$$x_{n+1} = [a_0; a_1, \cdots, a_{n-1}, \underbrace{a_n + 1/a_{n+1}}_{\tilde{a}_n}] = \frac{\tilde{p}_n}{\tilde{q}_n} \, , \tag{136}$$

where

$$\frac{\tilde{p}_n}{\tilde{q}_n} = \frac{\tilde{a}_n \, \tilde{p}_{n-1} + \tilde{p}_{n-2}}{\tilde{a}_n \, \tilde{q}_{n-1} + \tilde{q}_{n-2}} \, . \tag{137}$$

It is clear that $\tilde{p}_{n-2} = p_{n-2}$, $\tilde{p}_{n-1} = p_{n-1}$ and $\tilde{q}_{n-2} = q_{n-2}$, $\tilde{q}_{n-1} = q_{n-1}$, and we therefore find

$$x_{n+1} = \frac{\Big(a_n + \dfrac{1}{a_{n+1}}\Big) p_{n-1} + p_{n-2}}{\Big(a_n + \dfrac{1}{a_{n+1}}\Big) q_{n-1} + q_{n-2}} = \frac{(a_n a_{n+1} + 1) p_{n-1} + p_{n-2} a_{n+1}}{(a_n a_{n+1} + 1) p_{n-1} + p_{n-2} a_{n+1}} \tag{138}$$

$$= \frac{a_{n+1}(a_n p_{n-1} + p_{n-2}) + p_{n-1}}{a_{n+1}(a_n q_{n-1} + q_{n-2}) + q_{n-1}} \quad \Leftarrow \text{ use (127)} \tag{139}$$

$$= \frac{a_{n+1}p_n + p_{n-1}}{a_{n+1}q_n + q_{n-1}} \equiv \frac{p_{n+1}}{q_{n+1}} \,. \tag{140}$$

Thus, the theorem is true for $n+1$. It is obvious that if the coefficients a_ℓ are positive integers, then p_n and q_n are as well. This completes the proof. □

Theorem 2. If the coefficients of the continued fraction $[a_0; a_1, a_2, \cdots, a_m]$ are integers, then the integers p_n and q_n of Theorem 1 are *relatively prime*, and satisfy the relation

$$q_n p_{n-1} - p_n q_{n-1} = (-1)^n \tag{141}$$

for $n \geq 1$.

Proof. We prove (141) by induction on n. From (128) and (129) we find

$$n = 1: \quad q_1 p_0 - p_1 q_0 \;=\; a_1 a_0 - (a_1 a_0 + 1)\cdot 1 = -1 = (-1)^1 \,, \tag{142}$$

so that (141) holds for $n = 1$. Similarly, expressions (131) and (132) imply that

$$n = 2: \quad q_2 p_1 - p_2 q_1 \;=\; (a_2 a_1 + 1)(a_1 a_0 + 1) - (a_2 a_1 a_0 + a_2 + a_0)a_1 = 1 = (-1)^2 \,, \tag{143}$$

so that (141) also holds for $n = 2$. Let us now assume that (141) holds for some $n \geq 3$, and let us prove that it continues to hold for $n+1$. Taking $n \to n+1$ in (127), the induction hypothesis gives

$$\begin{aligned} n+1: \quad q_{n+1}p_n - p_{n+1}q_n &= (a_{n+1}q_n + q_{n-1})p_n - (a_{n+1}p_n + p_{n-1})q_n \\ &= -(q_n p_{n-1} - p_n q_{n-1}) = -(-1)^n = (-1)^{n+1} \,. \end{aligned} \tag{144}$$

Therefore, (127) holds for $n+1$, and this completes the proof of the first part of the theorem.

We must now show that p_n and q_n have no common factors other than unity. Let us therefore assume that $k_n \geq 1$ is a common factor of p_n and q_n, so that

$$p_n = k_n \tilde{p}_n \;\text{ and }\; q_n = k_n \tilde{q}_n \;\text{ with }\; \frac{p_n}{q_n} = \frac{\tilde{p}_n}{\tilde{q}_n} \,. \tag{145}$$

Let us also assume that $k_{n-1} \geq 1$ is a common factor of p_{n-1} and q_{n-1}, so that

$$p_{n-1} = k_{n-1} \tilde{p}_{n-1} \;\text{ and }\; q_{n-1} = k_{n-1} \tilde{q}_{n-1} \;\text{ with }\; \frac{p_{n-1}}{q_{n-1}} = \frac{\tilde{p}_{n-1}}{\tilde{q}_{n-1}} \,. \tag{146}$$

Then (141) now takes the form

$$k_n k_{n-1}(\tilde{q}_n \tilde{p}_{n-1} - \tilde{p}_n \tilde{q}_{n-1}) = (-1)^n \,. \tag{147}$$

It is obvious that $\tilde{p}_n/\tilde{q}_n$ has the same continued fraction representation as p_n/q_n for all $n \geq 1$, so that $\tilde{q}_n \tilde{p}_{n-1} - \tilde{p}_n \tilde{q}_{n-1} = (-1)^n$, from which it follows that

$$k_n k_{n-1} = 1 \,. \tag{148}$$

For $n = 1$, we have $k_1 k_0 = 1$. Since $1 = q_0 = k_0\, \tilde{q}_0$, and since k_0 and q_0 are both integers, we must have $k_0 = 1$ (and $\tilde{q}_0 = 1$). Therefore, $k_1 = 1$. For $n = 2$, expression (148) becomes $k_2 k_1 = 1$, and therefore $k_2 = 1$. Continuing in this fashion, we find $k_n = 1$ for all $n \geq 1$, and hence p_n and q_n are relatively prime. □

We move on to our primary result, the theorem that will allow us to extract the desired period from an approximately measured phase angle.

Theorem 3. Let x be a rational number. If two relatively prime integers p and q satisfy

$$\left|\frac{p}{q} - x\right| \leq \frac{1}{2q^2} \,, \tag{149}$$

then p/q is necessarily a convergent of x.

Proof. Let $p/q = [a_0; a_1, \cdots, a_n]$ be the continued fraction representation for p/q, and define the convergents p_ℓ/q_ℓ for $\ell = 0, 1, \cdots, n$ as in Theorem 1, so that $p_n/q_n = p/q$. The object of the proof is to construct the continued fraction representation for x, and this will explicitly show that p/q is one of its convergents.

Let us first define the error δ for $p/q = p_n/q_n$ by

$$x - \frac{p_n}{q_n} \equiv \frac{\delta}{2q_n^2} \,. \tag{150}$$

Note that inequality (149) gives $0 \leq \delta \leq 1$. We now define the parameter

$$\lambda \equiv 2\,\frac{q_n p_{n-1} - p_n q_{n-1}}{\delta} - \frac{q_{n-1}}{q_n} \,, \tag{151}$$

and with some algebra we can show that equations (150) and (151) imply

$$x = \frac{\lambda p_n + p_{n-1}}{\lambda q_n + q_{n-1}} \,. \tag{152}$$

To see this, note that (151) allows us to write

$$\frac{2}{\delta}\,(q_n p_{n-1} - p_n q_{n-1}) = \lambda + \frac{q_{n-1}}{q_n} = \frac{\lambda q_n + q_{n-1}}{q_n} \quad \Rightarrow \tag{153}$$

$$\frac{\delta}{2} = \frac{q_n}{\lambda q_n + q_{n-1}}(q_n p_{n-1} - p_n q_{n-1}) \,. \tag{154}$$

Using this result in equation (150) gives

$$x = \frac{p_n}{q_n} + \frac{\delta}{2}\,\frac{1}{q_n^2} = \frac{p_n}{q_n} + \frac{q_n p_{n-1} - p_n q_{n-1}}{q_n(\lambda q_n + q_{n-1})} \tag{155}$$

$$= \frac{p_n(\lambda q_n + q_{n-1}) + (q_n p_{n-1} - p_n q_{n-1})}{q_n(\lambda q_n + q_{n-1})} \tag{156}$$

$$= \frac{(\lambda p_n + p_n q_{n-1}/q_n) + (p_{n-1} - p_n q_{n-1}/q_n)}{\lambda q_n + q_{n-1}} = \frac{\lambda p_n + p_{n-1}}{\lambda q_n + q_{n-1}} \,. \tag{157}$$

Now that we have established (152), Theorem 1 implies x is a continued fraction with coefficients $a_0, a_1, \cdots, a_n, a_{n+1}$, where $a_{n+1} = \lambda$, *i.e.*

$$x = [a_0; a_1, \cdots, a_n, a_{n+1}] = [a_0; a_1, \cdots, a_n, \lambda] \,. \tag{158}$$

Without loss of generality, we can assume that n is even, so that Theorem 2 gives

$$\lambda = \frac{2}{\delta} - \frac{q_{n-1}}{q_n} > 2 - \frac{q_{n-1}}{q_n} > 2 - 1 > 1 \,. \tag{159}$$

Thus, λ is a rational number greater one, and it therefore has a continued fraction expansion of the form $\lambda = [b_0; b_1, \cdots, b_m]$ (since $\lambda > 0$ we must have $b_0 > 0$, and we henceforth drop the semicolon after the initial coefficient b_0). Therefore, we find

$$x = [a_0; a_1, \cdots, a_n, b_0, \cdots, b_m] \,, \tag{160}$$

which shows that x is a finite continued fraction with $p/q = p_n/q_n$ as one of its convergents. □

5. Factoring with Shor's Algorithm

5.1. Basic Observation

We now address an essential observation on our way to building Shor's algorithm. Let N be the positive integer we wish to factor. We assume that N is not even, and not a power of a prime number (otherwise we can find a factor quickly). We say that two integers $a, b \in \mathbb{Z}$ are *congruent modulo N* provided the difference $a - b$ is divisible by N, and we express this by writing

$$a = b \pmod{N} . \tag{161}$$

That is to say, a and b are congruent modulo N provided there exists another integer $m \in \mathbb{Z}$ such that

$$a - b = mN . \tag{162}$$

Shor's factoring algorithm relies on the following observation. Suppose we can find a non-trivial or proper square root of unity modulo N. In other words, suppose that we have found an integer b such that

$$b^2 = 1 \pmod{N} . \tag{163}$$

Then b is a modular square root of unity. By a *proper* square root, we mean that

$$b \neq \pm 1 \pmod{N} . \tag{164}$$

Equation (163) implies that there exists an integer $m \in \mathbb{Z}$ such that

$$b^2 - 1 = mN . \tag{165}$$

The latter equation is the key to the factoring algorithm, as it can be expressed as

$$(b + 1)(b - 1) = mN , \tag{166}$$

and therefore, we see that the greatest common divisors $d_\pm \equiv \gcd(b \pm 1, N)$ are factors of N. We note that finding the greatest common divisor of two integers can be performed very quickly (in polynomial time) on a classical computer. It can be shown that (163) and (164) indeed lead to non-trivial or proper factors, in that $d_\pm \neq 1, N$. This is formalized in the following theorem.

Theorem 4. Let $b \in \mathbb{Z}$ be a proper square root of unity modulo N. That is to say, let $b^2 = 1 \pmod{N}$ and $b \neq \pm 1 \pmod{N}$. Then $\gcd(b + 1, N)$ and $\gcd(b - 1, N)$ are *proper* factors of N.

Proof. First consider $d = \gcd(b - 1, N)$, which is indeed a factor of N. We will show that $d \neq 1$ and $d \neq N$. The proof will be by contradiction.

- First assume $d = \gcd(b - 1, N) = N$. Then N divides $b - 1$, so that $b - 1 = mN$ for some $m \in \mathbb{Z}$, or equivalently, $b = 1 \pmod{N}$. This contradicts the fact that b is a proper root of unity.
- Now assume $d = \gcd(b - 1, N) = 1$. Since $b - 1$ and N are relatively prime, there exists integers $u, v \in \mathbb{Z}$ such that

$$(b - 1)u + Nv = 1 . \tag{167}$$

Multiplying both sides by $b + 1$ gives the expression

$$b + 1 = (b^2 - 1)u + (b + 1)Nv . \tag{168}$$

Let us divide both sides of this equation by N, and employ equation (165) for the first term $(b^2 - 1)u$ on the right-hand side of (168):

$$\frac{b+1}{N} = \underbrace{\frac{b^2-1}{N}}_{\text{integer} \equiv m} \cdot u + (b + 1) \cdot v = m \cdot u + (b + 1) \cdot v \in \mathbb{Z} , \tag{169}$$

where u, v, m, b are all integers. Thus, N divides $b + 1$, so that $b = -1 \pmod{N}$. Again, this contradicts the fact that b is a proper root of unity. Similar reasoning holds for $d = \gcd(b + 1, N)$. □

5.2. Period Finding and Factorization

Consider two integers a and N such that $a < N$. The integer N is the number we wish to factor, while a is an initial "guess" for one of the factors. We will usually refer to a as the *base*. In fact, we can randomly choose the base from $\{2, 3, \cdots, N-1\}$, provided that a and N are relatively prime, so that $\gcd(a, N) = 1$ (otherwise we have found a non-trivial factor of N). Let us now define the *order* of a modulo N as the *least* positive integer r such that

$$a^r = 1 \pmod{N} \,. \tag{170}$$

Therefore, if r is even, then $b = a^{r/2}$ is an integer, and it is a square root of unity. If it is also non-trivial, we can perform the factoring algorithm outlined above based on this value of b. If the order r is odd and a is not a perfect square, then we must try a new base a. On the other hand, if r is odd and a is a perfect square, then $b = a^{r/2}$ is still an integer square root of unity, and it can be used in the algorithm [11]. For these cases of r and a, we see that $b = a^{r/2}$ is a square root of unity, and provided that it is non-trivial, then the factors of N are given by $\gcd(a^{r/2} \pm 1, N)$.

There is an equivalent way of looking at this that employs the periodic *modular exponential function* defined by

$$f_{a\,N}(x) = a^x \pmod{N} \,, \tag{171}$$

where a, x, and N are non-negative integers. We will usually drop the subscripts and simply write $f(x)$. The modular order r is nothing more than the period of $f(x)$. To see this, note that $f(0) = 1$, and since r is the smallest integer such that $f(r) = a^r \pmod{N} = 1$, we see that $f(r) = 1$. In fact, for any argument x, we have $f(x + r) = f(x)$, and thus the order r is the period of $f(x)$. As an example, let us take $N = 15$. The base a and the number N cannot have any non-trivial common factors, and the base must satisfy $1 < a < N$, which limits the allowed values to

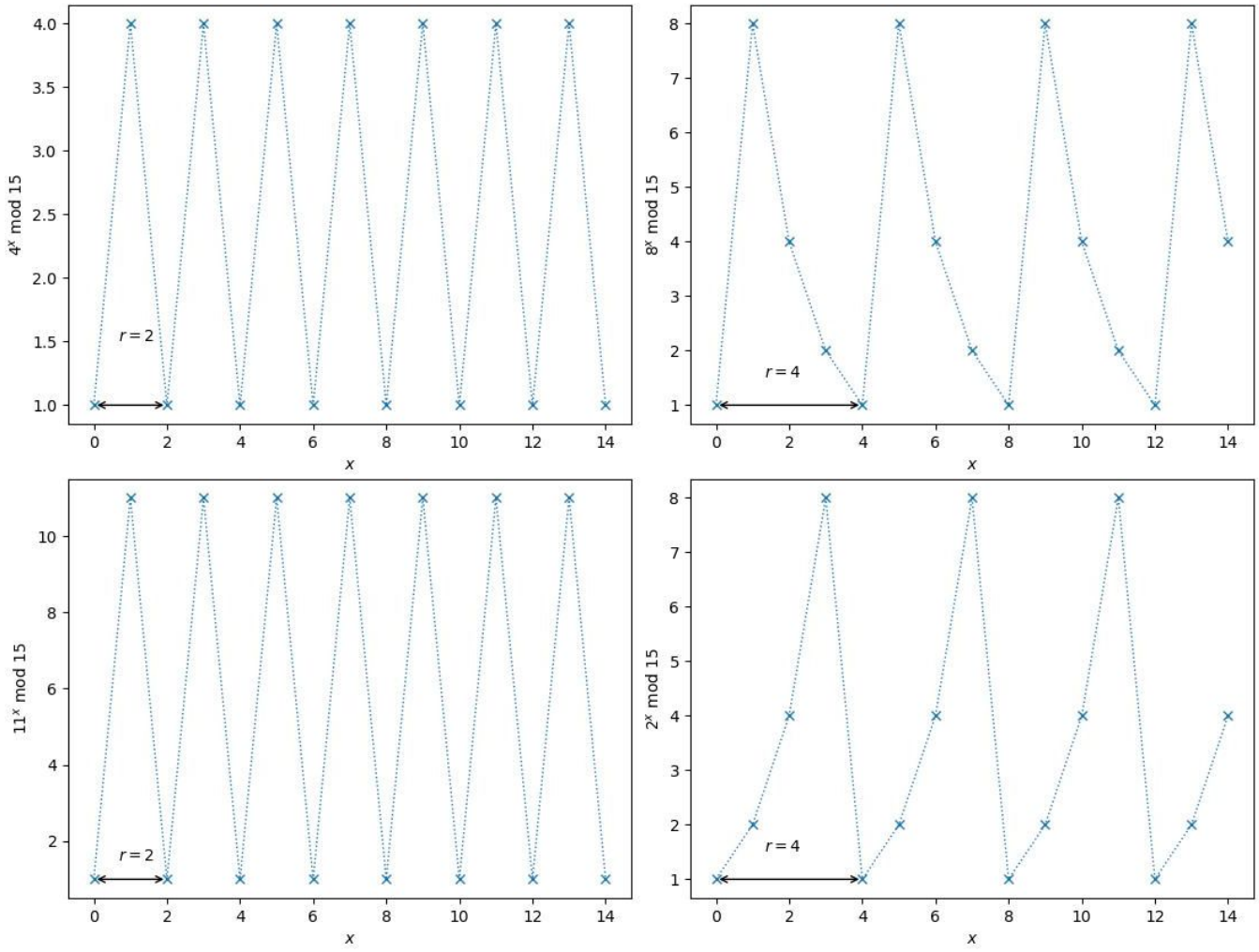


Figure 11: *The function $f(x) = a^x \pmod{N}$ for $N = 15$ and the bases $a \in \{4, 8, 11, 2\}$.*

$a \in \{2, 4, 7, 8, 11, 13, 14\}$. Figures 11 and 12 illustrate the functions $f(x)$ for $N = 15$ for these values of a (note that $a = 14$ gives a trivial square root with no factors, so we do not bother to provide a plot). We summarize below the factorization algorithm from the last section based on the periods r for the cases specified in the Figures. For the bases a in Fig. 11 we find:

• $a = 4 \Rightarrow r = 2$:
$a^{r/2} - 1 = 4^1 - 1 = 3 \;\Rightarrow\; \gcd(3, 15) = 3$
$a^{r/2} + 1 = 4^1 + 1 = 5 \;\Rightarrow\; \gcd(5, 15) = 5$

• $a = 2 \Rightarrow r = 4$:
$a^{r/2} - 1 = 2^2 - 1 = 4 - 1 = 3 \;\Rightarrow\; \gcd(3, 15) = 3$
$a^{r/2} + 1 = 2^2 + 1 = 4 + 1 = 5 \;\Rightarrow\; \gcd(5, 15) = 5$

• $a = 11 \Rightarrow r = 2$:
$a^{r/2} - 1 = 11^1 - 1 = 10 \;\Rightarrow\; \gcd(10, 15) = 5$
$a^{r/2} + 1 = 11^1 + 1 = 12 \;\Rightarrow\; \gcd(12, 15) = 3$

• $a = 8 \Rightarrow r = 4$:
$a^{r/2} - 1 = 8^2 - 1 = 64 - 1 = 63 \;\Rightarrow\; \gcd(63, 15) = 3$
$a^{r/2} + 1 = 8^2 + 1 = 64 + 1 = 65 \;\Rightarrow\; \gcd(65, 15) = 5$

• $a = 14 \Rightarrow r = 2$:
$a^{r/2} - 1 = 14^1 - 1 = 13$
$a^{r/2} + 1 = 14^1 + 1 = 15 \Leftarrow$ trivial

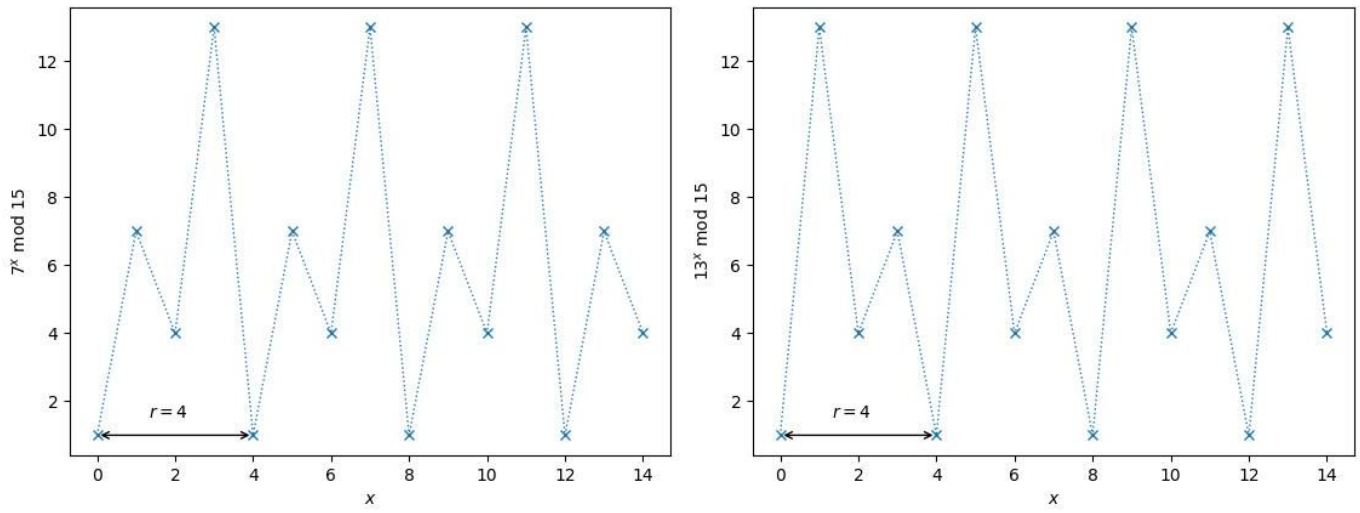


Figure 12: *The function $f(x) = a^x \pmod N$ for $N = 15$ and the bases $a \in \{7, 13\}$.*

And for the bases a in Fig. 12 we find:
• $a = 7 \Rightarrow r = 4$:
- $a^{r/2} - 1 = 7^2 - 1 = 49 - 1 = 48 \;\Rightarrow\; \gcd(48, 15) = 3$
- $a^{r/2} + 1 = 7^2 + 1 = 49 + 1 = 50 \;\Rightarrow\; \gcd(50, 15) = 5$

• $a = 13 \Rightarrow r = 4$:
- $a^{r/2} - 1 = 13^2 - 1 = 169 - 1 = 168 \;\Rightarrow\; \gcd(168, 15) = 3$
- $a^{r/2} + 1 = 13^2 + 1 = 169 + 1 = 170 \;\Rightarrow\; \gcd(170, 15) = 5$

5.3. The Factorization Circuit

Our primary references for this section are Refs. [8, 9, 12]. We will show that one can find the period r of the modular exponential function $f_{a\,N}(x) = a^x \pmod N$ by exploiting the quantum phase estimation (QPE) algorithm developed in Section 3. The crucial step in this procedure is to define a unitary operator $U_{a\,N}$ whose Eigen-phases contain information about the period r. We then employ the QPE algorithm to find the phases of $U_{a\,N}$, thereby permitting us to determine the exact value of r. Recall that the QPE algorithm consists of two quantum registers, a control register and a work register. The control register contains m qubits that dictate the resolution of the measured output phase of

$U_{a\,N}$, while the work register encodes information about the number N, and we therefore take the number of work qubits to be $n = \lceil \log_2 N \rceil$ (the binary length of N). We shall define a linear unitary operator $U_{a\,N}$ by its action on the computational basis states of the work register,

$$U_{a\,N}\,|w\rangle = |a \cdot w \pmod N\rangle\,. \tag{172}$$

We will usually drop the N and a subscripts and write $U = U_{a\,N}$ for simplicity. We shall refer to U as either the *phase* operator or the *modular exponentiation* (ME) operator. To continue, let us now solve the Eigenvalue problem for the ME operator U. This is accomplished by the simple observation that

$$U\,|a^x \pmod N\rangle \quad = \quad |a^{x+1} \pmod N\rangle \tag{173}$$

for any non-negative integer x. Let us now define the r states

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i k\,s/r}\,|a^k \pmod N\rangle \quad \text{for} \;\; s \in \{0, 1, \cdots, r-1\}\,, \tag{174}$$

from which equation (173) gives

$$U|u_s\rangle = e^{2\pi i\,\phi_s}\,|u_s\rangle \quad \text{with} \;\; \phi_s = \frac{s}{r}\,. \tag{175}$$

The possible phases of the ME operator U are therefore $\phi_s = s/r$ for $s \in \{0, 1, \cdots, r-1\}$, where r is the period of the function $f(x) = a^x \pmod N$. To prove this result, it is instructive to expand the states in (174) term-by-term,

$$\begin{aligned} |u_s\rangle \quad = \quad & \frac{1}{\sqrt{r}}\Big[\,|a^0 \pmod N\rangle + e^{-2\pi i\,s/r}\,|a^1 \pmod N\rangle + e^{-2\pi i\,2s/r}\,|a^2 \pmod N\rangle + \cdots + \\ & e^{-2\pi i\,(r-2)s/r}\,|a^{r-2} \pmod N\rangle + e^{-2\pi i\,(r-1)s/r}\,|a^{r-1} \pmod N\rangle\Big]\,. \end{aligned} \tag{176}$$

The series terminates after r terms because $a^r \pmod N = 1$, which leads us back to the first term $|1\rangle = |a^0 \pmod N\rangle$. From relation (173), we can now easily prove expression (175):

$$\begin{aligned} U|u_s\rangle \quad = \quad & \frac{1}{\sqrt{r}}\Big[\,|a^1 \pmod N\rangle + e^{-2\pi i\,s/r}|\,a^2 \pmod N\rangle + e^{-2\pi i\,2s/r}|\,a^3 \pmod N\rangle + \cdots + \\ & e^{-2\pi i\,(r-2)s/r}|\,a^{r-1} \pmod N\rangle + e^{-2\pi i\,(r-1)s/r}|\,a^r \pmod N\rangle\Big] \end{aligned} \tag{177}$$

$$\begin{aligned} = \quad & e^{2\pi i\,s/r}\,\frac{1}{\sqrt{r}}\Big[\,e^{-2\pi i\,s/r}\,|a^1 \pmod N\rangle + e^{-2\pi i\,2s/r}\,|a^2 \pmod N\rangle + e^{-2\pi i\,3s/r}\,|a^3 \pmod N\rangle \\ & + \cdots + e^{-2\pi i\,(r-1)s/r}|\,a^{r-1} \pmod N\rangle + \underbrace{e^{-2\pi i\,s}}_{1}\,|\,a^0 \pmod N\rangle\Big] \end{aligned} \tag{178}$$

$$= \quad e^{2\pi i\,s/r}\,|u_s\rangle\,. \tag{179}$$

Also note that the phases $e^{2\pi i k\,s/r}$ sum to zero over s for any value of the non-zero integers r and k. This is easy to prove, as the sum is geometric and can be performed exactly:

$$\sum_{s=0}^{r-1} e^{2\pi i\,ks/r} \quad = \quad \sum_{s=0}^{r-1} \left[e^{2\pi i\,k/r}\right]^s = \frac{1 - \left[e^{2\pi i\,k/r}\right]^r}{1 - e^{2\pi i\,k/r}} = \frac{1 - e^{2\pi i\,k}}{1 - e^{2\pi i\,k/r}} = 0 \tag{180}$$

for any integer $k \neq 0$. The sum over s in (176) therefore gives

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = |1\rangle\,, \tag{181}$$

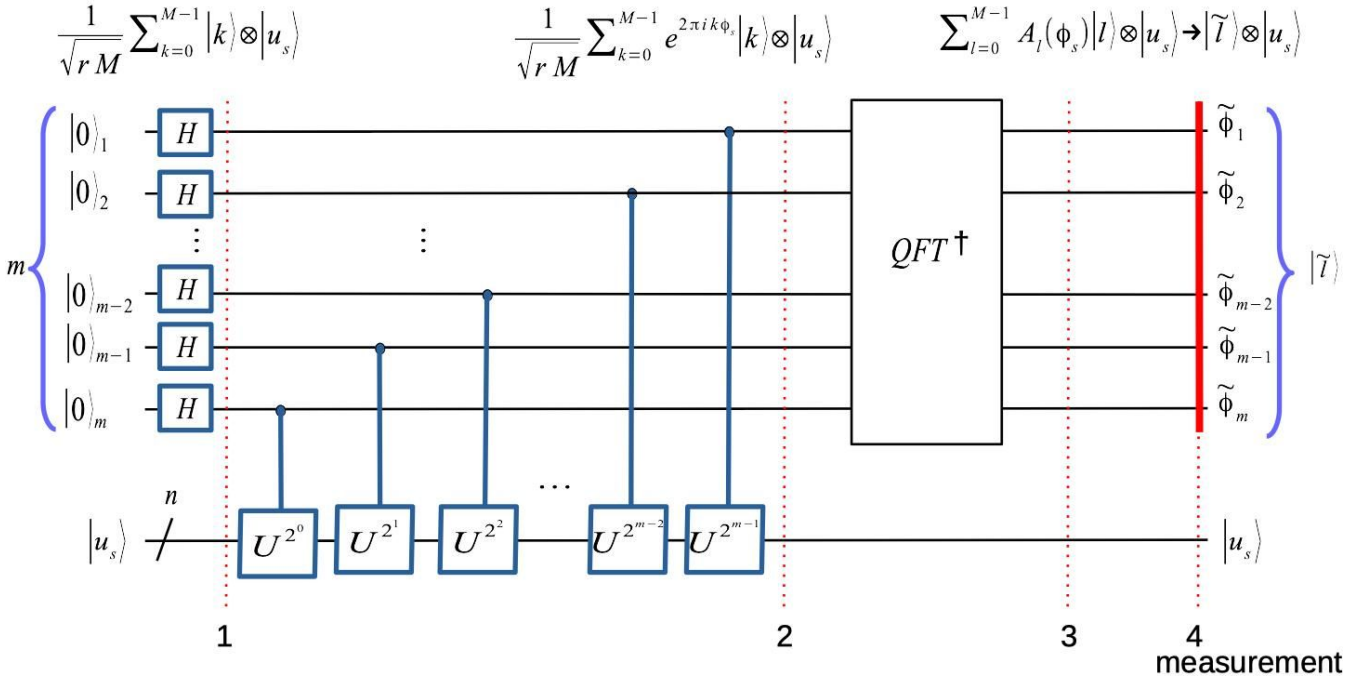


Figure 13: *First attempt at Shor's factoring circuit for the physics and mathematics Convention 2. The control register has m qubits, and the work register has* $n = \lceil \log_2 N \rceil$ *qubits. We populate the work register with one of the Eigenstates* $|u_s\rangle$*, where* $U|u_s\rangle = e^{2\pi i\phi_s}|u_s\rangle$ *with phase* $\phi_s = s/r$ *for some* $s \in \{0, 1, \cdots, r-1\}$*. Measuring the control register projects the wavefunction into an Eigenstate* $|\tilde{\ell}\,\rangle \otimes |u_s\rangle$*, where the state is indexed by the m-bit integer* $\tilde{\ell} = \tilde{\phi}_1 \cdots \tilde{\phi}_m$*, from which we obtain the measured phase* $\tilde{\phi}_\ell = \tilde{\ell}/2^m = 0.\tilde{\phi}_1 \ldots \tilde{\phi}_m$ *to m bits of accuracy. We expect* $\tilde{\phi}_\ell \approx \phi_s$*, thereby allowing us to determine r. The only problem with this reasoning is that we do not know the state* $|u_s\rangle$ *in advance since we do not* a priori *know the period r.*

which will prove to be a key ingredient for Shor's algorithm. Note that only the first term $|a^0 \pmod N\rangle = |1\rangle$ contributes to the sum, as all other terms have phases that add to zero. In fact, if we multiply (174) or (176) by $e^{2\pi i ks/r}$, we remove the phase of the term $|a^k \pmod N\rangle$, and upon summing over s we find the generalized result

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1} e^{2\pi ik\,\phi_s}\,|u_s\rangle = |a^k \pmod N\rangle = |f(k)\rangle \quad \text{where} \quad \phi_s = s/r\,. \tag{182}$$

The central observation here is that every phase $\phi_s = s/r$, except $\phi_0 = 0$, contains the period r. This is the basis of Shor's algorithm: by measuring a phase $\phi_s = s/r$ of the ME operator U for which s and r are relatively prime, we can infer the period r of the function $f(x)$. And from the corresponding factoring procedure outlined above, we can then find non-trivial factors of N.

We can now address the quantum circuit for Shor's algorithm, the first attempt of which is illustrated in Fig. 13. We employ a QPE algorithm with the phase operator U as defined in (172). We emphasize that the QPE circuit consists of a *control* register of m qubits and a *work* register of n qubits. As stated above, the work register stores information about the number N. Suppose that the work register is populated by a specific Eigenstate $|u_s\rangle$ with phase $\phi_s = s/r$, so that $n = \lceil \log_2 N \rceil$. Note that there are only r phase states $|u_s\rangle$ out of a possible 2^n work states, so that the states $|u_s\rangle$ are very sparse indeed. Upon measuring the final state of the control register, the wavefunction collapses into an Eigenstate $|\tilde{\ell}\,\rangle \otimes |u_s\rangle$ for $\tilde{\ell} \in \{0, 1, \cdots, M-1\}$. The control register index takes the measured value $\tilde{\ell} = \tilde{\phi}_1 \cdots \tilde{\phi}_m$, where each $\tilde{\phi}_k \in \{0, 1\}$ is the measured outcome of qubit $k \in \{1, 2, \cdots, m\}$ of the control register. From this we can readily infer the measured phase to be $\tilde{\phi}_\ell = \tilde{\ell}/2^m = 0.\tilde{\phi}_1 \ldots \tilde{\phi}_m$. From here on, we will place a tilde over measured quantities. If the m-bit resolution is sufficiently large, then we expect $\tilde{\phi}_\ell \approx \phi_s$, from which we can extract the value of r (provided s and r are relatively prime). In fact, as we will establish in the next section, we should choose $m = 2n + 1$ to ensure sufficient phase resolution.

By populating the work register with $|u_s\rangle$, Fig. 13 uses the QPE algorithm to find the phase $\phi_s = s/r$. The problem with this method, however, is that we do not know the state $|u_s\rangle$ in advance, as it depends upon the unknown period r. In fact, if we knew the Eigenstates $|u_s\rangle$, then we should also know the value of the period r, and we would have no

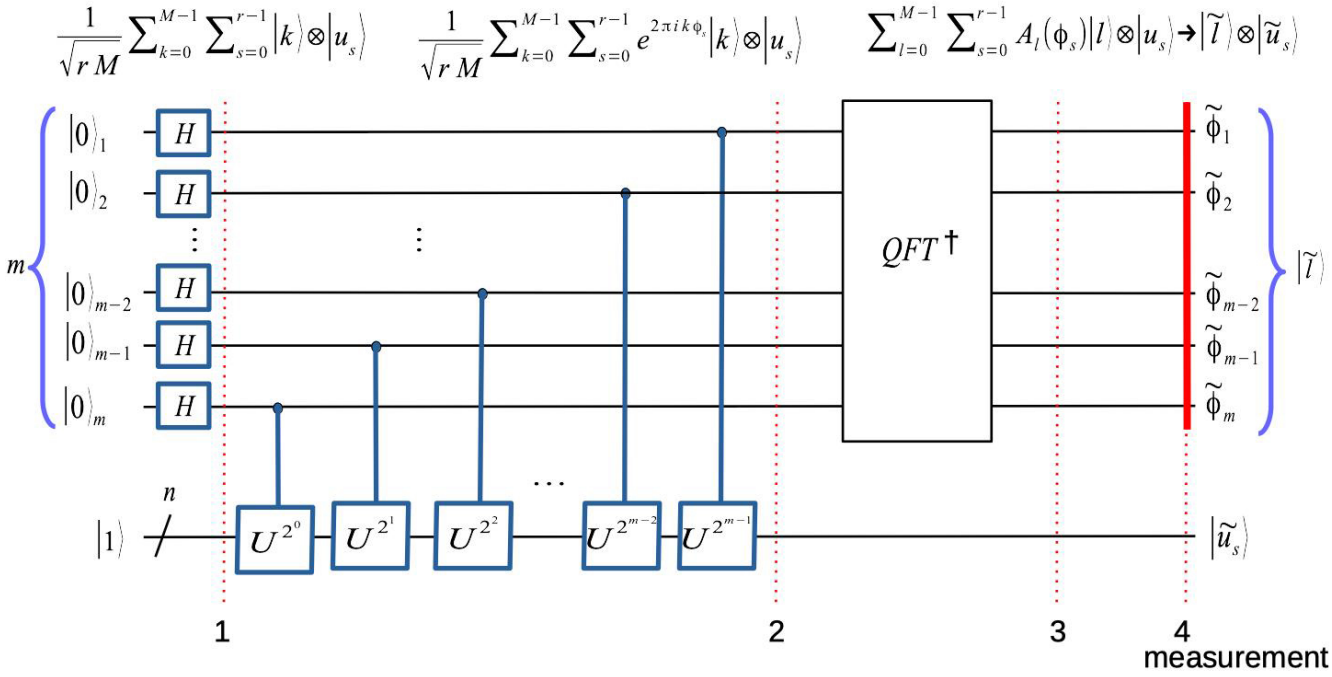


Figure 14: *Shor's factoring circuit. Convention 2. We populate the work register with the state* $|1\rangle$*, which is just a uniform linear combination of the Eigenstates* $|u_s\rangle$*. This requires no prior knowledge of these Eigenstates. Through phase kickback, the control register becomes a linear combination of terms involving the states* $|u_s\rangle$ *and their phases* ϕ_s*. Measuring the control register thereby projects the system into a state* $|\tilde{\ell}\,\rangle\otimes|\tilde{u}_s\rangle$ *for some* $\tilde{\ell}\in\{0,1,\cdots,M-1\}$ *and* $s\in\{0,1,\cdots,r-1\}$*. The control state is indexed by the m-bit integer* $\tilde{\ell}=\tilde{\phi}_1\cdots\tilde{\phi}_m$*, where each* $\tilde{\phi}_k\in\{0,1\}$ *is the measured outcome of qubit* $k\in\{1,2,\cdots,m\}$ *of the control register. We then evaluate the m-bit measured phase* $\tilde{\phi}_\ell=\tilde{\ell}/2^m=0.\tilde{\phi}_1\cdots\tilde{\phi}_m$*, from which we can extract the exact phase* $\phi_s=s/r$ *using the method of continued fractions (provided s and r are relatively prime).*

need for Shor's algorithm. We can circumvent this difficulty by employing (181), which we repeat here for emphasis:

$$|1\rangle=\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle\,. \tag{183}$$

This suggests that we populate the work register with the easily prepared state $|1\rangle=|0\cdots01\rangle$, which is just a uniform linear combination of the phase Eigenstates $|u_s\rangle$. This is achieved by initializing the lowest order qubit in the work register to the 1-state, and all other work qubits to 0-states. This new circuit is illustrated in Fig. 14. Populating the work register with a linear superposition of the states $|u_s\rangle$ has the effect of rendering the state of the control register (through phase kickback) as a linear combination of the Eigenstates $|u_s\rangle$ and their corresponding phases ϕ_s. More specifically, the state of the quantum system right after the front-end of the circuit (position 2 in the Figure) will be given by

$$|\psi_2\rangle=\frac{1}{\sqrt{rM}}\sum_{k=0}^{M-1}\sum_{s=0}^{r-1}e^{2\pi ik\,\phi_s}\,|k\rangle\otimes|u_s\rangle\,. \tag{184}$$

Finally, the $QFT^\dagger$ operation (position 3 in the Figure) transforms this state into

$$|\psi_3\rangle=\sum_{\ell=0}^{M-1}\sum_{s=0}^{r-1}A_\ell(\phi_s)\,|\ell\rangle\otimes|u_s\rangle\quad\text{with}\quad\phi_s=s/r\,, \tag{185}$$

where the amplitudes are defined by

$$A_\ell(\phi_s)\;\equiv\;\frac{1}{\sqrt{r}M}\,\frac{1-e^{2\pi i\,(\phi_s-\ell/M)M}}{1-e^{2\pi i\,(\phi_s-\ell/M)}} \tag{186}$$

$$= \frac{1}{\sqrt{r}M}\,\frac{1-e^{2\pi i\,(\phi_s-\phi_\ell)M}}{1-e^{2\pi i\,(\phi_s-\phi_\ell)}} = \frac{1}{\sqrt{r}M}\,\frac{1-e^{2\pi i\,(\ell_s-\ell)}}{1-e^{2\pi i\,(\ell_s-\ell)}/M}\,, \tag{187}$$

where $\phi_\ell \equiv \ell/M$ and $\ell_s \equiv M\phi_s$. The control register has therefore obtained knowledge of the phases $\phi_s = s/r$ and their corresponding Eigenstates $|u_s\rangle$.

Finally, we must measure the control register (position 4 in the Figure with the bold red line), which collapses the quantum state $|\psi_3\rangle$ of (185) into a definite Eigenstate state

$$|\tilde{\ell}\,\rangle \otimes |u_{\tilde{s}}\rangle\,. \tag{188}$$

In other words, the control register collapses to a state $|\tilde{\ell}\,\rangle$ labeled by the m-bit integer index $\tilde{\ell} = \tilde{\phi}_1\,\tilde{\phi}_2\cdots\tilde{\phi}_m \in \{0, 1, \cdots, M-1\}$, where the measured phase is then given by $\tilde{\phi}_\ell \equiv \tilde{\ell}/M = 0.\tilde{\phi}_1\,\tilde{\phi}_2\cdots\tilde{\phi}_m$. Likewise, the work register is projected into a state $|u_{\tilde{s}}\rangle$ determined by a random choice of the phase integer $\tilde{s} \in \{0, 1, \cdots, r-1\}$. The ME phase angle will be written $\tilde{\phi}_s = \tilde{s}/r$, and we define the corresponding mode $\tilde{\ell}_s = M\tilde{\phi}_s$. The probability of measuring the Eigenstate state (188) is therefore

$$P_{\tilde{\ell},\tilde{s}} = \left|A_{\tilde{\ell}}(\tilde{\phi}_s)\right|^2 = \frac{1}{rM^2}\,\frac{\sin^2\left[\pi\left(\tilde{\phi}_s - \dfrac{\tilde{\ell}}{M}\right)M\right]}{\sin^2\left[\pi\left(\tilde{\phi}_s - \dfrac{\tilde{\ell}}{M}\right)\right]} \tag{189}$$

$$= \frac{1}{rM^2}\,\frac{\sin^2\left[\pi(\tilde{\phi}_s - \tilde{\phi}_\ell)M\right]}{\sin^2\left[\pi(\tilde{\phi}_s - \tilde{\phi}_\ell)\right]} = \frac{1}{rM^2}\,\frac{\sin^2\left[\pi(\tilde{\ell}_s - \tilde{\ell}\,)\right]}{\sin^2\left[\pi(\tilde{\ell}_s - \tilde{\ell}\,)/M\right]}\,. \tag{190}$$

In applying Shor's algorithm it will be critical that $\tilde{\phi}_\ell$ is a rational number, and that $\tilde{s}$ and r are relatively prime. We will usually denote the state $|u_{\tilde{s}}\rangle$ by the simpler form $|\tilde{u}_s\rangle$, in which case we will drop the tilde from the ME phase and write $\phi_s = s/r$. Note that the control and work registers are entangled, and that the most likely control-output states $\tilde{\ell}$ are those for which $\tilde{\phi}_\ell \approx \phi_s$ for some $s \in \{0, 1, \cdots, r-1\}$. The rub is then to extract the *exact* phase ϕ_s from the *approximately* measured m-bit phase $\tilde{\phi}_\ell$, and in particular we must pull out the integer period r from the rational number $\tilde{\phi}_\ell$. This requires the mathematical technique of *continued fractions*, which we shall apply in the next section. If we obtain the zero-phase state $\phi_0 = 0$, or if s and r have non-trivial common factors, then the measurement will fail, and we must try again. Once we have a potential value for the period r, we must explicitly check that r is even, and that $a^r = 1 \pmod N$. We must also ensure that $a^{r/2} \neq \pm 1 \pmod N$. If these criteria are not met, then we must try again. Thus, Shor's algorithm is probabilistic in nature, but the chance of a successful run is quite high, and usually requires at most two or three attempts.

Let us now work through the circuit in Fig. 14 in more detail. First, we must prepare the zero-state

$$|0\rangle^{\otimes m} \otimes |0\rangle^{\otimes n}\,, \tag{191}$$

and then apply an X operator to the lowest order qubit of the work register, thereby producing the initial state

$$|\psi_0\rangle = |0\rangle^{\otimes m} \otimes |1\rangle\,. \tag{192}$$

We are using a short-hand notation for the states in the n-qubit work register, where we denote the computational basis states by their corresponding integer index, rather than breaking them out into their respective tensor products of single-particle qubit states. In the physics Convention 2, the explicit form of the work register would be

$$|1\rangle = |0\rangle_1 \otimes \cdots \otimes |0\rangle_{n-1} \otimes |1\rangle_n\,, \tag{193}$$

and in the Qiskit Convention 1 we would have

$$|1\rangle = |1\rangle_0 \otimes |0\rangle_1 \otimes \cdots \otimes |0\rangle_{n-1}\,. \tag{194}$$

We should also note that the phase operator U is distributed over the work register, and it takes different forms between Conventions 1 and 2.

Hadamard gates $H^{\otimes m}$ are then applied to the control register, which splits the state $|0\rangle^{\otimes m}$ into a linear superposition of all possible control states $|k\rangle$ for $k \in \{0, 1, \cdots, M-1\}$ with $M = 2^m$. Upon using (181), we can therefore express the resulting state in either of two forms,

$$|\psi_1\rangle = \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} |k\rangle \otimes |1\rangle \tag{195}$$

$$= \frac{1}{\sqrt{rM}} \sum_{k=0}^{M-1} \sum_{s=0}^{r-1} |k\rangle \otimes |u_s\rangle \,, \tag{196}$$

each of which reveals something essential about the quantum system. For example, (196) tells us that the state $|\psi_1\rangle$ is in fact just a uniform linear superposition involving the Eigenstates $|u_s\rangle$, which was crucial to the above measurement analysis. Next, the circuit operates on the state $|\psi_1\rangle$ with a sequence of m controlled-phase operators CU^p for $p \in \{2^0, 2^1, \cdots, 2^{m-1}\}$, giving the state

$$|\psi_2\rangle = \frac{1}{\sqrt{r\,M}} \sum_{k=0}^{M-1} \sum_{s=0}^{r-1} e^{2\pi i k\,\phi_s} |k\rangle \otimes |u_s\rangle \quad \text{for} \quad \phi_s = \frac{s}{r} \,. \tag{197}$$

By employing (182), we can express this state in the alternative form

$$|\psi_2\rangle = \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} |k\rangle \otimes \left(\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{2\pi i\, k\phi_s} |u_s\rangle \right) \tag{198}$$

$$= \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} |k\rangle \otimes |a^k \pmod N\rangle = \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} |k\rangle \otimes |f(k)\rangle \,. \tag{199}$$

Recall that the period of the function $f(x) = a^x \pmod N$ is r, and we therefore find the following sequence of operations on the state $|1\rangle$:

$$\begin{aligned}
U^1|1\rangle &= |a \pmod N\rangle \\
U^2|1\rangle &= U|a \pmod N\rangle = |a^2 \pmod N\rangle \\
&\cdots \\
U^k|1\rangle &= U|a^{k-1} \pmod N\rangle = |a^k \pmod N\rangle \\
&\cdots \\
U^{r-1}|1\rangle &= U|a^{r-2} \pmod N\rangle = |a^{r-1} \pmod N\rangle \\
U^r|1\rangle &= U|a^{r-1} \pmod N\rangle = |a^r \pmod N\rangle = |1\rangle \\
U^{r+1}|1\rangle &= U|a^r \pmod N\rangle = |a^{r+1} \pmod N\rangle = |a \pmod N\rangle \\
U^{r+2}|1\rangle &= U|a^{r+1} \pmod N\rangle = |a^{r+2} \pmod N\rangle = |a^2 \pmod N\rangle \\
&\cdots \quad ,
\end{aligned} \tag{200}$$

which can be summarized by

$$U^x|1\rangle = |f(x)\rangle \;\text{ and }\; |f(x+1)\rangle = U\,|f(x)\rangle \tag{201}$$

for any non-negative integer x. We shall use these relations later in the exposition when constructing the ME operators U. Note that we can express the front-end output state from (199) in the form

$$|\psi_2\rangle = \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} |k\rangle \otimes U^k\,|1\rangle \,, \tag{202}$$

which shows that the ME operators between point-1 and point-2 of Fig. 14 have the effect of augmenting the work register by $U^k|1\rangle$ for every mode $k \in \{0, 1, \cdots, M-1\}$. We have almost finished analyzing the circuit. Now that

we have state $|\psi_2\rangle$ of (197) in hand, we apply the inverse quantum Fourier transform $QFT^\dagger$ to the control register, producing the final state

$$|\psi_3\rangle \quad = \quad \sum_{\ell=0}^{M-1} \sum_{s=0}^{r-1} A_\ell(\phi_s)\, |\ell\rangle \otimes |u_s\rangle \,, \tag{203}$$

where the amplitudes $A_\ell(\phi_s)$ are given by (186). We now have a linear superposition of the states $|\ell\rangle \otimes |u_s\rangle$, and as described above, a measurement on the control register will produce any one of them,

$$\sum_{\ell=0}^{M-1} \sum_{s=0}^{r-1} A_\ell(\phi_s)\, |\ell\rangle \otimes |u_s\rangle \to |\tilde{\ell}\rangle \otimes |\tilde{u}_s\rangle \tag{204}$$

with probability $P_{\tilde{\ell},s} = |A_{\tilde{\ell}}(\phi_s)|^2$. The measured phase $\tilde{\phi}_\ell$ is given in terms of the measured output index $\tilde{\ell}$ of the control register by $\tilde{\phi}_\ell = \tilde{\ell}/2^m$, and Shor's algorithm is designed to give dominant peaks close to the exact phases $\phi_s = s/r$. We will have more to say about the details of this procedure in later sections. For now, we summarize the action of the Shor circuit in Table 1. Note that when the exact phases $\phi_s = s/r$ can be represented by m-bit fractions, the phase histogram simplifies considerably: There are exactly r equally likely peaks, each corresponding to one of the phases ϕ_s. This situation is relatively rare, and does not occur in most cases, thereby allowing for more complex phase histograms.

Table 1: *Quantum Period Finding*

1. Initialize the state:
   Prepare the state $|0\rangle^{\otimes m} \otimes |0\rangle^{\otimes n}$, and apply X to the lowest order qubit of the work register to produce the initial state

$$|\psi_0\rangle \quad = \quad |0\rangle^{\otimes m} \otimes |1\rangle \,, \tag{205}$$

   where the number of work qubits is given by $n = \lceil \log_2 N \rceil$.

2. Randomize the control register:
   Apply $H^{\otimes m}$ to the control register to give

$$|\psi_1\rangle \quad = \quad \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} |k\rangle \otimes |1\rangle \tag{206}$$

$$\quad = \quad \frac{1}{\sqrt{rM}} \sum_{k=0}^{M-1} \sum_{s=0}^{r-1} |k\rangle \otimes |u_s\rangle \,, \tag{207}$$

   where the total number of quantum states is $M = 2^m$.

3. Modular exponentiation (ME):
   Conditionally apply the ME operators U^p for $p \in \{2^0, 2^1, \cdots, 2^{m-1}\}$ successively to the control qubits to produce the state

$$|\psi_2\rangle \quad = \quad \frac{1}{\sqrt{r\,M}} \sum_{k=0}^{M-1} \sum_{s=0}^{r-1} e^{2\pi i k\, \phi_s}\, |k\rangle \otimes |u_s\rangle \quad \text{where} \quad \phi_s = \frac{s}{r} \tag{208}$$

$$\quad = \quad \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} |k\rangle \otimes |a^k \pmod N\rangle \,. \tag{209}$$

4. Perform the inverse Fourier transform:

$$|\psi_3\rangle \quad = \quad QFT^{\dagger}|\psi_2\rangle = \sum_{\ell=0}^{M-1}\sum_{s=0}^{r-1} A_\ell(\phi_s)\,|\ell\rangle \otimes |u_s\rangle\,, \tag{210}$$

where the amplitudes $A_\ell(\phi_s)$ are given by (186).

5. Perform a measurement of the control register:
The state then collapses into an Eigenstate,

$$\sum_{\ell=0}^{M-1}\sum_{s=0}^{r-1} A_\ell(\phi_s)\,|\ell\rangle \otimes |u_s\rangle \rightarrow |\tilde{\ell}\,\rangle \otimes |\tilde{u}_s\rangle\,, \tag{211}$$

with probability $P_{\tilde{\ell},s} = |A_{\tilde{\ell}}(\phi_s)|^2$. The probability peaks at values of $\tilde{\ell}$ close to the exact phases $\phi_s = s/r$, and the outcomes of the measurement are equally distributed between the values of $s \in \{0, 1, \cdots, r-1\}$.

6. Apply the method of continued fractions to extract the exact phase $\phi_s = s/r$ from the approximately measured m-bit phase $\tilde{\phi}_\ell$. This provides the exact period r, and therefore the factors of N.

5.4. Extracting the Exact Period from the Measured Phase

We turn now to the (non-trivial) task of extracting the *exact* phase $\phi_s = s/r$ from the *approximately* measured phase $\tilde{\phi}_\ell$ of the control resister using the method of *continued fractions* outlined in Section 4. For ease of notation, we shall henceforth drop the superfluous ℓ-subscript from $\tilde{\phi}_\ell$ and simply write $\tilde{\phi}$. Since the measured phase $\tilde{\phi}$ is a positive rational number less than one, it can be expressed as a finite continued fraction of the form

$$\tilde{\phi} = \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \ddots + \frac{1}{a_R}}}} \tag{212}$$

for some integer R, where $a_1, a_2, \cdots, a_R$ are all positive integer coefficients. We shall drop the zero digit $a_0 = 0$, and simply write $\tilde{\phi} = [a_1, a_2, \cdots, a_R]$ to denote the form of the continued fraction. Suppose now that s and r are two relatively prime positive integers that satisfy the inequality

$$\left|\frac{s}{r} - \tilde{\phi}\right| \leq \frac{1}{2r^2}\,. \tag{213}$$

Then by Theorem 3 of the Section 4, the ratio s/r is necessarily a *convergent* of the continued fraction (212) for $\tilde{\phi}$, so that $s/r = [a_1, a_2, \cdots, a_q]$ for some $q \leq R$. We shall exploit this fact to extract the exact period r of the modular exponential function $f_{a,N}(x) = a^x \pmod N$.

Continued fractions and their convergents can be calculated efficiently using a variation of the Euclidean algorithm for finding the greatest common divisor of two integers. Let us call the convergents of $\tilde{\phi}$ by $s_0/r_0, s_1/r_1, \cdots, s_\ell/r_\ell, \cdots, s_q/r_q$. We then cycle through these convergents, from the smallest to the largest values of the r's, checking to ensure that

$$r \text{ is even} \tag{214}$$

$$a^{r/2} \neq \pm 1 \pmod N \tag{215}$$

$$a^r = 1 \pmod N \tag{216}$$

for each $r = r_\ell$. Condition (215) simply means that $b = a^{r/2}$ is not a trivial root of unity. In passing, we note that r can in fact be odd, provided that a is a *perfect square*, so that $b = a^{r/2}$ is still an integer [11]. Apart from this caveat, if any of the conditions (214)–(216) are not met for r_ℓ, then the trial fails, and we move on to the next convergent. The special cases $s_\ell = 0$ and $s_\ell = 1$ correspond to the phases $\phi_0 = 0/r_\ell = 0$ and $\phi_1 = 1/r_\ell$. Technically, we cannot use the method of continued fractions for these cases since $0/r_\ell$ and $1/r_\ell$ are not the ratio of two primes (as 0 and 1

are not prime). In the former case, $\phi_0 = 0$ yields no information, and we must move on to the next iteration. However, $\phi_1 = 1/r_\ell$ yields an integer r_ℓ, which could in principle be the correct period that we are seeking. Therefore we shall check the case $s_\ell = 1/r_\ell$ just to make sure, even though $1/r_\ell$ does not satisfy the conditions of Theorem 3. In any event, a detailed analysis shows that the probability of achieving a solution after just a few trials is quite high. The solution for the smallest value of r_ℓ gives the period $r = r_\ell$ that we are seeking, and the factors of N are then given by $\gcd(a^{r/2} \pm 1, N)$.

In closing this section, we should call attention to an important feature of the method. We must somehow ensure that inequality (213) always holds, as it would be quite cumbersome if we had to check this by hand every time. However, this requirement can be hard-wired into the algorithm itself by choosing an appropriate number of qubits m for the control register, one that is based on the inequality (213) itself. Recall that the work register has $n = \lceil \log_2 N \rceil$ qubits, and thus for any phase $\phi_s = s/r$, we have $r < N \leq 2^n$. This leads to the inequality

$$\frac{1}{2^{2n+1}} \leq \frac{1}{2r^2} \,. \tag{217}$$

Therefore, if we take the control register to have $m = 2n+1$ qubits, then a measurement of the phase $\tilde{\phi}$ will necessarily have sufficient precision to ensure that

$$\left| \frac{s}{r} - \tilde{\phi} \right| = \left| \phi_s - \tilde{\phi} \right| \leq \frac{1}{2^{2n+1}} \leq \frac{1}{2r^2} \,, \tag{218}$$

and then inequality (213) is automatically satisfied. This is the source of our previous requirement that $m = 2n + 1$. Of course we must use even more control qubits to account for machine error. For a probability of success at least as large as $1 - \varepsilon$, where $\varepsilon > 0$ is a small probability of failure, we must use $m = 2n + 1 + n_\varepsilon$ qubits in the control register, where $n_\varepsilon \equiv \lceil \log_2(2 + 1/(2\varepsilon)) \rceil$. With this choice of m, upon taking a measurement of the control register we find: (i) inequality (213) will always be satisfied, (ii) the exact phase will be of the form $\phi_s = s/r$, where $s \in \{0, 1, \cdots, r-1\}$ is randomly selected, and (iii) the ratio s/r will be a convergent of the continued fraction for $\tilde{\phi}$ (provided that s and r are relatively prime). However, we should emphasize that conditions (214)–(216) must also be satisfied. If they are not, then the method will fail, and we must move on to another iteration of Shor's algorithm. However, a complete error analysis shows that the probability of success is quite high, and typically only a few iterations will be required before obtaining a factor. We summarize Shor's algorithm in Table 2.

Table 2: *Shor's Factorization Algorithm*

1. Chose a random number $a \in \{2, 3, \cdots, N-1\}$ for the base. If $\gcd(a, N) = 1$, then proceed to the next step (otherwise we have found a non-trivial factor of N as required).

2. Use quantum phase estimation (QPE) to measure the phase $\tilde{\phi}$ of the modular exponentiation operator U_{aN} defined by
$$U_{aN}|w\rangle = |a \cdot w \ (\mathrm{mod}\ N)\rangle \ \text{ with } \ U_{aN}|u_s\rangle = e^{2\pi i\, s/r}\,|u_s\rangle \,,$$
where $s \in \{0, 1, \cdots, r-1\}$. The QPE is the only quantum component of Shor's algorithm. This is also the bottleneck of the algorithm, as (i) most of the quantum resources are deployed here, and (ii) a different operator U_{aN} is required for every choice of a and N.

3. We then use the method of continued fractions to extract the *exact* period r from the *approximately* measured phase $\tilde{\phi}$. To do this, we examine all convergents $\phi_s = s/r$ of $\tilde{\phi}$ such that
$$\left| \frac{s}{r} - \tilde{\phi} \right| \leq \frac{1}{2r^2} \,,$$
which is achieved by requiring the number of control qubits to be $m = 2n + 1$. We then check the convergents from the smallest to the largest values of r. If r is odd, then return to step 1. If $a^{r/2} = \pm 1 \pmod N$, then return to step 1. If $a^r = 1 \pmod N$, then we have found a solution for r, and we proceed to the next step; otherwise return to step 1.

4. Factors of N are now given by $\gcd(a^{r/2} \pm 1, N)$. Finding the greatest common divisor can be done very efficiently on a classical computer using Euclid's algorithm.

6. Verifying the Theory: Factoring N = 15

To highlight the principal aspects of Shor's algorithm, we now build the computational machinery to factor the number $N = 15$. In the next section we will employ the scripts developed here to factor larger and more complex numbers. We shall employ IBM's circuit simulator Qiskit. This means that we must use the Qiskit qubit ordering convention in which the upper 0-th qubit corresponds to the lowest order bit. We developed the Shor factorization circuit in Section 5.3, which is illustrated in Fig. 14. However, this analysis used the physics and mathematics ordering convention rather than the Qiskit convention. Consequently, we must convert to the Qiskit ordering displayed in Fig. 15. In this convention, the measurement of the control register (position 4 in the Figure, and indicated by the bold red line across the register) gives the output state $|\tilde{\ell}\rangle = |\tilde{\phi}_{m-1} \cdots \tilde{\phi}_1 \, \tilde{\phi}_0\rangle$. Just as in the previous section, we will employ the convention in which the *measured* (m-bit) phase in the control register is written with a tilde, and where the output state is indexed by the binary integer

$$\tilde{\ell} \equiv \tilde{\phi}_{m-1} \, \tilde{\phi}_{m-2} \cdots \tilde{\phi}_1 \, \tilde{\phi}_0 \quad \text{where} \quad \tilde{\phi}_k \in \{0, 1\} \tag{219}$$

$$= 2^{m-1} \, \tilde{\phi}_{m-1} + 2^{m-2} \, \tilde{\phi}_{m-2} + \cdots + 2^1 \, \tilde{\phi}_1 + 2^0 \, \tilde{\phi}_0 \, . \tag{220}$$

One must keep in mind that there is always an implicit m-bit resolution associated with any measured quantity in the control register, and in particular, the corresponding angular phase is given by the m-bit fraction

$$\tilde{\phi}_\ell \equiv \frac{\tilde{\ell}}{2^m} = \frac{\tilde{\phi}_{m-1}}{2^1} + \frac{\tilde{\phi}_{m-2}}{2^2} + \cdots + \frac{\tilde{\phi}_1}{2^{m-1}} + \frac{\tilde{\phi}_0}{2^m} \tag{221}$$

$$= 0.\tilde{\phi}_{m-1} \, \tilde{\phi}_{m-2} \cdots \tilde{\phi}_1 \, \tilde{\phi}_0 \, . \tag{222}$$

For ease of notation, we shall drop the ℓ-subscript and denote the measured phase by $\tilde{\phi}$.

Since we will employ both binary and decimal numbers in this section, we will often denote binary numbers using a bracket with a 2-subscript, writing $\tilde{\phi} = [0.\tilde{\phi}_{m-1} \, \tilde{\phi}_{m-2} \cdots \tilde{\phi}_1 \, \tilde{\phi}_0]_2$. We will always place a tilde over measured quantities, so that $\tilde{\phi}$ denotes the phase as determined by a measurement of the m-bit control register, as opposed to the (as yet undetermined) exact phase $\phi_s = s/r$. If the control register contains a sufficient number of qubits, then the measurement should be quite accurate, and the measured phase will be very close to the exact phase, $\tilde{\phi} \approx \phi_s = s/r$. Since $\tilde{\phi}$ and $\phi_s = s/r$ are approximately equal, we can extract the *exact* value of the integers s and r using the method of continued fractions (provided that s and r are relatively prime). The integer r is the period of the modular exponential function $f_{a\,N}(x) = a^x \pmod N$ that we seek.

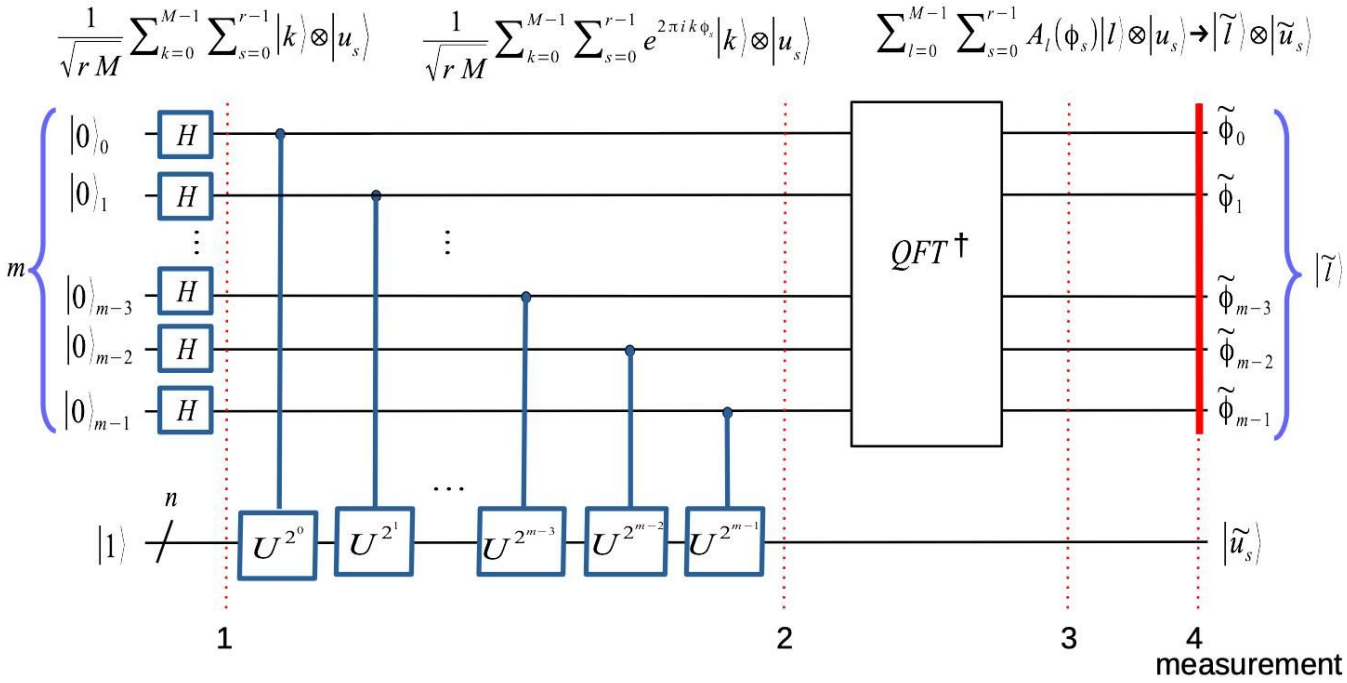


Figure 15: *Shor factorization algorithm for Qiskit convention 1. In comparison, Fig. 14 uses the physics Convention 2.*

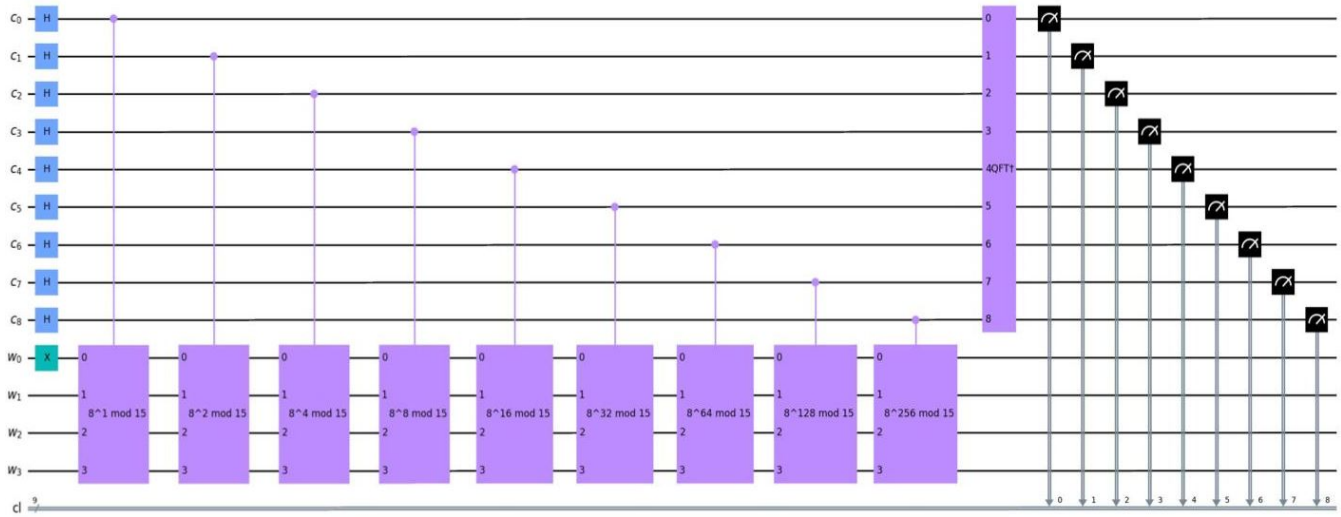


Figure 16: *Qiskit version of Shor's factoring circuit for $N = 15$ and $a = 8$, with $m = 9$ qubits in the control register and $n = 4$ qubits in the work register.*

Let us now concentrate on the specific example of $N = 15$. As we have established, the work space requirement is $n = \lceil \log_2 15 \rceil = 4$ qubits, and the control register must therefore contain $m = 2n + 1 = 9$ qubits (for simplicity we consider only perfect measurements in which $n_\epsilon = 0$). For a given integer N, we choose the base a to be a random integer such that $1 < a < N$ and $\gcd(a, N) = 1$. Therefore, for $N = 15$ we can only choose $a \in \{2, 4, 7, 8, 11, 13, 14\}$. It turns out that $a = 14$ gives a trivial root of unity, so we can neglect this choice. In fact, in this example we shall take either $a = 4$ or $a = 8$, where the former gives the period $r = 2$ and the latter gives $r = 4$. Figure 16 illustrates the Qiskit circuit for Shor's algorithm with the base $a = 8$. The first 9 qubits comprise the control register and are labeled by the index c, and the last 4 qubits are the work register and are labeled by w. We will therefore denote quantum states of the work register by $|w_3 w_2 w_1 w_0\rangle$. In the Qiskit qubit convention, the work register is initially populated by the state

$$|1\rangle = |0001\rangle = |1\rangle_0 \otimes |0\rangle_1 \otimes |0\rangle_2 \otimes |0\rangle_3 \,, \tag{223}$$

with the least significant bit being $w_0 = 1$. The controlled modular exponentiation operators $CU^p_{8,15}$ for $p \in \{2^0, 2^1, \cdots, 2^8\}$ are represented by purple boxes attached to their respective control qubits, and the operator $QFT^\dagger$ is represented by the large purple rectangle on the far right of the control register. At the end of the circuit, the control register undergoes a measurement on all m qubits. The work register might or might not undergo a measurement, and we shall return to this point later in the section.

6.1. Modular Exponentiation Operators

Let us now explore the modular exponentiation (ME) operators $U_{a\,N}$ in more detail. For every choice of the number N and the base a, we must design a separate implementation of the operator $U_{a\,N}$, and this is in fact the real bottleneck of Shor's algorithm. Indeed, this bottleneck occurs in two senses: (i) the ME operators consume the greatest majority of the quantum resources of the algorithm, and in the general case this will be of order $72\,n^3$ gates [5], and (ii) even specialized cases of the ME operators are often highly non-trivial to construct. The ME operators for $N = 15$ with $a = 4$ and $a = 8$ are illustrated in Fig. 17, and Table 3 gives the corresponding operations $U_{a,15}|w\rangle = |a \cdot w \ (\mathrm{mod}\ 15)\rangle$ for every basis element $|w\rangle$ in the work register. Each ME operator $U_{a,15}$ has two columns in the Table: One for the decimal representation of the basis elements $|w\rangle$, and another for the corresponding binary representation $|w_3 w_2 w_1 w_0\rangle$, where in binary form $w = [w_3 w_2 w_1 w_0]_2$. The operators $U_{a,15}$ in Fig. 17 were determined through simple inspection of the results of the binary columns in Table 3. For example, the $a = 8$ operator performs a permutation of the 4-bit binary work states, $U_{8,15}|w_3 w_2 w_1 w_0\rangle = |w_0 w_3 w_2 w_1\rangle$, which can be implemented by the three SWAP gates in the right panel of Fig. 17. Similarly, the ME operator for $a = 4$ performs two SWAP operations on the work register, so that $U_{4,15}|w_3 w_2 w_1 w_0\rangle = |w_1 w_0 w_3 w_2\rangle$. The other ME operators for $a \in \{2, 7, 11, 13\}$ act similarly, and are illustrated in Fig. 18.

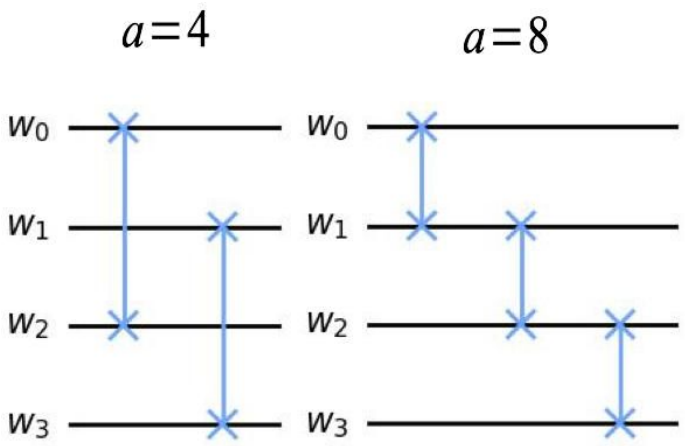


Figure 17: *Modular exponentiation operators $U_{a,15}$ for $a = 4$ in the left panel and $a = 8$ in the right panel. Their action on the work space is given by $U_{4,15}|w_3w_2w_1w_0\rangle = |w_1w_0w_3w_2\rangle$ and $U_{8,15}|w_3w_2w_1w_0\rangle = |w_0w_3w_2w_1\rangle$, which can be reproduced by a sequence of SWAP gates.*

Table 3: *Modular Exponentiation Operators $U = U_{a,15}$ for $a = 4$ and $a = 8$.*

$U_{4,15}\|w\rangle = \|4 \cdot w \pmod{15}\rangle$			$U_{8,15}\|w\rangle = \|8 \cdot w \pmod{15}\rangle$		
$U\|1\rangle = \|4\rangle$	$U\|0001\rangle = \|0100\rangle$		$U\|1\rangle = \|8\rangle$	$U\|0001\rangle = \|1000\rangle$	
$U\|2\rangle = \|8\rangle$	$U\|0010\rangle = \|1000\rangle$		$U\|2\rangle = \|1\rangle$	$U\|0010\rangle = \|0001\rangle$	
$U\|3\rangle = \|12\rangle$	$U\|0011\rangle = \|1100\rangle$		$U\|3\rangle = \|9\rangle$	$U\|0011\rangle = \|1001\rangle$	
$U\|4\rangle = \|1\rangle$	$U\|0100\rangle = \|0001\rangle$		$U\|4\rangle = \|2\rangle$	$U\|0100\rangle = \|0010\rangle$	
$U\|5\rangle = \|5\rangle$	$U\|0101\rangle = \|0101\rangle$		$U\|5\rangle = \|10\rangle$	$U\|0101\rangle = \|1010\rangle$	
$U\|6\rangle = \|9\rangle$	$U\|0110\rangle = \|1001\rangle$		$U\|6\rangle = \|3\rangle$	$U\|0110\rangle = \|0011\rangle$	
$U\|7\rangle = \|13\rangle$	$U\|0111\rangle = \|1101\rangle$		$U\|7\rangle = \|11\rangle$	$U\|0111\rangle = \|1011\rangle$	
$U\|8\rangle = \|2\rangle$	$U\|1000\rangle = \|0010\rangle$		$U\|8\rangle = \|4\rangle$	$U\|1000\rangle = \|0100\rangle$	
$U\|9\rangle = \|6\rangle$	$U\|1001\rangle = \|0110\rangle$		$U\|9\rangle = \|12\rangle$	$U\|1001\rangle = \|1100\rangle$	
$U\|10\rangle = \|10\rangle$	$U\|1010\rangle = \|1010\rangle$		$U\|10\rangle = \|5\rangle$	$U\|1010\rangle = \|0101\rangle$	
$U\|11\rangle = \|14\rangle$	$U\|1011\rangle = \|1110\rangle$		$U\|11\rangle = \|13\rangle$	$U\|1011\rangle = \|1101\rangle$	
$U\|12\rangle = \|3\rangle$	$U\|1100\rangle = \|0011\rangle$		$U\|12\rangle = \|6\rangle$	$U\|1100\rangle = \|0110\rangle$	
$U\|13\rangle = \|7\rangle$	$U\|1101\rangle = \|0111\rangle$		$U\|13\rangle = \|14\rangle$	$U\|1101\rangle = \|1110\rangle$	
$U\|14\rangle = \|11\rangle$	$U\|1110\rangle = \|1011\rangle$		$U\|14\rangle = \|7\rangle$	$U\|1110\rangle = \|0111\rangle$	

6.2. The Factorization Circuit

Our next goal is to construct the Python script used in creating the Shor circuit in Fig. 16, after which we shall run this code on the Qiskit simulator to factor $N = 15$. We will construct the requisite script slowly in sequences of Python code segments, each one illustrating an essential element of the algorithm. First, we must import the necessary python packages:

```
# import basics
import numpy as np
from random import randint
from math import gcd

# import Qiskit tools
from qiskit import Aer, transpile, assemble
from qiskit import QuantumCircuit, ClassicalRegister, QuantumRegister

# import plot tools
from qiskit.visualization import plot_histogram
import matplotlib.pyplot as plt
```

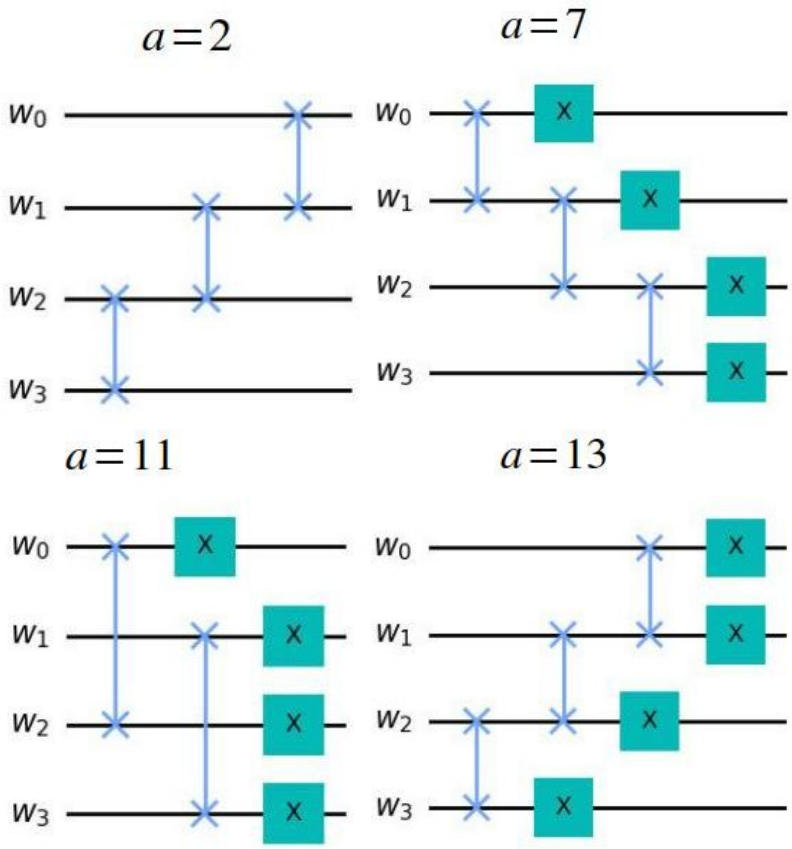


Figure 18: *Modular exponentiation operators $U_{a,15}$ for $a = 2, 7, 11, 13$ as labeled.*

The first few imports are for basic mathematical functionality. For example, we can create random integers with `randint()`, and find the greatest common divisor with `gcd()`. We have also imported packages to run Qiskit, and for post-processing the Qiskit data. We now select $N = 15$ and the corresponding base a:

```
# number to factor
N = 15

# random number a in [2,N-1] wtih gcd(a,N)=1
n = 0
while n == 0:
    a = randint(2, N-1)
    if gcd(a, N) == 1: n = 1
    print("**:", a, N, gcd(a, N))
```

This code segment chooses a random integer between 2 and $N - 1$ inclusive, and makes sure that the choice does not contain a non-trivial factor in common with N (otherwise we have found a sought after factor of N). This piece of code can be omitted if we wish to work only with a specific value of a (provided of course that we set the values of N and a here).

The next code segment defines the ME operators $CU^p_{a,N}$ for $N = 15$ for all permissible choices of a. This is really the heart of Shor's algorithm. For a general value of N, we would not be able to implement *all* values of a (we do this here only for purposes of illustration), as there are an exponentially large number of them. Finally, we define a subroutine for the inverse Fourier transform $QFT^\dagger$. The code is given below:

```
# modular exponentiation gates: p = 2^0, 2^1, .... , 2^(m-1)
def c_Uamod15(a, p):
    U = QuantumCircuit(4)
    # concatenate U-factors to form U^p
    for iteration in range(p):
        if a in [2,13]:
            U.swap(0,1)
            U.swap(1,2)
            U.swap(2,3)
        if a in [7,8]:
            U.swap(2,3)
            U.swap(1,2)
            U.swap(0,1)
        if a in [4, 11]:
            U.swap(1,3)
            U.swap(0,2)
        if a in [7,11,13]:
            for q in range(4):
                U.x(q)
    U = U.to_gate()
    U.name = "{0}^{1} mod {2}".format(a, p, N)
    c_U = U.control()
    return c_U

# inverse QFT
def qft_dagger(n):
    qc = QuantumCircuit(n)
    for q in range(n//2):
        qc.swap(q, n-q-1)
    for j in range(n):
        for m in range(j):
            qc.cp(-np.pi/float(2**(j-m)), m, j)
        qc.h(j)
    qc.name = "QFT†"
    return qc
```

Next we construct the quantum circuit itself. We must set the work register to 4 qubits and the control register to 9 qubits. We must also apply a Hadamard gate to every qubit in the control register, and we must populate the work register with the state $|1\rangle$ (using the Qiskit conventions). We then construct the ME gates to form the operators CU^p for the powers $p \in \{2^0, 2^1, \cdots, 2^8\}$. Finally, we perform the inverse QFT operation, after which we make the final measurements on the control register. We also draw the circuit and save it as a JPG file. This leads to the following code segment:

```
# Initialize registers and the quantum circuit
n_work = 4 # L
n_control = 2 * n_work + 1 # 2*L+1
c = QuantumRegister(n_control, name='c')
w = QuantumRegister(n_work, name='w')
cl  = ClassicalRegister(n_control, name='cl')
qc = QuantumCircuit(c, w, cl)

# Initialize control qubits
for q in range(n_control):
```

```
    qc.h(q)

# Populate work register with state |1>
qc.x(n_control)

# Controlled-U^p operations formed by concatenation
for k in range(n_control):
    qc.append(c_Uamod15(a, 2**k),
              [k] + [i+n_control for i in range(n_work)])

# Inverse-QFT
qc.append(qft_dagger(n_control), range(n_control))

# Measure control register
qc.measure(c, cl)
qc.draw(fold=-1)
plt.savefig('circuit_{0}.jpg'.format(a))
plt.show()
```

In constructing the modular exponentiation operators U^p for $p \in \{2^1, 2^2, \cdots, 2^8\}$, we have simply concatenated the operator U. This procedure will not do for general values of m, as it leads to an exponentially large number of gates. For a general m, we must produce m *distinct* operators U^p for each power $p \in \{2^0, 2^1, \cdots, 2^{m-1}\}$. This reduces the gate count to a polynomial order, and we will have more to say about this in the next section. In any event, this is the Qiskit code that produced Fig. 16, and it is adequate for any (small-ish) N, assuming of course that we modify the ME operators `c_Uamod15` accordingly. Finally, we must run the circuit on the Aer simulator (the QASM simulator has been deprecated):

```
# simulate
aer_sim = Aer.get_backend('aer_simulator')
t_qc = transpile(qc, aer_sim)
obj = assemble(t_qc)
results = aer_sim.run(obj, shots=1024).result()
counts = results.get_counts()
plot_histogram(counts, title='N = {0} a = {1}'.format(N, a), figsize=(6,8))
plt.savefig('hist_{0}.jpg'.format(a))
plt.show()
```

Figure 19 illustrates the output phase histogram of an ensemble of 1024 Qiskit runs for both $a = 4$ and $a = 8$. The histograms count the output measurements of the control register, and they consist of a series of well defined peaks at specific (binary integer) values $\tilde{\ell} = [\tilde{\phi}_{m-1}\tilde{\phi}_{m-2}\cdots\tilde{\phi}_1\tilde{\phi}_0]_2$. For example, $a = 4$ gives two peaks in the histogram, while $a = 8$ gives four peaks. From the peak values $\tilde{\ell}$, we then construct the measured phases $\tilde{\phi} = \tilde{\ell}/2^m = [0.\tilde{\phi}_{m-1}\tilde{\phi}_{m-2}\cdots\tilde{\phi}_1\tilde{\phi}_0]_2$, from which the exact period r can be extracted by the method of continued fractions.

6.3. The Spectrum for a = 4

Let us examine the $a = 4$ histogram in the left panel of Fig. 19. Recall that the control register has $m = 9$ qubits, and therefore the binary integers $\tilde{\ell}$ are 9 bits long. We see that there are two peaks at locations

$$\begin{aligned} \tilde{\ell}_0 &= [000000000]_2 = 0 \\ \tilde{\ell}_1 &= [100000000]_2 = 2^8 = 256\,, \end{aligned} \tag{224}$$

where we are now using subscripts to distinguish the different measurements. It is of course no accident that there are two peaks, as the modular exponential function $f_{4,15}(x)$ has a period $r = 2$, as verified by the upper-left panel of

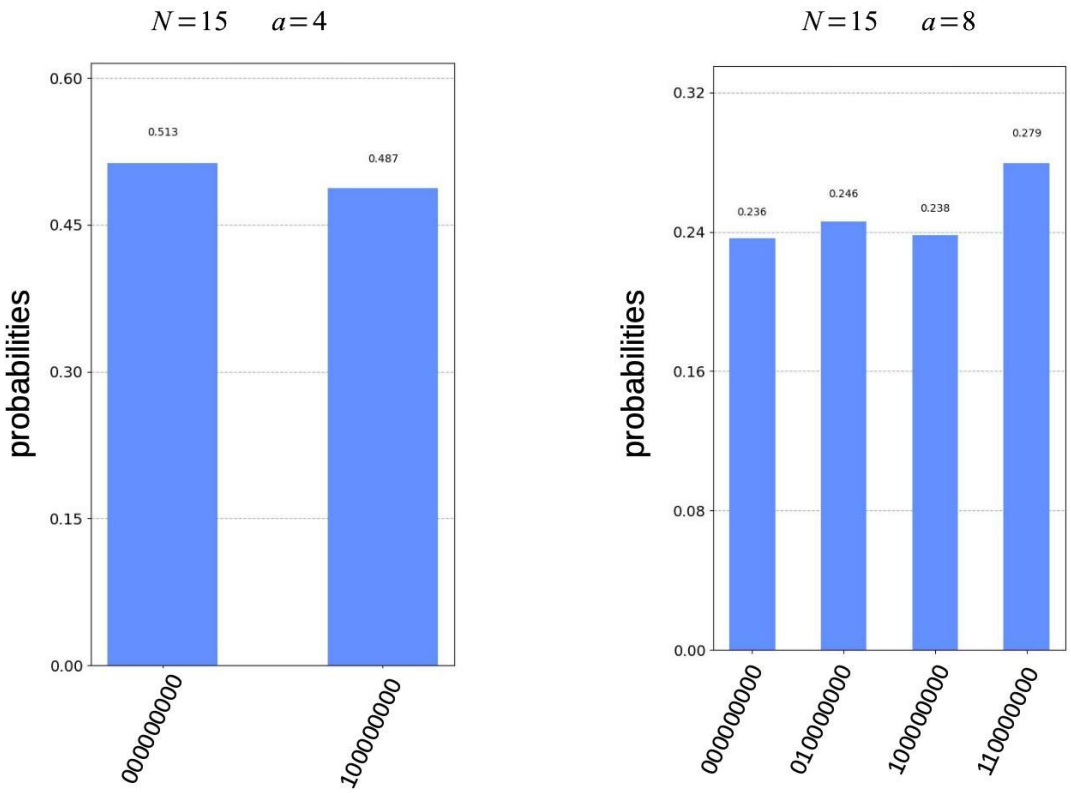


Figure 19: *The phase histograms for $N = 15$. The left and right panels show the output of Shor's algorithm for two Qiskit simulations in which the control register has $m = 9$ qubits and the work register has $n = 4$ qubits. The results for the base $a = 4$ are illustrated in the left panel, while the $a = 8$ simulation is shown in the right panel. The histograms peak at specific evenly spaced values $\tilde{\ell} = [\tilde{\phi}_{m-1} \cdots \tilde{\phi}_0]_2$, and the corresponding phases are given by $\tilde{\phi} = \tilde{\ell}/2^m = [0.\tilde{\phi}_{m-1} \cdots \tilde{\phi}_0]_2$, where $m = 9$. The period r of the modular exponential function $f_{aN}(x) = a^x \ (mod\ N)$ is encoded in the phase $\tilde{\phi} \approx \phi_s = s/r$ for $s \in \{0, 1, \cdots, r-1\}$.*

Fig. 11. The peaks $\tilde{\ell}_n$ (for $n = 0, 1$) correspond to positive (and rational) phase angles $\tilde{\phi}_n = \tilde{\ell}_n/2^m$, which take the values

$$\begin{aligned} \tilde{\phi}_0 &= [0.000000000]_2 = 0 \\ \tilde{\phi}_1 &= [0.100000000]_2 = 1/2\,. \end{aligned} \tag{225}$$

These are the measured phases of the ME operator $U_{4,15}$. Since the control register consists of 9 qubits, and all bits except the most significant bit are zero, the measurements can be regarded as exact. Consequently, there is no need to employ continued fractions for this example. The first peak at $\tilde{\phi}_0 = 0$ is guaranteed *not* to provide a factor, so we move on to the second peak at $\tilde{\phi}_1 = 1/2$. This gives an even period of $r = 2$, so that condition (214) is met. Furthermore, conditions (215) and (216) are also satisfied, since

$$a^{r/2} \ (\text{mod}\ N) = 4^1 \ (\text{mod}\ 15) = 4 \neq \pm 1 \ (\text{mod}\ 15) \tag{226}$$

$$a^r \ (\text{mod}\ N) = 4^2 \ (\text{mod}\ 15) = 16 \ (\text{mod}\ 15) = 1\,. \tag{227}$$

The factors of $N = 15$ are therefore given by $gcd(a^{r/2} - 1, N) = \text{gcd}(3, 15) = 3$ and $\text{gcd}(a^{r/2} + 1, N) = \text{gcd}(5, 15) = 5$.

6.4. The Spectrum for a = 8

We now turn to the $a = 8$ phase histogram in the right panel of Fig. 19. Again, upon changing subscript notation slightly, there are four peaks at locations

$$\begin{aligned} \tilde{\ell}_0 &= [000000000]_2 = 0 \\ \tilde{\ell}_1 &= [010000000]_2 = 128 \end{aligned}$$

$$\begin{aligned} \tilde{\ell}_2 &= [100000000]_2 = 256 \\ \tilde{\ell}_3 &= [110000000]_2 = 384\,, \end{aligned} \tag{228}$$

which correspond to the phase angles

$$\begin{aligned} \tilde{\phi}_0 &= [0.000000000]_2 = 0 \\ \tilde{\phi}_1 &= [0.010000000]_2 = 1/4 \\ \tilde{\phi}_2 &= [0.100000000]_2 = 1/2 \\ \tilde{\phi}_3 &= [0.110000000]_2 = 3/4\,. \end{aligned} \tag{229}$$

Again, these angles can be regarded as exact, and we can immediately extract the period r. We must, however, check every potential r to make sure that conditions (214)–(216) hold. We can skip the first peak at $\tilde{\phi}_0 = 0$, so let us now consider the third peak at $\tilde{\phi}_2 = 1/2$. The period $r = 2$ is even, but it does not satisfy requirement (216):

$$a^r \pmod N = 8^2 \pmod{15} = 4 \neq 1\,. \tag{230}$$

This illustrates that Shor's algorithm can fail for a given phase measurement $\tilde{\phi}$. However, the probability of success is quite high, and the algorithm usually requires at most a few tries before finding a factor. Let us move on to the second and fourth peaks, whose phases $\tilde{\phi}_1 = 1/4$ and $\tilde{\phi}_3 = 3/4$ give the period $r = 4$. Note that conditions (214)–(216) are indeed satisfied, since $r = 4$ is even, and

$$a^{r/2} \pmod N = 8^2 \pmod{15} = 4 \neq \pm 1 \pmod{15} \tag{231}$$

$$a^r \pmod N = 8^4 \pmod{15} = 1\,. \tag{232}$$

Thus, $r = 4$ is the exact period that we seek, which is confirmed by the upper-right panel of Fig. 11. The factors of $N = 15$ are therefore determined by $a^{r/2} = 8^2 = 64$, so that $\gcd(a^{r/2} - 1, N) = \gcd(63, 15) = 3$ and $\gcd(a^{r/2} + 1, N) = \gcd(65, 15) = 5$.

The results of this simulation are free from machine error, which would not be the case on a real quantum computer. One could build noise models for the various gates in Shor's algorithm, and then place acceptable error bounds on the circuit. This would require taking $n_\epsilon > 0$, which would increase the number of control qubits. In this document, we shall instead perform a simplified error analysis by just adding a 1 to the least significant bit of the phases $\tilde{\phi}$. That is to say, let us suppose the measurements are given by

$$\begin{aligned} \tilde{\ell}_0 &= [000000001]_2 = 1 \\ \tilde{\ell}_1 &= [010000001]_2 = 129 \\ \tilde{\ell}_2 &= [100000001]_2 = 257 \\ \tilde{\ell}_3 &= [110000001]_2 = 385\,, \end{aligned} \tag{233}$$

which produces the phases

$$\begin{aligned} \tilde{\phi}_0 &= [0.000000001]_2 = 1/2^9 = 0.001953125 = 1/512 \\ \tilde{\phi}_1 &= [0.010000001]_2 = 1/2^2 + 1/2^9 = 0.251953125 = 129/512 \\ \tilde{\phi}_2 &= [0.100000001]_2 = 1/2^1 + 1/2^9 = 0.501953125 = 257/512 \\ \tilde{\phi}_3 &= [0.110000001]_2 = 1/2^1 + 1/2^2 + 1/2^9 = 0.751953125 = 385/512\,. \end{aligned} \tag{234}$$

The method of continued fractions will now be required. We recommend a useful Python package called `contfrac`, which can be installed as follows [13]:

```
$ pip install contfrac
```

We can pick a phase $\tilde{\phi}$ at random, or we can examine every phase sequentially. Let us concentrate on $\tilde{\phi}_3 = 385/512$ as an example. With the above Python package, one can effortlessly find the continued fraction representation of the measured phase and its various convergents using the following code segment (with output included):

```
# import packages
import contfrac

#
phi = (385, 512) # phi3=[0.110000001]_2=0.751953125=385/512
coefficients = list(contfrac.continued_fraction(phi))
convergents = list(contfrac.convergents(phi))
#
print("cont frac of phi:",coefficients)
print("convergents of phi:", convergents)
```

output:

```
cont frac of phi: [0, 1, 3, 31, 1, 3]
convergents of phi: [(0,1),(1,1),(3,4),(94,125),(97,129),(385,512)]
```

Therefore, we can express the phase by the following continued fraction,

$$\tilde{\phi}_3 \quad = \quad [0.110000001]_2 = \frac{385}{512} = [0; 1, 3, 31, 1, 3] \, . \tag{235}$$

The convergents of $\tilde{\phi}_3$ have also been calculated:

$$\begin{aligned}
s_0/r_0 &= [0] = 0/1 \\
s_1/r_1 &= [0, 1] = 1/1 \\
s_2/r_2 &= [0, 1, 3] = 3/4 \quad \Leftarrow \text{ solution: } r = 4 \\
s_3/r_3 &= [0, 1, 3, 31] = 94/125 \\
s_4/r_4 &= [0, 1, 3, 31, 1] = 97/192 \\
s_5/r_5 &= [0, 1, 3, 31, 1, 3] = 385/512 \quad \Leftarrow \text{ trivial solution: } r = 512 = 4 \times 128 \, .
\end{aligned} \tag{236}$$

We must examine every convergent on the list, but fortunately there are only a handful of them. The convergents take the form s_ℓ/r_ℓ, where s_ℓ and r_ℓ are relatively prime. Since we are interested in the *smallest* value of r_ℓ such that (214)–(216) are satisfied, we must work our way *up* the list of convergents, from the *smallest* to the *largest* values of r_ℓ, testing every r_ℓ. This determines the *exact* period $r = r_\ell$ from the *approximately* measured phases. The first two convergents $s_0/r_0 = 0$ and $s_1/r_1 = 1$ are unacceptable, so we continue on to the convergent $s_2/r_2 = 3/4$, which gives the period $r_2 = 4$. This value indeed satisfies (214)–(216), giving the factors 3 and 5 as we have seen. We can stop here, but it is pedagogically useful to consider the other convergents. Note that $r_3 = 125$ and $r_4 = 192$ do not satisfy (216), and are therefore ruled out as possible periods. In contrast, note that $r_5 = 512$ is even and it does satisfies equation (216),

$$a^{r_5} \pmod N = 8^{512} \pmod{15} = 4096 \pmod{15} = 1 \, . \tag{237}$$

This is because $512 = 128 \times 4$ is a multiple of 4, and $r = 4$ satisfies (216). Note, however, that

$$a^{r_5/2} \pmod N = 8^{256} \pmod{15} = 1 \, , \tag{238}$$

and therefore $b = a^{r_5/2} = 8^{256}$ is a trivial root of unity, contrary to condition (215). This analysis can be automated using the following Python script. The first part of the script takes the binary input of the peak $\tilde{\ell}$, denoted by `l_phi`, and then converts it to a fraction $\tilde{\phi}_\ell = s_\ell/r_\ell$ with no common factors other than unity:

```
# import basics
import contfrac
from numpy import gcd

# construct decimal value of l_phi
n = 0
l_tilde = 0
for l in l_phi[::-1]:
    n += 1
    l_tilde = l_tilde + 2**(n-1) * int(l)
print("l_measured   :", l_phi, l_tilde)

# construct decimal value of phi
n = 0
phi_tilde = 0
for l in l_phi:
    n -= 1
    phi_tilde = phi_tilde + 2**n * int(l)
print("phi_phase_bin :", "0."+l_phi)
print("phi_phase_dec:", phi_tilde)

# express phi_tilde as a fraction
res = len(str(phi_tilde)) - 2 # subtract 2 for "0."
print("res:", res)
scale = 10**res # automated scale set by res
num = int(phi_tilde*scale)
den = int(scale)
# in lowest terms
c = gcd(num, den)
num = int(num / c)
den = int(den / c)
phi = (num, den)
print("phi:", phi)
```

We now pass the measured phase `phi` into the continued fraction package to find the convergents, and then we check each convergent to confirm that conditions (214)–(216) are satisfied. If they are not, we move on to the next peak in the histogram:

```
# construct convergents for phi
coefficients = list(contfrac.continued_fraction(phi))
convergents = list(contfrac.convergents(phi))
print("cont frac of phi:",coefficients)
print("convergents of phi:", convergents)

# check convergents for solution
for conv in convergents:
    r = conv[1]
    test1 = r % 2 # 0 if r is even
    test2 = (a**int(r/2)-1) % N # 0 if a^r/2 is a trivial root
    test3 = (a**int(r/2)+1) % N # 0 if a^r/2 is a trivial root
    test4 = a**r % N # 1 if r is a solution
    if (test1==0 and test2!=0 and test3!=0 and test4==1):
        print("conv:", conv, "r =", r, ": factors")
        print("factor1:", gcd(a**int(r/2)-1, N))
```

```
        print("factor2:", gcd(a**int(r/2)+1, N))
    else:
        print("conv:", conv, "r =", r, ": no factors found")
```

As an example of the script, we use the peak $\tilde{\ell}_3 = [110000001]_2$. As we see, this reproduces the previous analysis.

$\tilde{\ell}_3 = [110000001]_2 = 385$:

```
l_measured   : 110000001 385
phi_phase_bin: 0.110000001
phi_phase_dec: 0.751953125
res: 9
phi: (385, 512)
cont frac of phi  : [0, 1, 3, 31, 1, 3]
convergents of phi: [(0, 1), (1, 1), (3, 4), (94, 125), (97, 129), (385, 512)]
conv: (0, 1) r = 1 : no factors found
conv: (1, 1) r = 1 : no factors found
conv: (3, 4) r = 4 : factors
factor1: 3
factor2: 5
conv: (94, 125) r = 125 : no factors found
conv: (97, 129) r = 129 : no factors found
conv: (385, 512) r = 512 : no factors found
```

For completeness, we use the script to analyze the other three peaks of the histogram, starting with the 0-th peak. $\tilde{\ell}_0 = [000000001]_2 = 1$:

```
l_measured   : 000000001 1
phi_phase_bin: 0.000000001
phi_phase_dec: 0.001953125
res: 9
phi: (1, 512)
cont frac of phi  : [0, 512]
convergents of phi: [(0, 1), (1, 512)]
conv: (0, 1) r = 1 : no factors found
conv: (1, 512) r = 512 : no factors found
```

$\tilde{\ell}_2 = [010000001]_2 = 129$:

```
l_measured   : 010000001 129
phi_phase_bin: 0.010000001
phi_phase_dec: 0.251953125
res: 9
phi: (129, 512)
cont frac of phi  : [0, 3, 1, 31, 4]
convergents of phi: [(0, 1), (1, 3), (1, 4), (32, 127), (129, 512)]
conv: (0, 1) r = 1 : no factors found
conv: (1, 3) r = 3 : no factors found
conv: (1, 4) r = 4 : factors
factor1: 3
factor2: 5
conv: (32, 127) r = 127 : no factors found
conv: (129, 512) r = 512 : no factors found
```

$\tilde{\ell}_3 = [100000001]_2 = 257$:

```
l_measured   : 100000001 257
phi_phase_bin: 0.100000001
phi_phase_dec: 0.501953125
res: 9
phi: (257, 512)
cont frac of phi  : [0, 1, 1, 127, 2]
convergents of phi: [(0, 1), (1, 1), (1, 2), (128, 255), (257, 512)]
conv: (0, 1) r = 1 : no factors found
conv: (1, 1) r = 1 : no factors found
conv: (1, 2) r = 2 : no factors found
conv: (128, 255) r = 255 : no factors found
conv: (257, 512) r = 512 : no factors found
```

We see that $\tilde{\ell}_1$ and $\tilde{\ell}_3$ give the period $r = 4$, which results in the factors 3 and 5, while the other two peaks do not pass the requisite tests.

6.5. Analysis of the Phase Histogram

We close this section with a theoretical analysis of the phase histogram of the control register for a general number of qubits m. Our main focus will be calculating the locations of the peaks $\tilde{\ell}_n$. We have already examined the situation for $m = 9$, in which Fig. 19 illustrates the Qiskit output histograms for $N = 15$ with the bases $a = 4$ and $a = 8$. In this section we will primarily concentrate on the $a = 8$ histogram with four peaks. Before looking at a general value of m, however, let us first examine the simpler case of $m = 5$ qubits (with $a = 8$), where the Qiskit phase histogram is illustrated in Fig. 20.

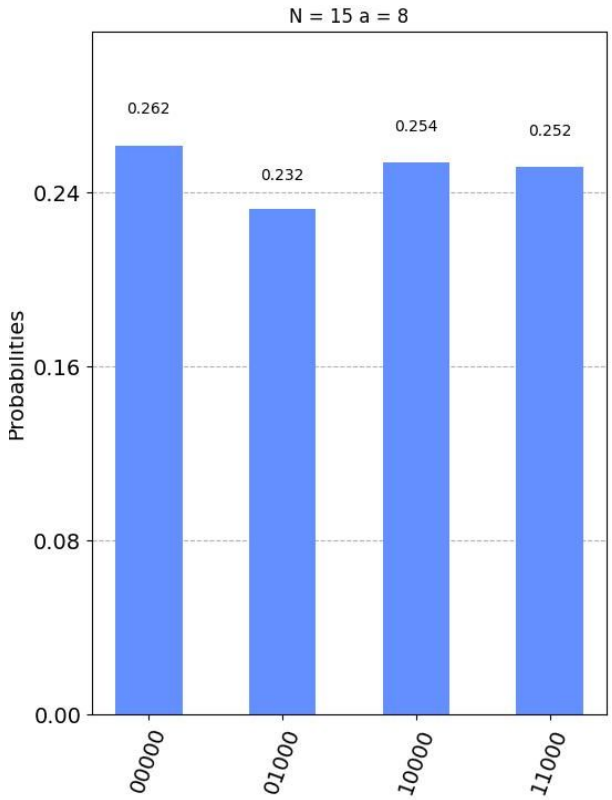


Figure 20: *The phase histogram from Qiskit for $N = 15$ and $a = 8$, as in the right panel of Fig. 19, except that the control register has $m = 5$ qubits. There are a total of $M = 2^5 = 32$ states, and the peaks occur at $\tilde{\ell}_0 = 0$, $\tilde{\ell}_1 = 8$, $\tilde{\ell}_2 = 16$, and $\tilde{\ell}_3 = 24$. The corresponding phases are $\tilde{\phi}_0 = 0$, $\tilde{\phi}_1 = 1/4$, $\tilde{\phi}_2 = 1/2$, and $\tilde{\phi}_3 = 3/4$. From these phases, we can infer that $r = 4$. Note that Fig. 19 uses $m = 9$ qubits, and the peaks therefore occur at different values of $\tilde{\ell}_n$, although the phases turn out to be the same.*

We do this because the $m = 5$ case can be calculated quite easily using a minimum of algebra. The histogram peaks now lie at different values of $\tilde{\ell}_n$ from those of the $m = 9$ simulation since the value of m differs, but the phases $\tilde{\phi}_n = \tilde{\ell}_n/2^m$ are identical. For $m = 5$ there are $M = 2^5 = 32$ quantum states, and the corresponding output phases are given by

$$\begin{aligned} \tilde{\ell}_0 &= [00000]_2 = 0 & \tilde{\phi}_0 &= [0.00000]_2 = 0 \\ \tilde{\ell}_1 &= [01000]_2 = 8 & \tilde{\phi}_1 &= [0.01000]_2 = 1/4 \\ \tilde{\ell}_2 &= [10000]_2 = 16 & \tilde{\phi}_2 &= [0.10000]_2 = 1/2 \\ \tilde{\ell}_3 &= [11000]_2 = 24 & \tilde{\phi}_3 &= [0.11000]_2 = 3/4\,. \end{aligned} \tag{239}$$

Just as in Fig. 15, we measured the $m = 5$ control registers after the inverse Fourier transform $QFT^\dagger$ was applied (at position 4 in the Figure). Note that the control and work registers are entangled at position 2 (just before the $QFT^\dagger$ operator) because of the action of the ME operators CU^p. The measurement of the control register at position 4 therefore collapses the quantum state of the work register. While we have not yet talked about work register measurements, there is no reason why we should not be able to simultaneously measure the work register and the control register, as illustrated in position 4 in the top panel of Fig. 21. The work register measurement is indicated by the short red bar across the register at position 4, right after the $QFT^\dagger$ operation acts on the control register. In fact, we could measure the work register *before* the $QFT^\dagger$ operation, at position 3 in the bottom panel of Fig. 21. When the work register is measured in this way, the state of the control register collapses at position 2, but we must obtain the same result as in previous measurement at position 4.

Let us examine this situation in more detail. We have in fact essentially performed this calculation near the end of Section 3.4, but the following method brings out additional physics. Note that the wavefunction of the control- and work-register system at position 2 takes the form (before the measurement),

$$|\psi_2\rangle = \frac{1}{\sqrt{32}} \sum_{k=0}^{31} |k\rangle \otimes \left|8^k \pmod{15}\right\rangle\,. \tag{240}$$

As we have seen, the modular exponential function $f(x) = 8^x \pmod{15}$ has a period $r = 4$, and it takes the values $f(0) = 1$, $f(1) = 8$, $f(2) = 4$, and $f(3) = 2$. We can therefore express the state $|\psi_2\rangle$ in (240) by the following:

$$|\psi_2\rangle = \frac{1}{\sqrt{32}} \sum_{k=0}^{31} |k\rangle \otimes \underbrace{|8^k \pmod{15}\rangle}_{1,8,4,2} \tag{241}$$

$$\begin{aligned} = \frac{1}{\sqrt{32}}\Big[\, & |0\rangle \otimes |1\rangle + |1\rangle \otimes |8\rangle + |2\rangle \otimes |4\rangle + |3\rangle \otimes |2\rangle + \\ & |4\rangle \otimes |1\rangle + |5\rangle \otimes |8\rangle + |6\rangle \otimes |4\rangle + |7\rangle \otimes |2\rangle + \\ & |8\rangle \otimes |1\rangle + |9\rangle \otimes |8\rangle + |10\rangle \otimes |4\rangle + |11\rangle \otimes |2\rangle + \\ & |12\rangle \otimes |1\rangle + |13\rangle \otimes |8\rangle + |14\rangle \otimes |4\rangle + |15\rangle \otimes |2\rangle + \\ & |16\rangle \otimes |1\rangle + |17\rangle \otimes |8\rangle + |18\rangle \otimes |4\rangle + |19\rangle \otimes |2\rangle + \\ & |20\rangle \otimes |1\rangle + |21\rangle \otimes |8\rangle + |22\rangle \otimes |4\rangle + |23\rangle \otimes |2\rangle + \\ & |24\rangle \otimes |1\rangle + |25\rangle \otimes |8\rangle + |26\rangle \otimes |4\rangle + |27\rangle \otimes |2\rangle + \\ & |28\rangle \otimes |1\rangle + |29\rangle \otimes |8\rangle + |30\rangle \otimes |4\rangle + |31\rangle \otimes |2\rangle \,\Big]\,. \end{aligned} \tag{242}$$

Since $M = 2^5 = 32$ is so small, we have been able to write down every term in the wavefunction. Upon collecting like states in the work register, we find

$$\begin{aligned} |\psi_2\rangle = \frac{1}{\sqrt{32}}\Big[& \Big(|0\rangle + |4\rangle + |8\rangle + |12\rangle + |16\rangle + |20\rangle + |24\rangle + |28\rangle\Big) \otimes |1\rangle + \\ & \Big(|1\rangle + |5\rangle + |9\rangle + |13\rangle + |17\rangle + |21\rangle + |25\rangle + |29\rangle\Big) \otimes |8\rangle + \end{aligned} \tag{243}$$

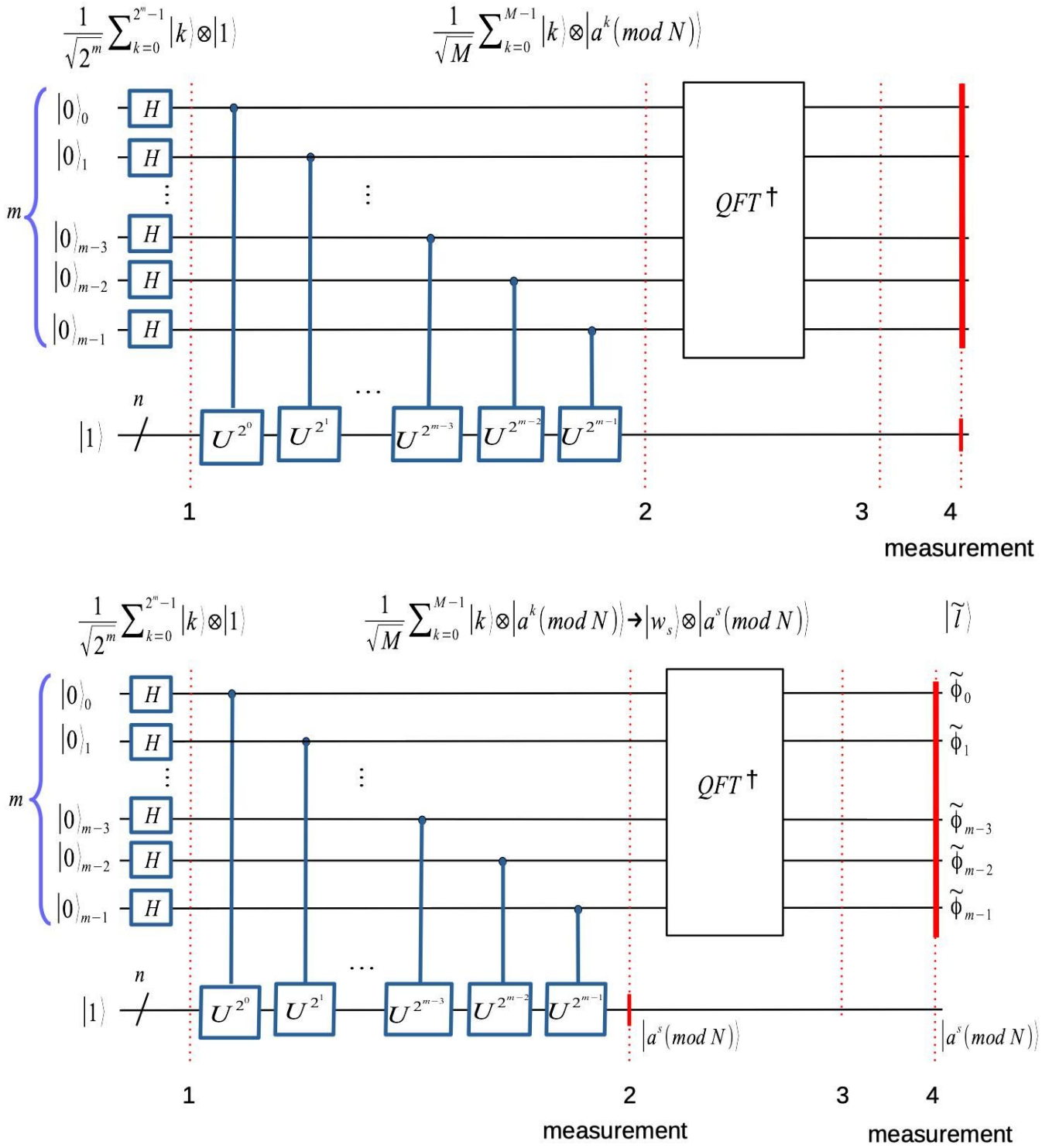


Figure 21: *Changing the order of measurements.*

$$\Big(|2\rangle+|6\rangle+|10\rangle+|14\rangle+|18\rangle+|22\rangle+|26\rangle+|30\rangle\Big)\otimes|4\rangle+$$
$$\Big(|3\rangle+|7\rangle+|11\rangle+|15\rangle+|19\rangle+|23\rangle+|27\rangle+|31\rangle\Big)\otimes|2\rangle\ \Big]$$
$$= \frac{1}{\sqrt{4}}\Big[|w_0\rangle\otimes|f(0)\rangle+|w_1\rangle\otimes|f(1)\rangle+|w_2\rangle\otimes|f(2)\rangle+|w_3\rangle\otimes|f(3)\rangle\Big], \tag{244}$$

where the four control register states are defined by

$$|w_0\rangle = \sqrt{\frac{4}{32}}\Big(|0\rangle+|4\rangle+|8\rangle+|12\rangle+|16\rangle+|20\rangle+|24\rangle+|28\rangle\Big) \tag{245}$$

$$|w_1\rangle = \sqrt{\frac{4}{32}}\Big(|1\rangle + |5\rangle + |9\rangle + |13\rangle + |17\rangle + |21\rangle + |25\rangle + |29\rangle\Big) \tag{246}$$

$$|w_2\rangle = \sqrt{\frac{4}{32}}\Big(|2\rangle + |6\rangle + |10\rangle + |14\rangle + |18\rangle + |22\rangle + |26\rangle + |30\rangle\Big) \tag{247}$$

$$|w_3\rangle = \sqrt{\frac{4}{32}}\Big(|3\rangle + |7\rangle + |11\rangle + |15\rangle + |19\rangle + |23\rangle + |27\rangle + |31\rangle\Big)\,. \tag{248}$$

We can re-express (244) in the more general form

$$|\psi_2\rangle = \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|w_s\rangle \otimes |f(s)\rangle\,, \tag{249}$$

which also holds for period $r = 2$, although the states $|w_s\rangle$ and $|f(s)\rangle$ will be different. Returning to $r = 4$, we can now generalize (245)–(248) to arbitrary m, where $M = 2^m$:

$$|w_s\rangle = \sqrt{\frac{4}{M}}\sum_{k=0}^{M/4-1}|s+4k\rangle \quad \text{for} \;\; s \in \{0,1,2,3\}\,. \tag{250}$$

This form relies on a special feature of $N = 15$, namely that $M/r = 2^m/4 = 2^{m-2}$ is an integer for $r = 4$. Also note that $M/r = M/2 = 2^{m-1}$ is also an integer for $r = 2$. This leads to the further generalization

$$|w_s\rangle = \sqrt{\frac{r}{M}}\sum_{k=0}^{M/r-1}|s+rk\rangle \quad \text{for} \;\; s \in \{0,\cdots,r-1\}\,. \tag{251}$$

When $r = 2$ we have $s \in \{0, 1\}$, and when $r = 4$ we have $s \in \{0, 1, 2, 3\}$.

Returning again to $r = 4$, let us now measure the state $|\psi_2\rangle$ at position 2, as illustrated by the lower panel of Fig. 21. This leads to wavefunction collapse, so that

$$|\psi_2\rangle = \frac{1}{\sqrt{4}}\sum_{s=0}^{3}|w_s\rangle \otimes |f(s)\rangle \to |w_s\rangle \otimes |f(s)\rangle\,, \tag{252}$$

where $s \in \{0, 1, 2, 3\}$ is randomly selected with a uniform probability of $1/4$. We must now apply the inverse Fourier transform to the control register, thereby giving the state

$$QFT^\dagger\,|w_s\rangle \equiv \sum_{\ell=0}^{M-1}A_{\ell,s}\,|\ell\rangle\,. \tag{253}$$

The next step is to find the amplitudes $A_{\ell,s}$ by performing the inverse Fourier transform on the state $|w_s\rangle$, which can be calculated exactly:

$$QFT^\dagger\,|w_s\rangle = \sqrt{\frac{4}{M}}\sum_{k=0}^{M/4-1}QFT^\dagger\,|s+4k\rangle = \sqrt{\frac{4}{M}}\sum_{k=0}^{M/4-1}\frac{1}{\sqrt{M}}\sum_{\ell=0}^{M-1}e^{-2\pi i\,\ell(s+4k)/M}\,|\ell\rangle \tag{254}$$

$$= \frac{2}{M}\sum_{\ell=0}^{M-1}e^{-2\pi i\,\ell s/M}\sum_{k=0}^{M/4-1}e^{-8\pi i\,k\ell/M}\,|\ell\rangle \tag{255}$$

$$= \frac{2}{M}\sum_{\ell=0}^{M-1}e^{-2\pi i\,\ell s/M}\,\frac{1-\left[e^{-8\pi i\,\ell/M}\right]^{M/4}}{1-e^{-8\pi i\,\ell/M}}\,|\ell\rangle \tag{256}$$

$$= \frac{2}{M}\sum_{\ell=0}^{M-1}e^{-2\pi i\,\ell s/M}\,\frac{1-e^{-2\pi i\,\ell}}{1-e^{-8\pi i\,\ell/M}}\,|\ell\rangle\,. \tag{257}$$

We therefore find

$$A_{\ell,s} = e^{-2\pi i\,\ell s/M}\,\frac{2}{M}\,\frac{1-e^{-2\pi i\,\ell}}{1-e^{-8\pi i\,\ell/M}} \quad \text{for } \ell \in \{0,1,\cdots,M-1\}\,. \tag{258}$$

Note that the s-dependence lies only in the complex phases, and therefore the probabilities $P_\ell = |A_{\ell,s}|^2$ can be expressed by

$$P_\ell = \frac{4}{M^2}\left|\frac{1-e^{-2\pi i\,\ell}}{1-e^{-8\pi i\,\ell/M}}\right|^2 = \frac{4}{M^2}\,\frac{1-\cos(2\pi\ell)}{1-\cos(8\pi\ell/M)} \tag{259}$$

$$= \frac{4}{M^2}\,\frac{\sin^2(\pi\ell)}{\sin^2(4\pi\ell/M)} \quad \text{for } \ell \in \{0,1,\cdots,M-1\}\,. \tag{260}$$

Also note that the numerator vanishes for every integer ℓ, so the only way we can obtain a non-zero probability is when the denominator also vanishes (so that we have the indeterminate form $0/0$). However, the denominator vanishes only for ℓ such that

$$\frac{4\pi\ell}{M} = n\pi \quad \text{for } n \in \mathbb{Z}\,, \tag{261}$$

or equivalently for

$$\ell_n = \frac{nM}{4} \quad \text{for } n \in \{0,1,2,3\}\,. \tag{262}$$

We have dropped the tilde over ℓ_n since this is a theoretical prediction and not a measurement. The value of n is restricted to $\{0,1,2,3\}$ because $\ell = \ell_n$ must be a non-negative integer that cannot exceed $M-1$. Note that expression (261) means that

$$\sin(4\pi\ell_n/M) = 0 \tag{263}$$

$$\cos(4\pi\ell_n/M) = \pm 1\,, \tag{264}$$

relations that we shall use momentarily. From the probability (260), we see that P_ℓ vanishes for all values of ℓ except for $\ell = \ell_n$. Thus, P_ℓ vanishes at all but four of its 2^m possibilities! We must next calculate the corresponding probabilities at the four poles $\ell = \ell_n$, and it should come as no surprise that they are all equally likely with probability $P = 1/4$.

Let us now turn to calculating these probabilities. In the language of quantum field theory, we must perform a *regularization* procedure on the function (260), thereby eliminating the poles at $\ell_n = nM/4$. To do this, we shall (i) displace each pole ℓ_n by a small distance ε, (ii) evaluate this *regularized* probability exactly for a non-zero ε, and (iii) only afterward take the limit of zero displacement $\varepsilon \to 0$. Therefore, let us make the shift

$$\ell_n \to \ell_n + \varepsilon \tag{265}$$

in expression (260). Upon using the more suggestive notation $P(\ell) \equiv P_\ell$, we thereby *define* the regularized probabilities by

$$P_n \equiv \lim_{\varepsilon\to 0} P(\ell_n + \varepsilon)\,. \tag{266}$$

We can now calculate the probabilities at the poles $\ell = \ell_n$:

$$P_n = \frac{4}{M^2}\lim_{\varepsilon\to 0}\left[\frac{\sin(\pi\ell_n + \pi\varepsilon)}{\sin(4\pi\ell_n/M + 4\pi\varepsilon/M)}\right]^2 \tag{267}$$

$$= \frac{4}{M^2}\lim_{\varepsilon\to 0}\left[\frac{\sin(\pi\ell_n)\cos(\pi\varepsilon) + \cos(\pi\ell_n)\sin(\pi\varepsilon)}{\sin(4\pi\ell_n/M)\cos(4\pi\varepsilon/M) + \cos(4\pi\ell_m/M)\sin(4\pi\varepsilon/M)}\right]^2 \tag{268}$$

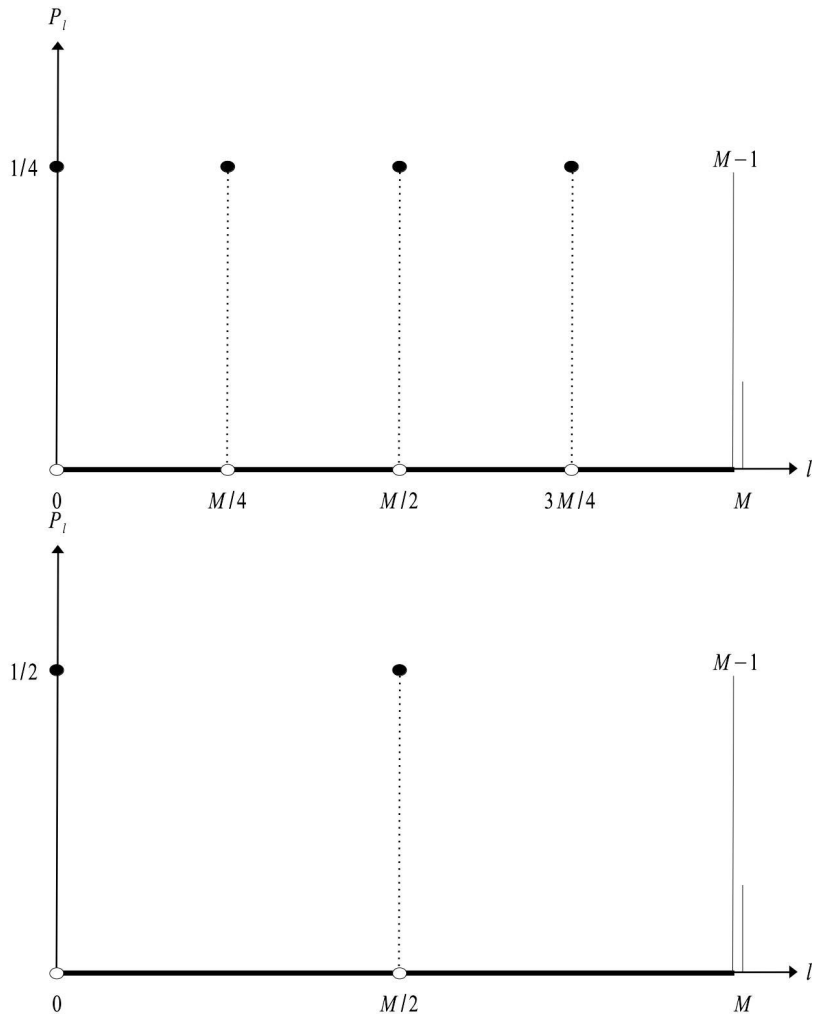


Figure 22: *Probabilities P_ℓ for $\ell \in \{0, 1, \cdots, M-1\}$. Top panel is for $a = 8$ (period $r = 4$) , and the bottom panel is for $a = 4$ (period $r = 2$).*

$$= \frac{4}{M^2} \lim_{\varepsilon \to 0} \left[\frac{\sin(\pi\varepsilon)}{\sin(4\pi\varepsilon/M)} \right]^2 = \frac{4}{M^2} \lim_{\varepsilon \to 0} \left[\frac{\pi\varepsilon}{4\pi\varepsilon/M} \right]^2 = \frac{1}{4} , \tag{269}$$

where we have used relations (263) and (264). These are the only non-zero values of P_ℓ, and the graph of the probabilities for $a = 8$ is shown in the top panel of Fig. 22. For the period $r = 2$, we would find

$$\ell_n = \frac{nM}{2} \quad \text{for} \quad n \in \{0, 1\} , \tag{270}$$

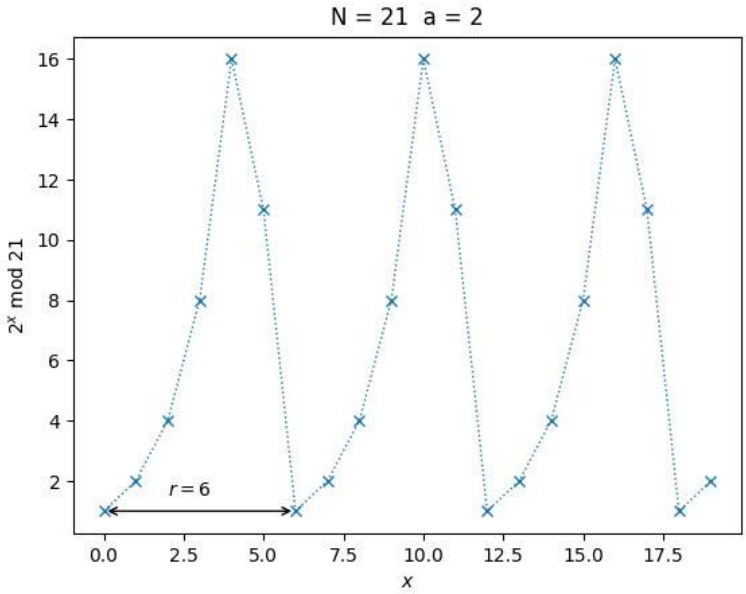
with $P_n = 1/2$, which is illustrated in the bottom panel of Fig. 22.

Let us now check this calculation against the previous Qiskit output for $m = 5$ and $m = 9$.

(i) $a = 8$ or $r = 4$: For $M = 2^5 = 32$, we find $\ell_n = 8n$, or $\ell_0 = 0, \ell_1 = 8, \ell_2 = 16, \ell_3 = 24$, in agreement with (239). For $M = 2^9$, we have $\ell_n = 128n$, or $\ell_0 = 0, \ell_1 = 128, \ell_2 = 256, \ell_3 = 384$, in agreement with (228).

(ii) $a = 4$ or $r = 2$: For $m = 9$ we have $M = 2^9$, and therefore $\ell_n = 256n$. Thus, $\ell_0 = 0$ and $\ell_1 = 256$, in agreement with (224).

This analysis is actually a special case of a result that we have already derived. In Section 3.4 we showed that when the phase angle θ is such that $\ell_\theta \equiv 2^m\theta$ is an integer for m control qubits, then the amplitude simplifies to $A_\ell(\theta) = \delta_{\ell,\ell_\theta}$. This is exactly what we have shown here for $N = 15$ and $a = 4, 8$, since the phases are $\phi_s = s/r$ with the periods $r = 2, 4$, which are just powers of 2.



$U\|w\rangle =$	$\|2 \cdot w \pmod{21}\rangle$
$U\|1\rangle = \|2\rangle$	$U\|00001\rangle = \|00010\rangle$
$U\|2\rangle = \|4\rangle$	$U\|00010\rangle = \|00100\rangle$
$U\|4\rangle = \|8\rangle$	$U\|00100\rangle = \|01000\rangle$
$U\|8\rangle = \|16\rangle$	$U\|01000\rangle = \|10000\rangle$
$U\|16\rangle = \|11\rangle$	$U\|10000\rangle = \|01011\rangle$
$U\|11\rangle = \|1\rangle$	$U\|01011\rangle = \|00001\rangle$

Figure 23: *$N = 21$, $a = 2$, $r = 6$: The left panel illustrates the modular exponential function $f_{2,21}(x) = 2^x \pmod{21}$, while the right panel shows the action of the ME operator $U_{2,21}$ on the closed sequence $[1, 2, 4, 8, 16, 11, 1]$.*

7. Further Examples

7.1. Factoring Larger Numbers: N = 21 = 3 × 7, a = 2, r = 6

We now turn to factoring numbers larger than $N = 15$, where we will need to construct more complex *modular exponentiation* (ME) operators $U_{a\,N}$ for an appropriate base a. While the $N = 15$ operators were rather easy to construct, this is not the case for larger values of N. The ME operators $U_{a,15}$ were completely general, valid for any permissible base a, and acting on any computational basis element in the work-state Hilbert space. In contrast, creating such general operators for larger values of N appears to be extremely difficult. However, we do not require the general structure of the ME operators. This is because the first operation of $U_{a\,N}$ acts on the work-state $|1\rangle = |0 \cdots 01\rangle$, and the next operation acts on the output of the first, and so on. Since

$$U_{a\,N}^{x}|1\rangle = \big|f_{a\,N}(x)\big\rangle \;\text{ for any } x \in \{0, 1, 2, \cdots\}\,, \tag{271}$$

we therefore only need to find the operation of $U_{a\,N}$ on the states $|f_{a\,N}(x)\rangle$ for $x = 0, 1, \cdots, r-1$, where r is the period of $f_{a\,N}(x)$. Let us return momentarily to a general number of work qubits n. We see that the work space, which we shall denote by $\mathcal{W}_n$, has dimension 2^n, and a general ME operator U can act on this entire Hilbert space. Consider now the r-dimensional subspace defined by

$$\mathcal{U}_r \equiv \mathrm{S}pan\Big\{\,\big|f(x)\big\rangle\;\Big|\;x \in \{0, 1, \cdots, r-1\}\Big\} \subseteq \mathcal{W}_n\,. \tag{272}$$

As discussed above, the U operator transforms one basis element of $\mathcal{U}_r$ into another basis element, that is to say, the ME operator U leaves the r-dimensional space $\mathcal{U}_r$ invariant, so that $U[\mathcal{U}_r] = \mathcal{U}_r$. Thus, as the U operator acts successively, the states in the work register only vary over the r-dimensional subspace $\mathcal{U}_r$. The operators CU^p therefore entangle the control register and the subspace $\mathcal{U}_r$ (and not the entire work space), and this plays a crucial role in the exponential speedup of Shor's algorithm. We have reduced the problem to the action of ME operator U on the lower dimensional subspace $\mathcal{U}_r$ of the exponentially large work space $\mathcal{W}_n$, and we can henceforth restrict our attention to this subspace $\mathcal{U}_r$. This is quite similar to Grover's search algorithm that reduces to movement within a 2-dimensional subspace.

In the case of $N = 21$ with base $a = 2$, the left panel of Fig. 23 illustrates the modular exponential function $f_{2,21}(x)$. The period is observed to be $r = 6$, with the closed cycle $[1, 2, 4, 8, 16, 11, 1]$, as illustrated in the right panel of the Figure. Thus, we only need to consider $U_{2,21}$ on the states $|1\rangle$, $|2\rangle$, $|4\rangle$, $|8\rangle$, $|16\rangle$, and $|11\rangle$. We must employ $n = \lceil \log_2 21 \rceil = 5$ qubits to enumerate these states in the work register.

The work states can therefore be indexed by a binary string of the form $w_4w_3w_2w_1w_0$ with $w_k \in \{0, 1\}$, and we can use a collection of multi-control C^nX and X gates to transform this string into the next string in the sequence. Furthermore, we must measure the phase in the m-bit control register with sufficient accuracy to extract the correct period. For $r = 6$, the permissible Eigen-phases of the ME operator are $\phi_s = s/6$ for $s \in \{0, 1, \cdots, 5\}$, and we must therefor be able to resolve a phase difference of $\Delta\phi = 1/6 \approx 0.16666$. We showed earlier in the text that the continued fractions method requires $m = 2n + 1 = 11$ control qubits (for $n = 5$); however, since $r = 6$ is so small, it turns out that we can get by with only $m = 5$. Therefore, the Shor circuit for $N = 21$ and $a = 2$ will have the same structure as the one illustrated in Fig. 16, except that the number of control qubits will be reduced to $m = 5$.

We will have to construct the appropriate ME operators $U^p_{2,21}$ for $p \in \{2^0, 2^1, 2^2, 2^3, 2^4\}$, namely U, U^2, U^4, U^8, U^{16}, since there are five control qubits. Later in this section we shall consider $m = 6$ control qubits, in which case we will also require U^{32}. For the time being, we will only construct the U operator, and then concatenate this operator to form the composite operators U^p for $p > 1$. We will refer to this procedure by version number `u_ver` = 0. It is clear that this method will not work for general N, as it requires an exponentially large number of concatenations. We will shortly illustrate how to directly construct the set of composite operators U^p for $p > 1$, but for the time being we shall continue with our current line of development using simple concatenation.

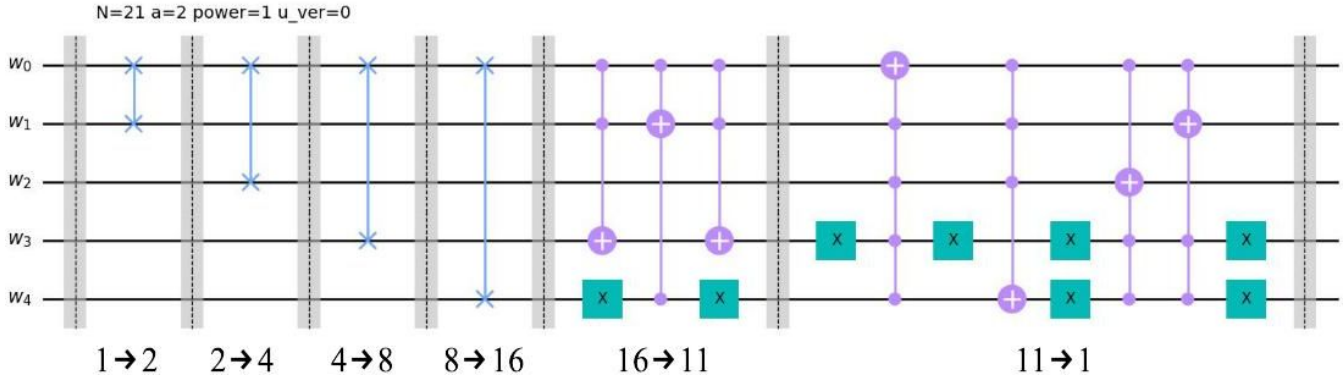


Figure 24: *$N = 21$, $a = 2$, $r = 6$: The quantum circuit for the modular exponentiation (ME) operator $U_{2,21}$. The quantum gates between the barriers transform the state from one value of $f_{2,21}(x)$ to the next in the closed sequence* $[1, 2, 4, 8, 16, 11, 1]$. *We will call this version of the ME operator by* `u_ver` = 0.

Figure 24 illustrates the ME operator $U = U_{2,21}$. This operator is constructed only from multi-control-NOT gates C^nX and single-qubit NOT gates X. The operator $U_{2,21}$ is partitioned into six sections, where each section is indexed by an integer $x = 0, 1, \cdots, 5$, and the gates in section x transform the work-state $|f(x)\rangle \equiv |w_4 \cdots w_0\rangle$ into $|f(x + 1)\rangle \equiv |w'_4 \cdots w'_0\rangle$. For example, since $U|1\rangle = |2\rangle$, the SWAP gate between the first barrier changes the initial state $|1\rangle = |00001\rangle$ into the next state $|2\rangle = |00010\rangle$. The second SWAP operation transforms $|2\rangle = |00010\rangle$ into $|4\rangle = |00100\rangle$, and so on. This is illustrated in the right panel of Fig. 23, and by the annotations under the barriers in Fig. 24. We shall use this restricted version of the ME operator $U_{2,21}$ in Shor's algorithm. For $m = 5$ control qubits and $n = 5$ work qubits, Fig. 25 illustrates the output phase histogram from a Qiskit simulation using 4096 shots. The abscissa indexes the possible phases, and the ordinate provides their corresponding probabilities. In agreement with the phase histogram of Fig. 25, Shor's algorithm is designed so that the most dominant phases correspond to the Eigen-phases of $U_{2,21}$, which take the simple form $\phi_s = s/6$ for$s \in \{0, 1, \cdots, 5\}$. The phases for which $\gcd(s, 6) = 1$ (namely $s = 1$ and $s = 5$) lead to the factors of $N = 21$, and these peaks are plotted in red. The code output for these phases is detailed in Table 4. Note that the phase histogram for $N = 21$ is considerably more complex than the $N = 15$ histogram in Fig. 20.

Let us examine the phase histogram in Fig. 25 in a little more detail. As noted above, the abscissa gives the phases of the ME operator $U_{2,21}$. They are represented by 5-bit integers $\tilde{\ell} = \tilde{\phi}_4 \cdots \tilde{\phi}_0$, and correspond to the binary phases $\tilde{\phi} = \tilde{\ell}/2^5 = 0.\tilde{\phi}_4 \cdots \tilde{\phi}_0$, where $\tilde{\phi}_k \in \{0, 1\}$ is the measured value of qubit k in the control register. The ordinate gives the probability that the given phase will be observed during a measurement of the control register. As noted in the previous paragraph, we expect the phase histogram to have large peaks close to the six phases of the ME operator,

$$\begin{aligned} \phi_0 &= 0 = 0.00000\cdots \\ \phi_1 &= 1/6 = 0.16666\cdots \\ \phi_2 &= 2/6 = 0.33333\cdots \end{aligned} \tag{273}$$

$$
\begin{aligned}
\phi_3 &= 3/6 = 0.50000\cdots \\
\phi_4 &= 4/6 = 0.66666\cdots \\
\phi_5 &= 5/6 = 0.83333\cdots \ .
\end{aligned}
$$

And indeed it does, as the six dominant peaks in Fig. 25 occur at

$$
\begin{array}{lll}
\tilde{\ell}_0 = [00000]_2 = 0 & \tilde{\phi}_0 = [0.00000]_2 = 0.00000 = \phi_0 & \\
\tilde{\ell}_1 = [00101]_2 = 5 & \tilde{\phi}_1 = [0.00101]_2 = 0.15625 \approx \phi_1 & \Leftarrow \text{ factors : } 3, 7 \\
\tilde{\ell}_2 = [01011]_2 = 11 & \tilde{\phi}_2 = [0.01011]_2 = 0.34375 \approx \phi_2 & \\
\tilde{\ell}_3 = [10000]_2 = 16 & \tilde{\phi}_3 = [0.10000]_2 = 0.50000 = \phi_3 & \\
\tilde{\ell}_4 = [10101]_2 = 21 & \tilde{\phi}_4 = [0.10101]_2 = 0.65625 \approx \phi_4 & \\
\tilde{\ell}_5 = [11011]_2 = 27 & \tilde{\phi}_5 = [0.11011]_2 = 0.84375 \approx \phi_5 & \Leftarrow \text{ factors : } 3, 7 \ .
\end{array}
\tag{274}
$$

A phase measurement $\tilde{\phi}_s$ will typically not exactly equal the associated Eigen-phase ϕ_s because of the finite resolution of the control register. However, each peak $\tilde{\phi}_s$ lies very close to an actual phase $\phi_s = s/6$, although only the phases $s = 1$ and $s = 5$ produce the factors of 3 and 7. The probability of finding a factor during each shot is about $2/6 \approx 30\%$, and we are therefore almost guaranteed to find a factor after only a few iterations.

Let us next examine the first entry in Table 4 in some detail. This entry corresponds to the first red peak in the phase histogram, $\tilde{\ell}_1 = [00101]_2 = 5$, which gives a measured phase of $\tilde{\phi}_1 = [0.00101]_2 = 0.15625$. This lies very close to $\phi_1 = 1/6 = 0.16666\cdots$. In general, the difference between each measured phase and the exact ME phase is of order 0.01, which is less than the resolution $2^{-5} = 0.03125$. We also list the frequency of occurrence out of the total number of shots of 4096. The Python continued fraction module `contfrac` prefers fractional inputs, so we convert the decimal value of the phase to a fraction, $\tilde{\phi}_1 = 5/32$, and the corresponding continued fraction is found to be

$$
\tilde{\phi}_1 = 0.15625 = 5/32 = [0; 0, 6, 2, 2] \ . \tag{275}
$$

The module `contfrac` then calculates all possible convergents of the rational number $\tilde{\phi}_1 = 5/32$, thereby giving $c_0 = 0/1$, $c_1 = 1/6$, $c_2 = 2/13$, $c_3 = 5/32$. Note that these fractions are represented by the ordered pairs $(0, 1)$,

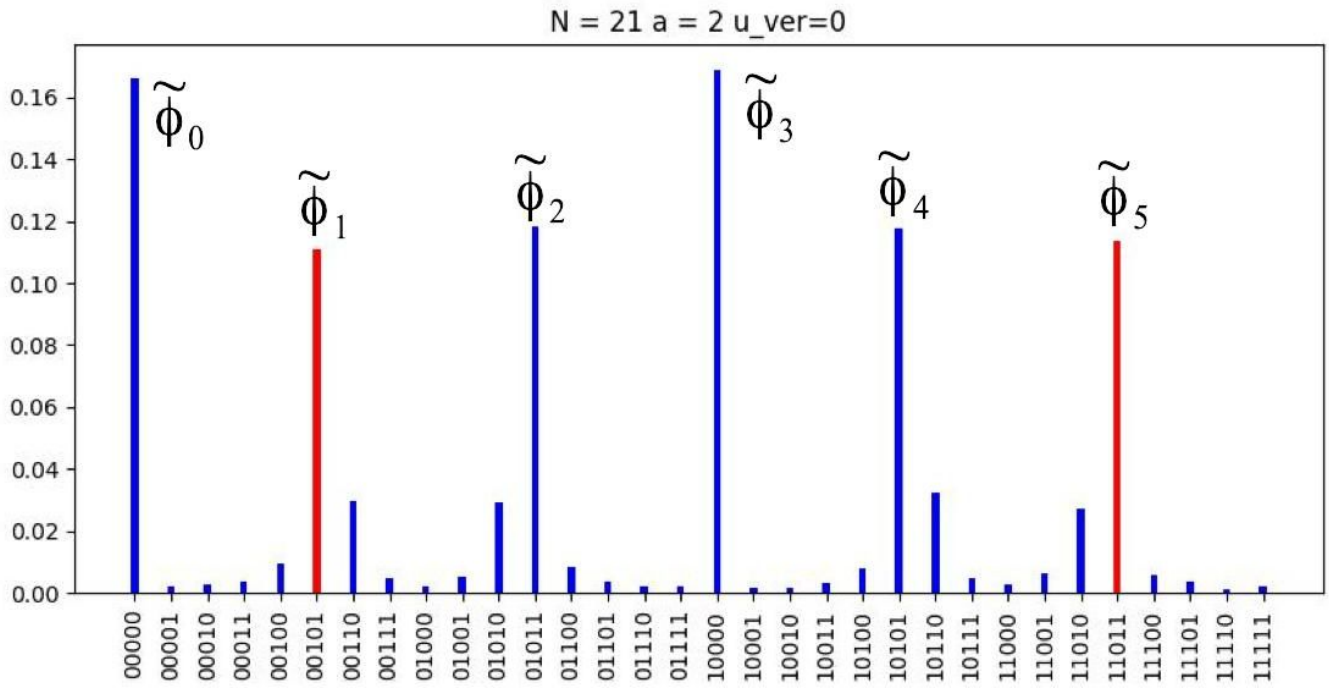


Figure 25: *The phase histogram for $N = 21$, $a = 2$ and $m = 5$ for* `u_ver` $= 0$ *from a Qiskit simulation with 4096 runs. The six dominant peaks of the histogram occur very close to the six Eigen-phases $\phi_s = s/6$ of the ME operator $U_{2,21}$, where $s \in \{0, 1, \cdots, 5\}$. The phases that produce factors are shown in red, occurring at the (binary) values $\tilde{\phi}_1 = [0.00101]_2 \approx \phi_1 = 1/6$ and $\tilde{\phi}_5 = [0.11011]_2 \approx \phi_5 = 5/6$. Note that these phase peaks lie well above the noise.*

Table 4: *The output of Shor's algorithm for $N = 21$, $a = 2$ and $m = 5$ for version* `u_ver` $= 0$*. Only the two phase values that produced factors are listed. The variable* `l_measured` *corresponds to the control register state indexed by the integer $\tilde{\ell} = \tilde{\phi}_4 \cdots \tilde{\phi}_0$, while* `phi_phase_bin` *corresponds to the 5-bit binary (measured) phase $\tilde{\phi} = \tilde{\ell}/2^5 = 0.\tilde{\phi}_4 \cdots \tilde{\phi}_0$. The decimal representation of the phase is also provided for convenience. The continued fraction representation, and the associated convergents (from the Python package* `contfrac`*) are also given, where each convergent $c = s/r$ is represented by an ordered pair (s, r). The code checks to see if the denominator r is a solution to (214)–(216). If r is a solution, then the factors are given by $gcd(a^{r/2} \pm 1, N)$.*

```
l_measured   : 00101 5 frequency: 466
phi_phase_bin: 0.00101
phi_phase_dec: 0.15625
phi: (5, 32)
cont frac of phi  : [0, 6, 2, 2]
convergents of phi: [(0, 1), (1, 6), (2, 13), (5, 32)]
conv: (0, 1) r = 1 : no factors found
conv: (1, 6) r = 6 : factors
factor1: 7
factor2: 3
conv: (2, 13) r = 13 : no factors found
conv: (5, 32) r = 32 : no factors found

l_measured   : 11011 27 frequency: 458
phi_phase_bin: 0.11011
phi_phase_dec: 0.84375
phi: (27, 32)
cont frac of phi  : [0, 1, 5, 2, 2]
convergents of phi: [(0, 1), (1, 1), (5, 6), (11, 13), (27, 32)]
conv: (0, 1) r = 1 : no factors found
conv: (1, 1) r = 1 : no factors found
conv: (5, 6) r = 6 : factors
factor1: 7
factor2: 3
conv: (11, 13) r = 13 : no factors found
conv: (27, 32) r = 32 : no factors found
```

$(1, 6)$, $(2, 13)$ and $(5, 32)$ in the continued fraction package. We then sequence through this small list of convergents $c_\ell = s_\ell/r_\ell$, testing each value of $r = r_\ell$ for a solution to (214)–(216). As we expect, only $r_1 = 6$ provides such a solution. As a matter of completeness, we provide below the code output for phase $\tilde{\phi}_2$, which does not produce factors.

Code output for $N = 21$, $a = 2$, $r = 6$, $m = 5$, peak $\tilde{\phi}_2 = [0.01011]_2 = 0.34375$ (no factors):

```
l_measured   : 01011 11 frequency: 480
phi_phase_bin: 0.01011
phi_phase_dec: 0.34375
phi: (11, 32)
cont frac of phi  : [0, 2, 1, 10]
convergents of phi: [(0, 1), (1, 2), (1, 3), (11, 32)]
conv: (0, 1) r = 1 : no factors found
conv: (1, 2) r = 2 : no factors found
conv: (1, 3) r = 3 : no factors found
conv: (11, 32) r = 32 : no factors found
```

It is interesting to increase the number of control qubits to $m = 6$. This involves the operator $U^{32}_{2,21}$, which can be

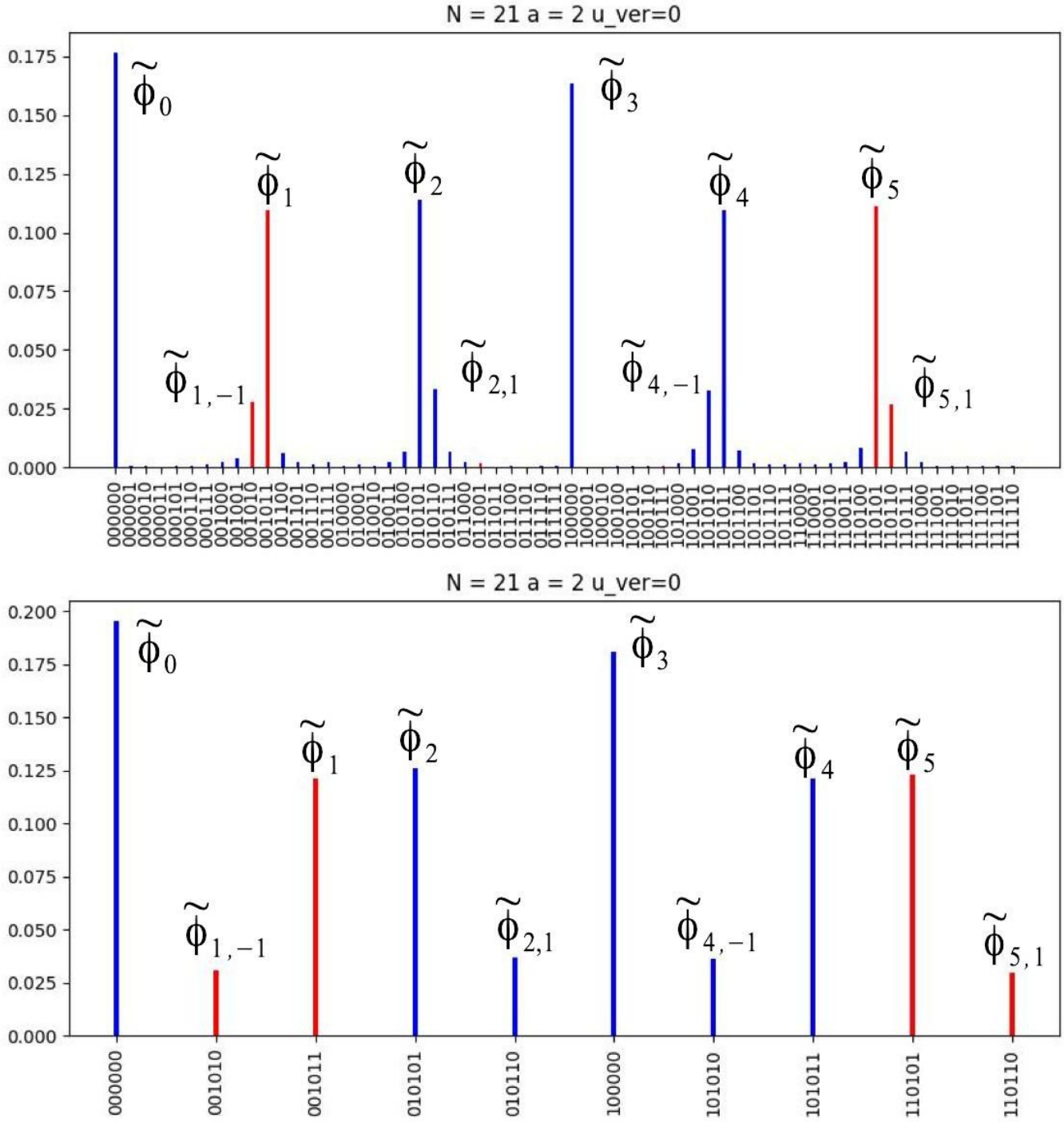


Figure 26: *The corresponding phase histogram for $m = 6$. There is a dominant and sub-dominate peak for each ME phase ϕ_s. Note that sub-dominant peaks can also produce factors.*

constructed from $U_{2,21}$ by simple concatenation, albeit with the price of increasing the number of gates substantially. Figure 26 illustrates the output phase histogram from another Qiskit run with 4096 shots. The top panel of the Figure shows all output phases, while the bottom panel gives only the most likely phases.

The measured phases are peaked near each of the exact phases ϕ_s, and they lie within the resolution $2^{-6} = 0.015625$ of these phases. Note, however, that most phases have a dominant and a sub-dominant peak, labeled by $\tilde{\phi}_\ell$ and $\tilde{\phi}_{\ell,k}$ respectively, where $k = \pm 1$ depending on whether the sub-dominant peak lies to the left or right of the dominant peak. Note that the sub-dominant peaks for $s = 1$ and $s = 5$ also produce factors. The measured values of each of these peaks are given below:

$$\tilde{\ell}_0 \quad = \quad [000000]_2 = 0 \qquad\qquad \tilde{\phi}_0 \quad = [0.000000]_2 = 0.000000 = \phi_0$$

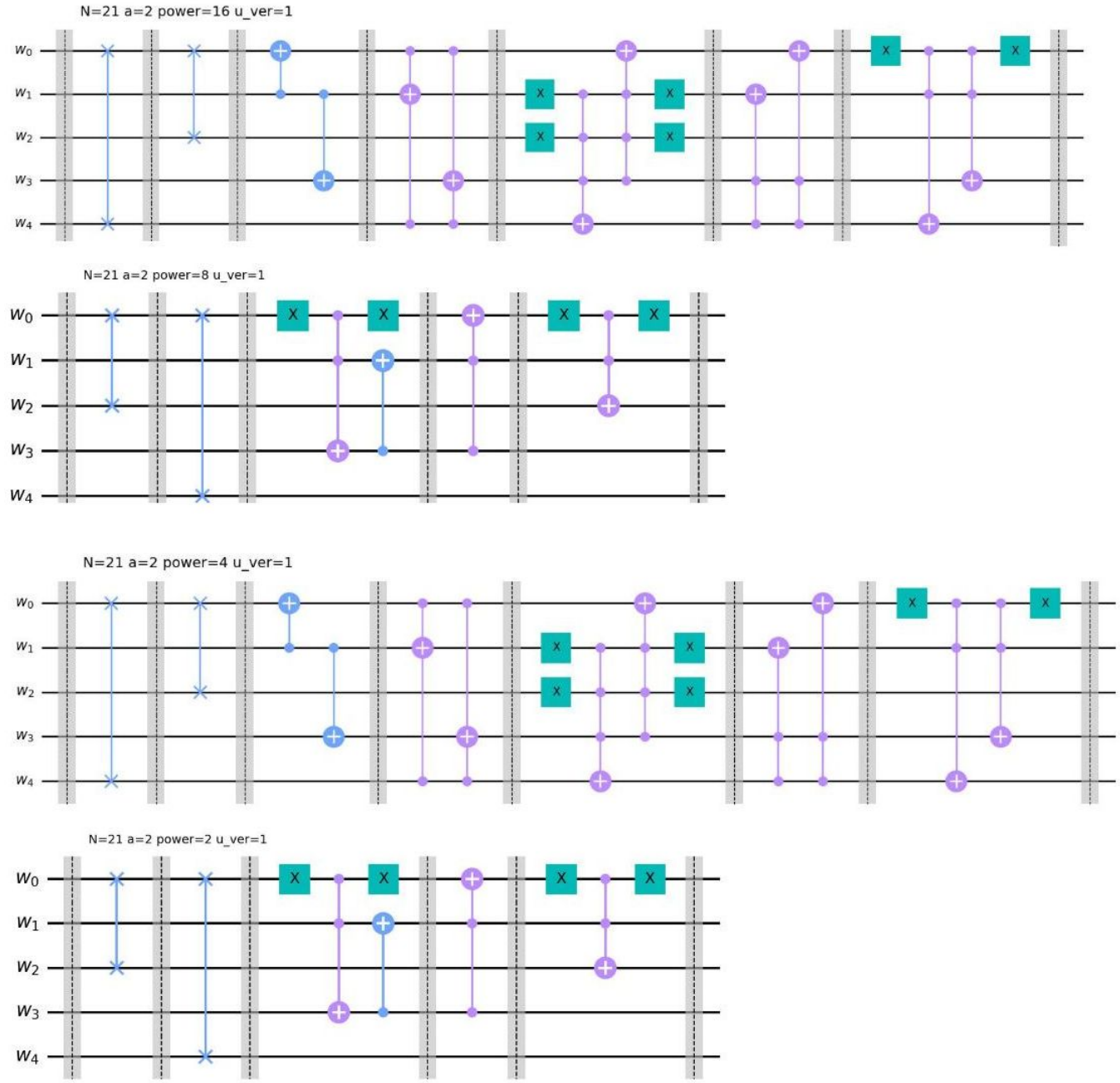


Figure 27: *$N = 21$, $a = 2$, $r = 6$: The ME operators U^2, U^4, U^8 and U^{16} for version* `u_ver` $= 1$ *given in expression (277).*

$$
\begin{array}{rcll}
\tilde{\ell}_{1,-1} & = & [001010]_2 = 10 & \tilde{\phi}_{1,-1} = [0.001010]_2 = 0.156250 \approx \phi_1 \Leftarrow \text{factors} : 3, 7 \\
\tilde{\ell}_1 & = & [001011]_2 = 11 & \tilde{\phi}_1 = [0.001011]_2 = 0.171875 \approx \phi_1 \Leftarrow \text{factors} : 3, 7 \\
\tilde{\ell}_2 & = & [010101]_2 = 21 & \tilde{\phi}_2 = [0.010101]_2 = 0.328125 \approx \phi_2 \\
\tilde{\ell}_{2,1} & = & [010110]_2 = 22 & \tilde{\phi}_{2,1} = [0.010110]_2 = 0.343750 \approx \phi_2 \\
\tilde{\ell}_3 & = & [100000]_2 = 32 & \tilde{\phi}_3 = [0.100000]_2 = 0.500000 = \phi_3 \\
\tilde{\ell}_{4,-1} & = & [101010]_2 = 42 & \tilde{\phi}_{4,-1} = [0.101010]_2 = 0.656250 \approx \phi_4 \\
\tilde{\ell}_4 & = & [101011]_2 = 43 & \tilde{\phi}_4 = [0.101011]_2 = 0.671875 \approx \phi_4 \\
\tilde{\ell}_5 & = & [110101]_2 = 53 & \tilde{\phi}_5 = [0.110101]_2 = 0.828125 \approx \phi_5 \Leftarrow \text{factors} : 3, 7 \\
\tilde{\ell}_{5,1} & = & [110110]_2 = 54 & \tilde{\phi}_{5,1} = [0.110110]_2 = 0.84375 \approx \phi_5 \Leftarrow \text{factors} : 3, 7 \, .
\end{array}
\tag{276}
$$

We are now ready to address the concatenation issue. Let us return to $m = 5$ control qubits. Up to now we have produced U^2, U^4, U^8 and U^{16} by simply concatenating the operator U from Fig. 24. However, we only require the five operators U^p on the 6-dimensional subspace $\mathcal{U}_{r=6} \subseteq \mathcal{W}_{n=5}$. Note that the operator U^2 acts on every other element of the sequence $[1, 2, 4, 8, 16, 11, 1]$, producing two closed sub-sequences $[1, 4, 16, 1]$ and $[2, 8, 11, 2]$.

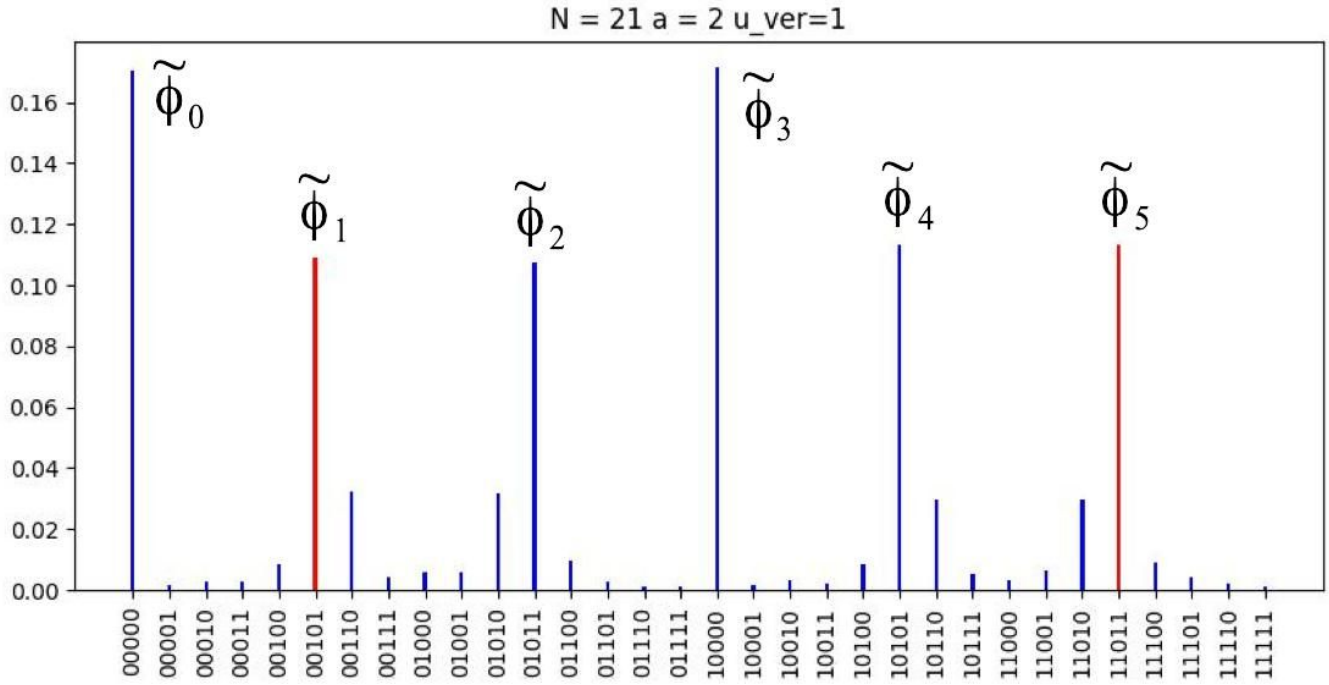


Figure 28: *$N = 21$, $a = 2$, $r = 6$, $m = 5$: Phase histogram for ME operator version* `u_ver` $= 1$ *agrees with the original phase histogram of Fig. 25. As before, the phases that produce factors are shown in red.*

Similarly, U^4 chooses every 4-th element of the sequence and so on, so that the ME operators U^2, U^4, U^8, and U^{16} all act on pairs of closed sub-sequences:

$$\begin{array}{lcl} U_{2,21} & : & [1,2,4,8,16,11,1] \\ U^2_{2,21} & : & [1,4,16,1] \text{ and } [2,8,11,2] \\ U^4_{2,21} & : & [1,16,4,1] \text{ and } [2,11,8,2] \\ U^8_{2,21} & : & [1,4,16,1] \text{ and } [2,8,11,2] \\ U^{16}_{2,21} & : & [1,16,4,1] \text{ and } [2,11,8,2]\,, \end{array} \tag{277}$$

where we have restored the $N = 21$ and $a = 2$ subscripts on the ME operator $U = U_{2,21}$ for clarity. The corresponding circuits that produce these sequences are given in Fig. 27, and we will refer to this collection as version number `u_ver` $= 1$. Figure 28 illustrates the phase histogram from Shor's algorithm with these ME operators. The graph is identical to that of Fig. 25 (as it should be). We see that the phases $\tilde{\phi}_1 = [0.00101]_2 \approx 1/6$ and $\tilde{\phi}_5 = [0.11011]_2 \approx 5/6$ still lie well above the noise, producing the correct factorization.

At this point, one can (and should) levy another charge against this procedure: we have used the *entire* cycle $[1,2,4,8,16,11,1]$ for the ME operator $U_{2,21}$, which means that we know *a priori* that the period of the modular exponential function $f_{2,21}(x)$ is $r = 6$. In other words, if we knew the complete closed-sequence for a general N, then this is equivalent to knowing the period r, so there is no need for Shor's algorithm. However, we do *not* require the complete sequence! This is because when employing the method of continued fractions, it is not necessary to know the *exact* phase, but only an *approximate* phase. Therefore, we require only as much resolution in the phases $\tilde{\phi}_s$ as to extract the corresponding convergents s/r using continued fractions. Figure 29 illustrates a *truncated* version of the operators U, U^2, U^4, U^8, and U^{16}, in which we have omitted several stages from each ME operator U^p. We shall refer to this as version `u_ver` $= 2$. We see from the phase histogram in Fig. 30 that employing these operators in Shor's algorithm still permits one to extract the appropriate phases, and therefore the correct factors. Not surprisingly, the phase histogram has more noise, but this does not overwhelm the signal. We have explored a number of truncation procedures, and they all produce similar results. We will see in the next section that these methods continue to work for even larger values of N.

Before continuing on to larger numbers, however, let us confirm a result derived in the previous sections. We have shown that when $\ell_s = 2^m \phi_s$ is an integer for all $s \in \{0, 1, \cdots, r-1\}$, then the final state amplitudes are non-zero only for $\ell = \ell_s$. In the upper panel of Fig. 31, we see that the modular exponential function for $N = 21$ and $a = 13$

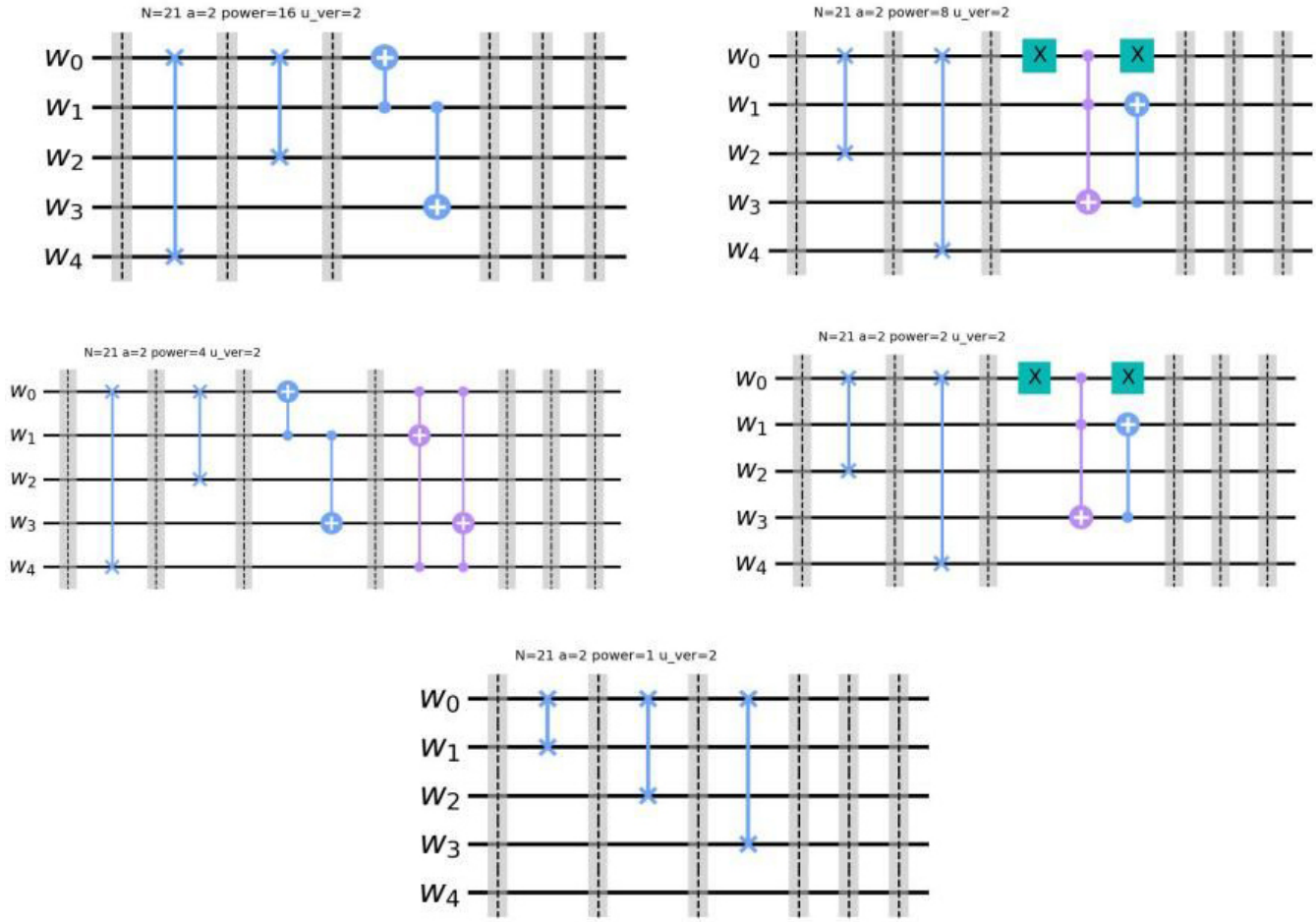


Figure 29: *$N = 21$ and $a = 2$: Truncated ME operators U, U^2, U^4, U^8 and U^{16} for ME operator version* `u_ver` $= 2$.

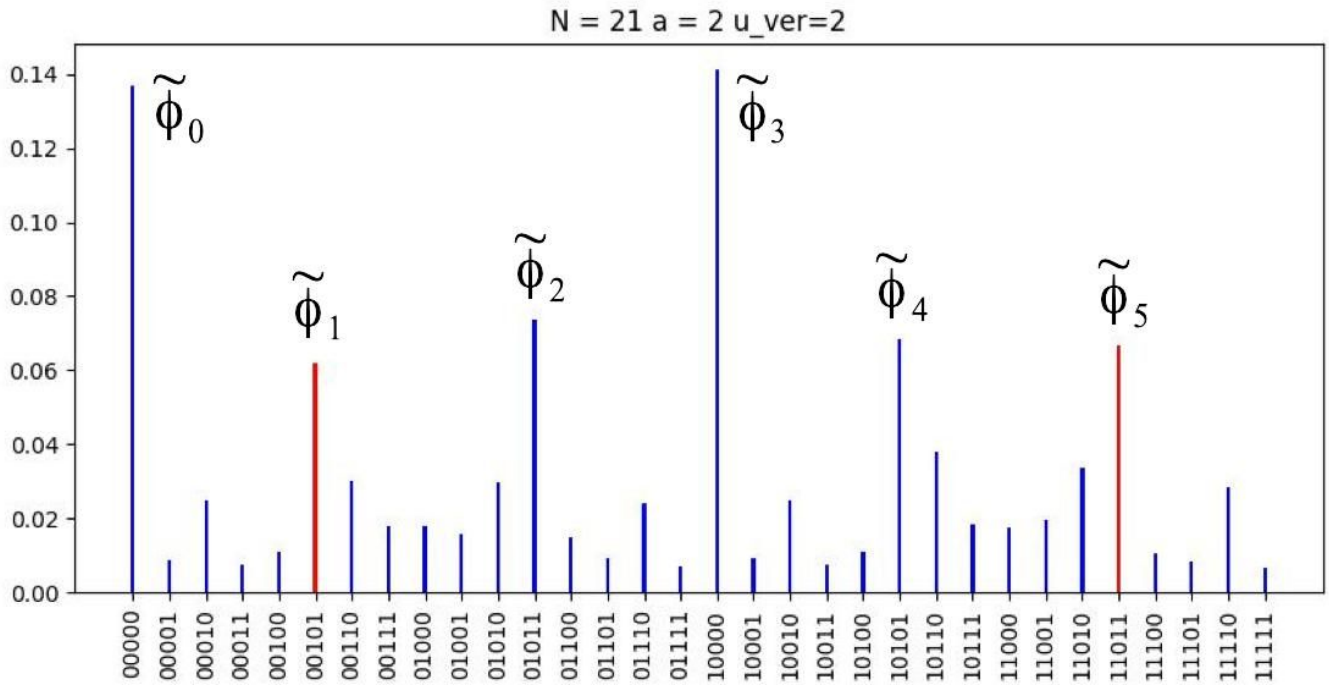
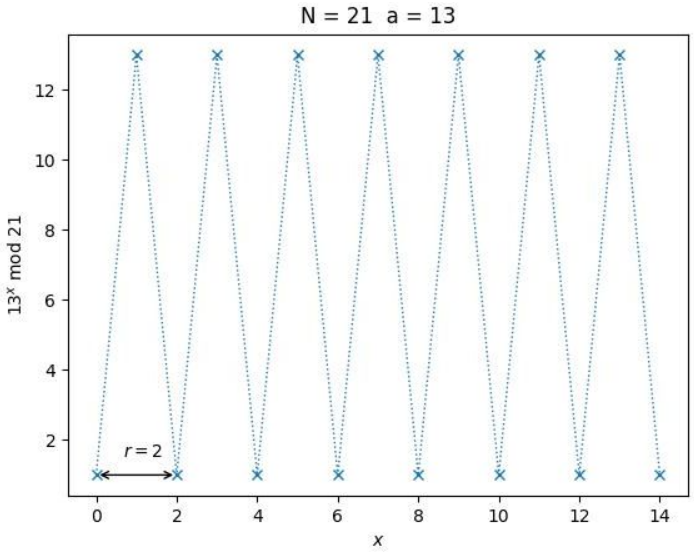


Figure 30: *$N = 21$, $a = 2$, $r = 6$, $m = 5$: Phase histogram for version* `u_ver` $= 2$, *which employs the truncated ME operators in Fig. 29. The signal agrees with the previous two versions, with only slightly more noise, and the peaks in red correspond to phases that produce the factors of 3 and 7.*



$U\|w\rangle =$	$\|14 \cdot w \pmod{21}\rangle$
$U\|1\rangle = \|13\rangle$ $U\|13\rangle = \|1\rangle$	$U\|00001\rangle = \|01101\rangle$ $U\|01101\rangle = \|00001\rangle$

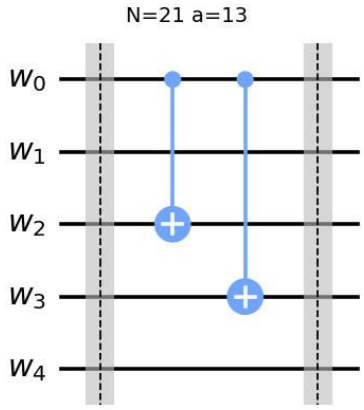


Figure 31: *$N = 21$, $a = 13$, $r = 2$: The left panel illustrates the modular exponential function $f_{2,13}(x) = 2^x \pmod{21}$, while the right panel shows the action of the ME operator $U_{2,13}$ on the closed sequence* $[1, 13, 1]$.

has a period of $r = 2$. The lower panel gives the corresponding ME operator, while Fig. 32 illustrates the phase histogram for $m = 6$. There are exactly two peaks at the values $\tilde{\ell}_0 = 000000$ and $\tilde{\ell}_1 = 100000$, as expected. These peaks correspond to the phase angles $\tilde{\phi}_0 = 0.000000$ and $\tilde{\phi}_1 = 0.100000 = 1/2$, and the latter produces the correct factors of 3 and 7.

7.2. More Factoring

We now turn to factoring even large numbers N. As pointed out in Ref. [7], the difficulty for Shor's algorithm lies not in the size of the number N, but in the length of the period r. As illustrated in Table 5, we have therefore chosen a collection of composite numbers, $N = 21, 35, 33, 143, 247$, and corresponding bases a, that cover a wide range of periods from $r = 6$ to $r = 36$.

Note that we require larger values of m for the control register with increasing period r. We plot the corresponding modular exponential functions $f_{a\,N}(x)$ in Fig. 33 for each value of N and its respective base a. For readability, the plots in Fig. 33 are restricted to small values of the domain variable x. In general, however, the function $f_{a\,N}(x)$ looks highly random over the entire domain of x (although it is not), as illustrated in Fig. 34, where we plot the exponential function $f_{5,143}(x)$ over an extended domain of x-values.

7.2.1. $N = 35 = 5 \times 7$, $a = 4$, $r = 6$

Having examined $N = 21$, we now address $N = 35 = 5 \times 7$ with the base $a = 4$. As illustrated in the top panel of Fig. 35 (and the top-left of Fig 33), these parameters give a modular exponential function $f_{4,35}(x)$ with period $r = 6$,

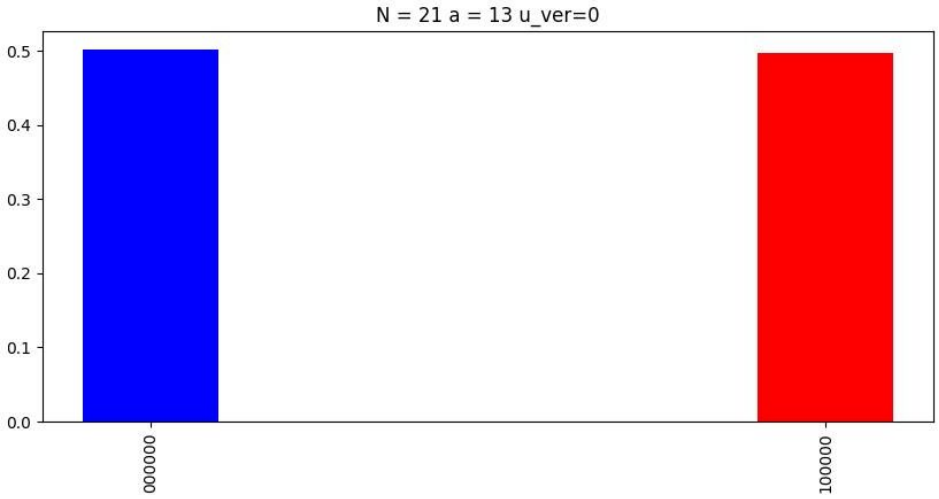


Figure 32: *$N = 21$, $a = 13$, $r = 2$: The corresponding phase histogram for $m = 6$ and* `u_ver` $= 0$*. The red peak produces the factors of 3 and 7.*

just as with $N = 21$ and $a = 2$. We can therefore get by with $m = 5$ control register qubits. However, we must increase the work register to $n = \lceil \log_2 35 \rceil = 6$ qubits. We see that the ME operator $U_{4,35}$ only acts on the closed cycle $[1, 4, 16, 29, 11, 9, 1]$, and the corresponding circuit representation of $U_{4,35}$ is given in the lower panel of Fig 35. Note that qubit w_5 is not used. Finally, the phase histogram of Shor's algorithm from a Qiskit simulation of 4096 runs is presented in Fig. 36. Again, we call this version `u_ver` $= 0$, as the operators $U^p_{4,35}$ are formed by simple concatenation of $U_{4,35}$. The two peaks in red occur at $\tilde{\ell}_1 = [00101]_2$ and $\tilde{\ell}_5 = [11011]_2$, which correspond to the phases $\tilde{\phi}_1 = [0.00101]_2 \approx 1/6$ and $\tilde{\phi}_5 = [0.11011]_2 \approx 5/6$. Each such phase provides the factors of 5 and 7. Note that the two red phase peaks occur at the same values as for $N = 21$ and $a = 2$, although these peaks give different factors because N and a differ.

Recall that by concatenating the ME operator $U_{4,35}$ to form the composite operators $U^p_{4,35}$, we will eventually employ an exponential number of terms, and this procedure will consequently break down for large values of N. As before, we can address this problem by noting that the ME operators $U^p_{4,35}$ for $p = 1, 2, 4, 8, 16$ possess the following closed cycles:

$$\begin{array}{lcl} U_{4,35} & : & [1, 4, 16, 29, 11, 9, 1] \\ U^2_{4,35} & : & [1, 16, 11, 1] \text{ and } [4, 29, 9, 4] \\ U^4_{4,35} & : & [1, 11, 16, 1] \text{ and } [4, 9, 29, 4] \\ U^8_{4,35} & : & [1, 16, 11, 1] \text{ and } [4, 29, 9, 4] \\ U^{16}_{4,35} & : & [1, 11, 16, 1] \text{ and } [4, 9, 29, 4] \, . \end{array} \tag{278}$$

We can now construct operators $U^p_{4,35}$ that reproduce these cycles by concatenating commensurate cycle-pairs together; for example, we must ensure that $U^2_{4,35}$ reproduces the double cycle $[1, 16, 11, 1, 4, 29, 9, 4]$, and we refer to this procedure as version number `u_ver` $= 1$. The composite operators $U^p_{4,35}$ for $p > 1$ are illustrated in Fig. 37, while the corresponding phase histogram from a Qiskit simulation with 4096 runs is given in Fig. 38. We see that the

Table 5: *Composite numbers $N = p \times q$ for primes p and q, together with the corresponding Shor parameters: the work-space length $n = \lceil \log_2 N \rceil$, the base a, the period r of $f_{a,N}(x)$, and the length m of the control register.*

$N = p \times q$	n	a	r	m
$21 = 3 \times 7$	5	2	6	5, 6
$35 = 5 \times 7$	6	4	6	5
$33 = 3 \times 11$	6	7	10	6
$143 = 13 \times 11$	8	5	20	8, 9, 10
$247 = 13 \times 19$	8	2	36	9, 10

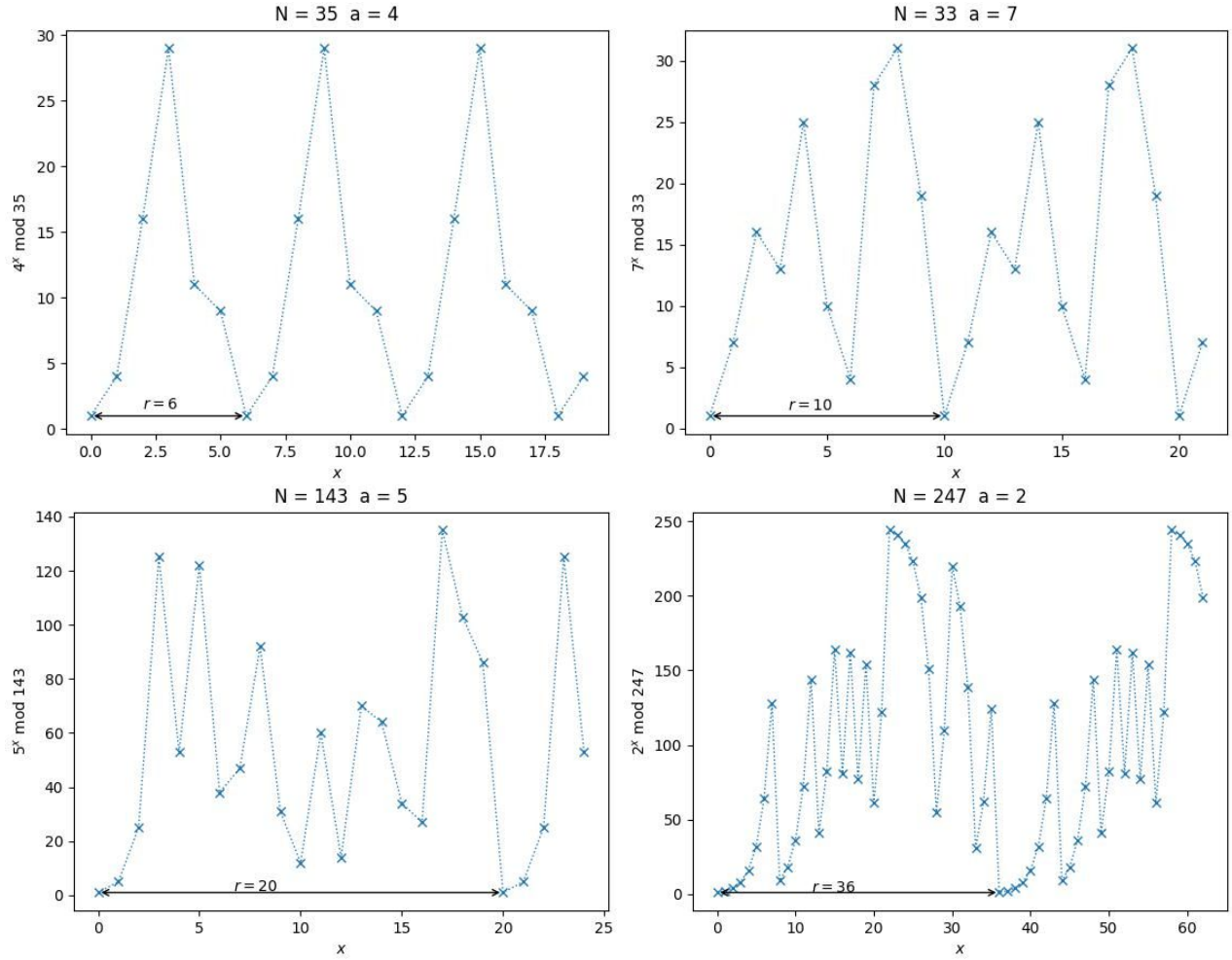


Figure 33: *The modular exponential functions $f_{a,N}(x) = a^x \ (mod\ N)$ for the numbers N and bases a in Table 5. We shall use these in Shor's algorithm: the top left corresponds to $N = 35$ and $a = 4$, the top right is for $N = 33$ and $a = 7$, the bottom left is for $N = 143$ and $a = 5$, and the bottom right is for $N = 247$ and $a = 2$.*

results still agree with the previous version from Fig. 36, in which we concatenated the $U_{4,35}$ operator to form the $U^p_{4,35}$ operators.

Finally, Figs. 39 and 40 illustrates a truncated version of the ME operators $U^p_{4,35}$, with the corresponding phase histogram given in Fig. 41. We see that the truncated operators $U^p_{4,35}$ still provide the correct peaks in the phase histogram, albeit with more noise.

7.2.2. $N = 33 = 3 \times 11$, $a = 7$, $r = 10$

We now move on to factoring a number with a larger period: $N = 33 = 3 \times 11$ with base $a = 7$ has period $r = 10$. It turns out that $m = 6$ control qubits give sufficient resolution for $r = 10$, while the number of work qubits must be set to $n = \lceil \log_2 33 \rceil = 6$, as with the previous example. The circuit representation of the ME operator is shown in Fig. 42, while the corresponding modular exponential function $f_{7,33}(x)$ is plotted in the left panel of Fig. 43, with the action of the ME operator $U_{7,33}$ on the closed sequence $[1, 7, 16, 13, 25, 10, 4, 28, 31, 19, 1]$ given in the right panel. As usual, we form the composite operators U^p for $p > 1$ by concatenation, and call this version `u_ver` $= 0$. The phase histogram for 4096 runs is illustrated in Fig. 44, where the top panel gives the histogram over the full range of phases from the Qiskit simulation, while the bottom panel shows only the most frequent peaks.

Note that the ten dominant peaks in Fig. 44 lie close to the ME Eigen-phases $\phi_s = s/10$ for $s \in \{0, 1, \cdots, 9\}$, as they should. Furthermore, the peaks corresponding to the factors of 3 and 11 occur only when $\gcd(s, 10) = 1$, or for $s = 1, 3, 7, 9$, and they are plotted in red. The phase values of all ten dominant peaks are listed below:

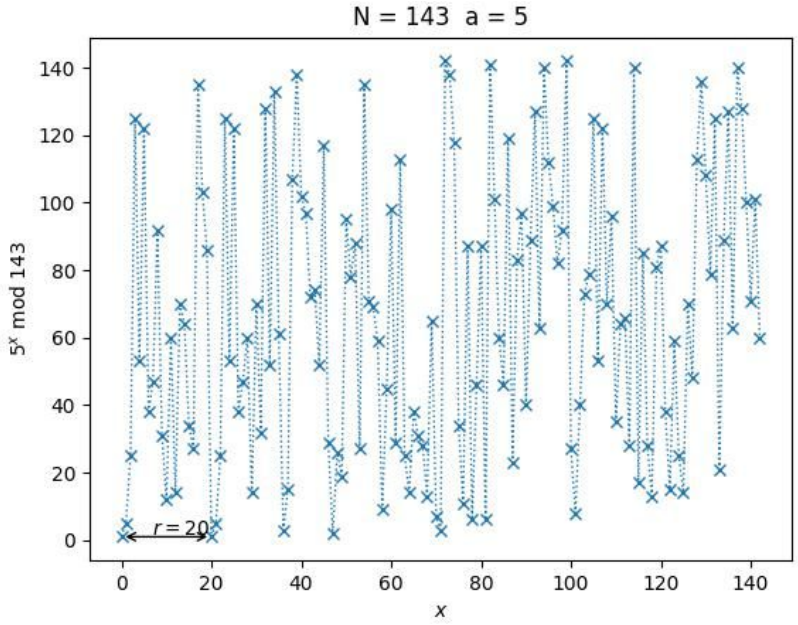
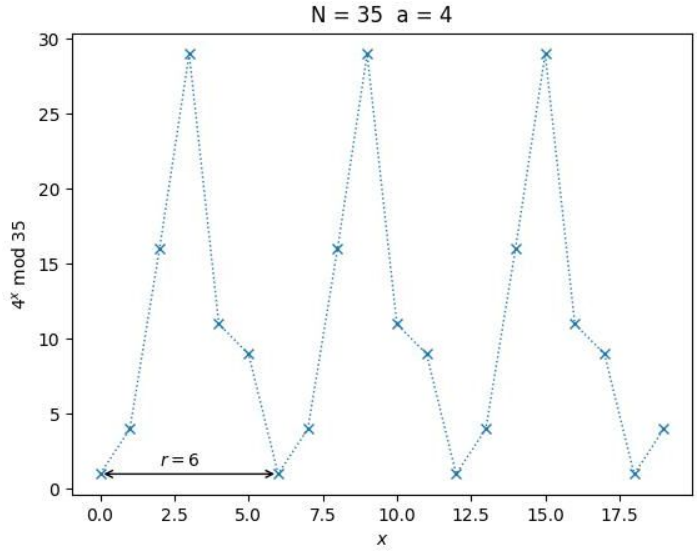


Figure 34: *The range of $f_{5,143}(x)$ has been extended. The function appears random, and the period $r = 20$ is not apparent at first glance.*



$U\|w\rangle =$	$\|4 \cdot w \pmod{35}\rangle$
$U\|1\rangle = \|4\rangle$	$U\|000001\rangle = \|000100\rangle$
$U\|4\rangle = \|16\rangle$	$U\|000100\rangle = \|010000\rangle$
$U\|16\rangle = \|29\rangle$	$U\|010000\rangle = \|011101\rangle$
$U\|29\rangle = \|11\rangle$	$U\|011101\rangle = \|001011\rangle$
$U\|11\rangle = \|9\rangle$	$U\|001011\rangle = \|001001\rangle$
$U\|9\rangle = \|1\rangle$	$U\|001001\rangle = \|000001\rangle$

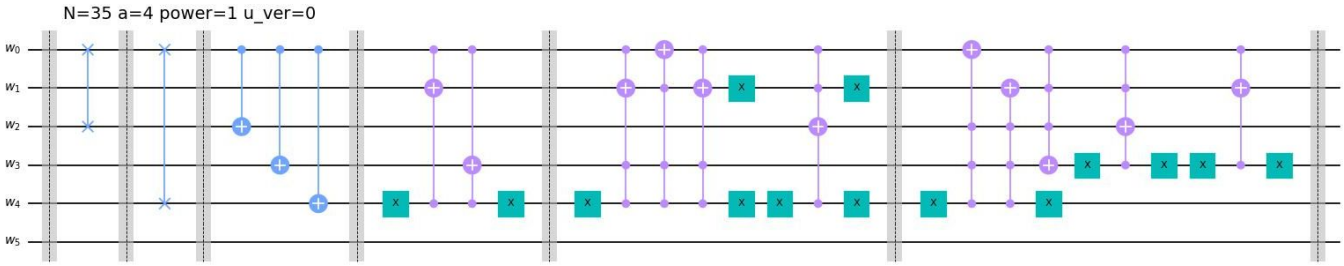


Figure 35: *$N = 35$, $a = 4$, $r = 6$: The top-left panel shows the modular exponential function $f_{4,35}(x) = 4^x \pmod{35}$, which is seen to have period $r = 6$. The top-right panel gives the action of the ME operator $U_{4,35}$ on the closed sequence $[1, 4, 16, 11, 9, 1]$. The bottom panel illustrates the circuit formulation of $U_{4,35}$. Note that qubit w_5 is not used.*

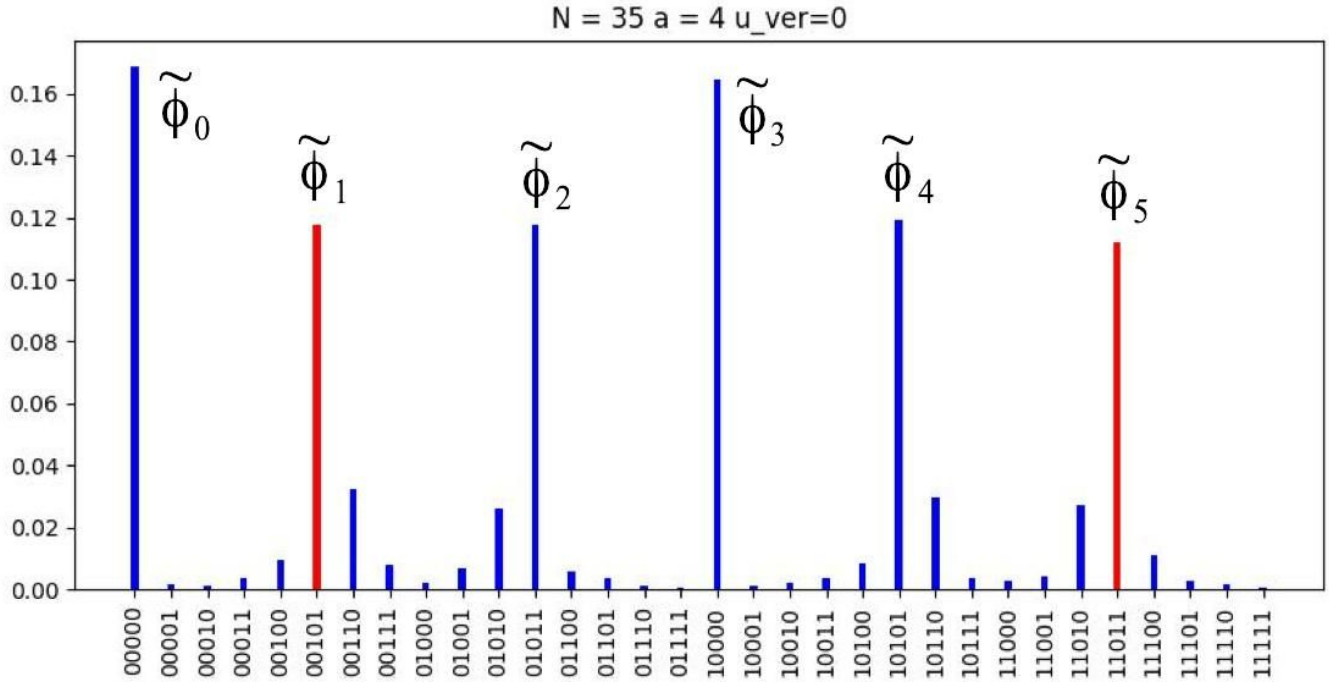


Figure 36: *$N = 35$, $a = 4$, $r = 6$, $m = 5$: The six dominant peaks of the phase histogram occur very close to the six phases $\phi_s = s/6$ of the ME operator $U_{4,35}$, where $s \in \{0, 1, \cdots, 5\}$. The phases that produce factors are shown in red, and and occur at $\tilde{\phi}_1 = [0.00101]_2 \approx 1/6$ and $\tilde{\phi}_5 = [0.11011]_2 \approx 5/6$, each providing the factors of 3 and 7. These peaks are amplified above the noise by Shor's algorithm.*

$$
\begin{aligned}
\tilde{\phi}_0 &= [0.000000]_2 = 0.000000 = 0 \\
\tilde{\phi}_1 &= [0.000110]_2 = 0.093750 \approx 1/10 \quad \Leftarrow \text{ factors : } 3, 11 \\
\tilde{\phi}_2 &= [0.001101]_2 = 0.203125 \approx 2/10 \\
\tilde{\phi}_3 &= [0.010011]_2 = 0.296875 \approx 3/10 \quad \Leftarrow \text{ factors : } 3, 11 \\
\tilde{\phi}_4 &= [0.011010]_2 = 0.406250 \approx 4/10 \\
\tilde{\phi}_5 &= [0.100000]_2 = 0.500000 = 5/10 \\
\tilde{\phi}_6 &= [0.100110]_2 = 0.593750 \approx 6/10 \\
\tilde{\phi}_7 &= [0.101101]_2 = 0.703125 \approx 7/10 \quad \Leftarrow \text{ factors : } 3, 11 \\
\tilde{\phi}_8 &= [0.110011]_2 = 0.796875 \approx 8/10 \\
\tilde{\phi}_9 &= [0.111010]_2 = 0.906250 \approx 9/10 \quad \Leftarrow \text{ factors : } 3, 11 \, .
\end{aligned}
\tag{279}
$$

Let us now construct the composite operators U^p for $p = 2^0, 2^1, \cdots, 2^5$. Note that these operators possess the following cycles:

$$
\begin{aligned}
U_{7,33} &: [1, 7, 16, 13, 25, 10, 4, 28, 31, 19, 1] \\
U^2_{7,33} &: [1, 16, 25, 4, 31, 1] \text{ and } [7, 13, 10, 28, 19, 7] \\
U^4_{7,33} &: [1, 25, 31, 16, 4, 1] \text{ and } [7, 10, 19, 13, 28, 7] \\
U^8_{7,33} &: [1, 31, 4, 25, 16, 1] \text{ and } [7, 19, 28, 10, 13, 7] \\
U^{16}_{7,33} &: [1, 4, 16, 31, 25, 1] \text{ and } [7, 28, 13, 19, 10, 7] \\
U^{32}_{7,33} &: [1, 16, 25, 4, 31, 1] \text{ and } [7, 13, 10, 28, 19, 7] \, .
\end{aligned}
\tag{280}
$$

The operators U^p for $p > 1$ are given in Figs. 45 and 46, and the corresponding phase histogram for 4096 runs is shown in Fig. 47. We will call this version `u_ver` $= 1$, and it agrees with the previous result.

As we have seen, the choice of ME operators seems to be rather forgiving, as long as they encode sufficient correlations to yield an approximate phase for which the continued fractions algorithm can be employed. For example,

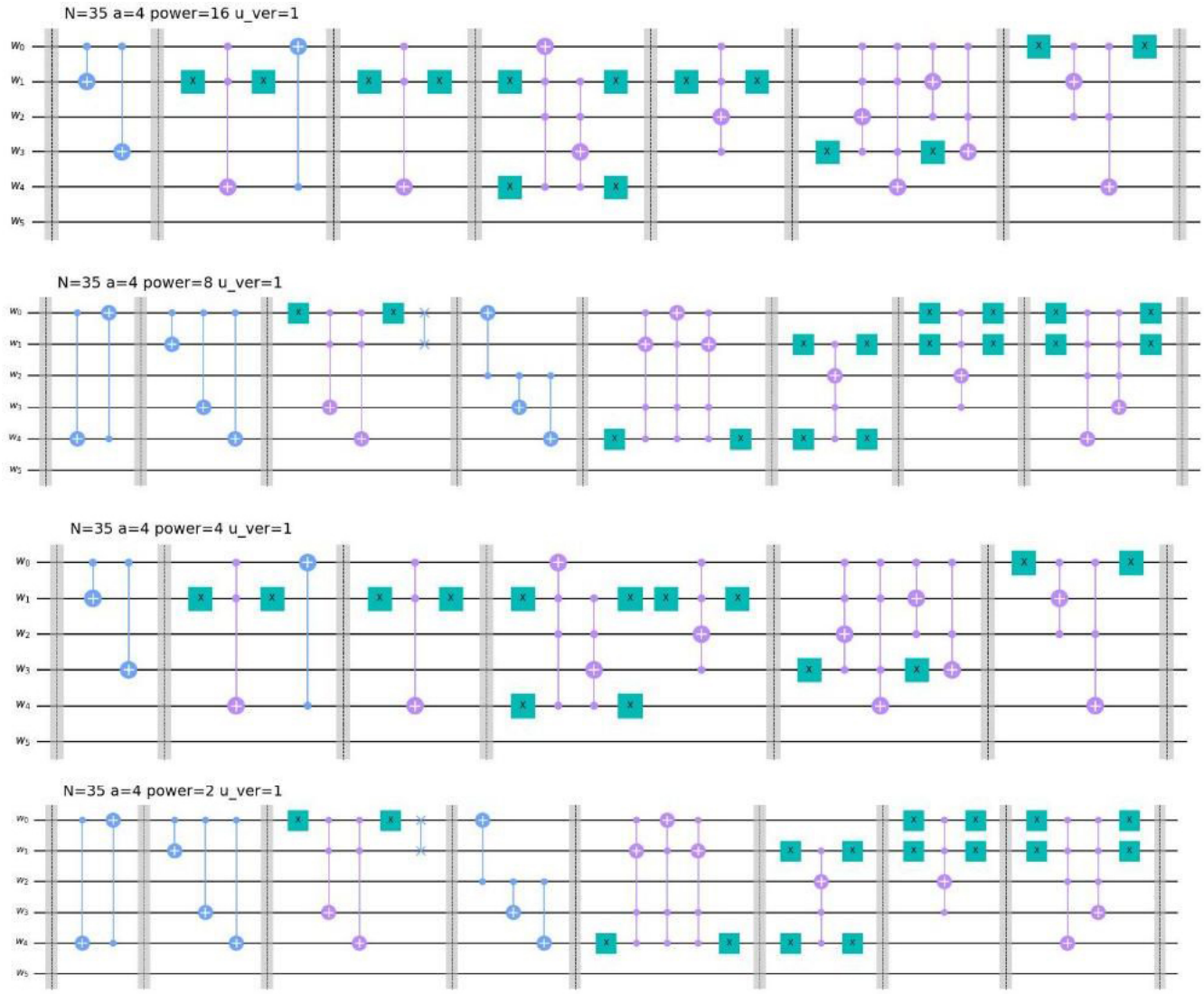


Figure 37: *$N = 35$, $a = 4$, $r = 6$: The ME operators U^2, U^4, U^8 and U^{16} for version* `u_ver` $= 1$.

consider the case in which we ignore half of the cycles in the previous version:

$$\begin{aligned} U_{7,33} &: [1,7,16,13,25,10,4] \\ U^2_{7,33} &: [1,16,25,4,31,1] \\ U^4_{7,33} &: [1,25,31,16,4,1] \\ U^8_{7,33} &: [1,31,4,25,16,1] \\ U^{16}_{7,33} &: [1,4,16,31,25,1] \\ U^{32}_{7,33} &: [1,16,25,4,31,1]\,. \end{aligned} \tag{281}$$

We shall call this version `u_ver` $= 2$. The operators are given in Figs. 48 and 49, and the phase histogram given by Fig. 50 agrees with the previous results (although there is a bit more noise). The lesson here is that much freedom is permitted when constructing the ME operators.

7.2.3. $N = 143 = 13 \times 11$, $a = 5$, $r = 20$

The next number we shall factor is $N = 143 = 11 \times 13$. As illustrated in the left panel of Fig. 51, the base $a = 5$ gives a modular exponential function $f_{5,143}(x)$ with a period of $r = 20$. The work register must have $n = \lceil \log_2 143 \rceil = 8$ qubits, and the corresponding ME operator $U_{5,143}$ is given in Fig. 52. We will perform a resolution study on the control register by taking $m = 8, 9, 10$. We will start our analysis with $m = 8$, so we must implement the ME operators $U^p_{5,143}$ for $p = 2^0, 2^1, \cdots, 2^7$, *i.e.* we require the operators $U_{5,143}, U^2_{5,143}, U^4_{5,143}, U^8_{5,143}, U^{16}_{5,143}, U^{32}_{5,143}, U^{64}_{5,143}$ and

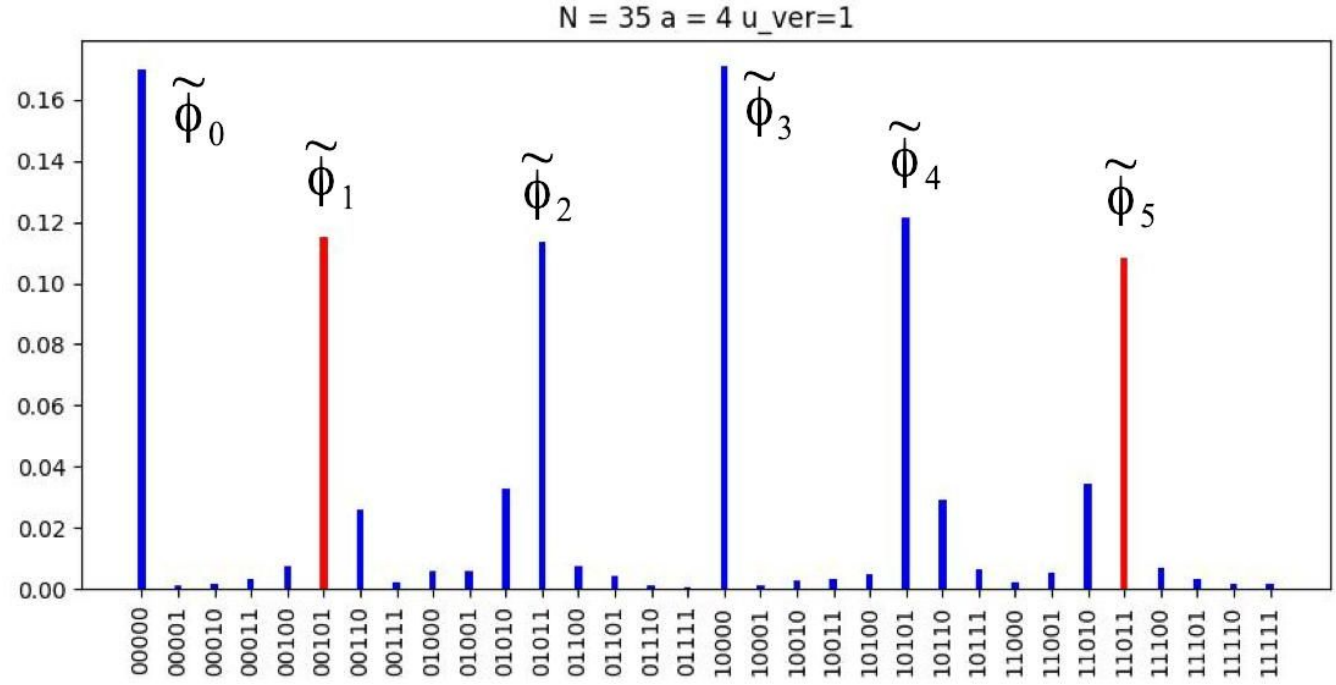


Figure 38: *$N = 35$, $a = 4$, $r = 6$, $m = 5$: Phase histogram for ME operator version* `u_ver` $= 1$ *from Fig. 37.*

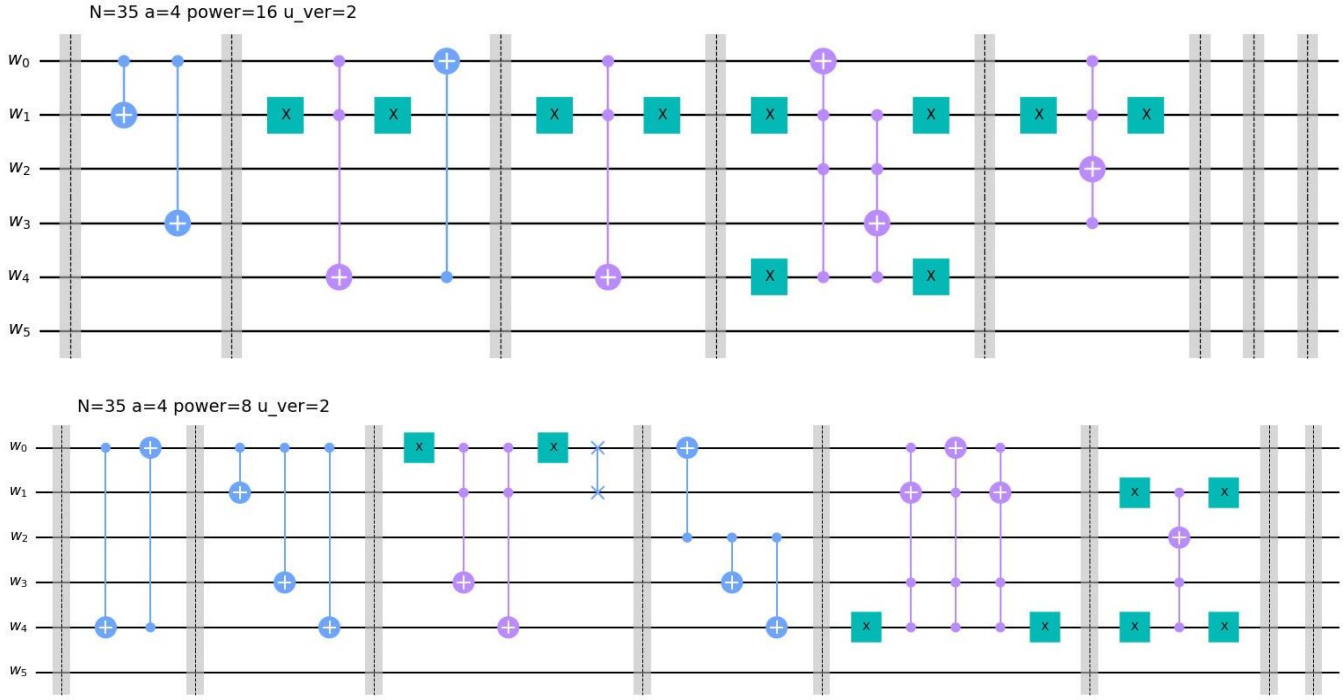


Figure 39: *$N = 35$, $a = 4$, $r = 6$: The ME operators U^8, U^{16} for version* `u_ver` $= 2$.

$U^{128}_{5,143}$. For $m = 9$ control qubits, we will also require the operator $U^{256}_{5,143}$, and for $m = 10$ we must implement $U^{512}_{5,143}$. As always, we refer to the concatenated operators by `u_ver` $= 0$. The phase histogram for $m = 8$ is given in Fig. 53. The top panel of the Figure gives the histogram over the full range of phases, while the bottom panel only plots the most frequent phases, with red phases providing factors.

The 20 phases of the ME operator $U_{5,143}$ are supposed to occur at $\phi_s = s/20$ for $s \in \{0, 1, \cdots, 19\}$, with the factors coming from the phases for which $\gcd(s, 20) = 1$, that is to say, at the eight phases $\phi_s = 1/20, 3/20, 7/20, 9/20, 11/20,$

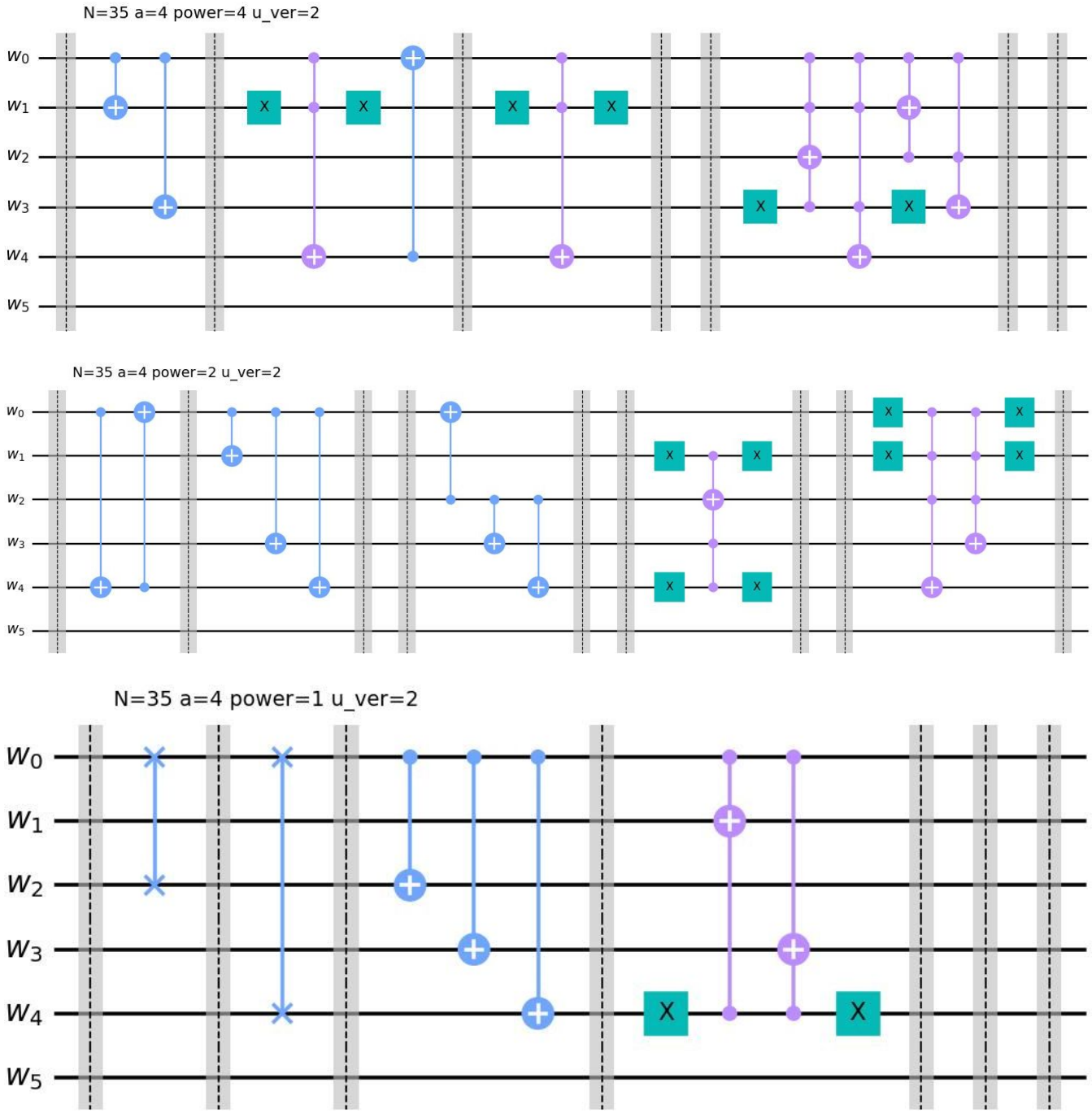


Figure 40: *$N = 35$, $a = 4$, $r = 6$: The ME operators U, U^2, U^4 for version* `u_ver` $= 2$.

13/20, 17/20, 19/20. Note, however, that the red peaks of the phase histogram only contain six of the eight phases:

$$\tilde{\phi}_1 = [0.00001101]_2 = 0.05078125 \approx \phi_1 = 1/20 \quad (282)$$
$$\tilde{\phi}_3 = [0.00100110]_2 = 0.14843750 \approx \phi_3 = 3/20 \quad (283)$$
$$\tilde{\phi}_9 = [0.01110011]_2 = 0.44921875 \approx \phi_9 = 9/20 \quad (284)$$
$$\tilde{\phi}_{11} = [0.10001101]_2 = 0.55078125 \approx \phi_{11} = 11/20 \quad (285)$$
$$\tilde{\phi}_{17} = [0.11011010]_2 = 0.85156250 \approx \phi_{17} = 17/20 \quad (286)$$
$$\tilde{\phi}_{19} = [0.11110011]_2 = 0.94921875 \approx \phi_{19} = 19/20\,. \quad (287)$$

The two phases corresponding to $\phi_7 = 7/20$ and $\phi_{13} = 13/20$ are absent. This is actually a resolution problem: when we increase the control register to $m = 9$ qubits, we obtain all eight phases, as the phase histogram of Fig. 54 reveals.

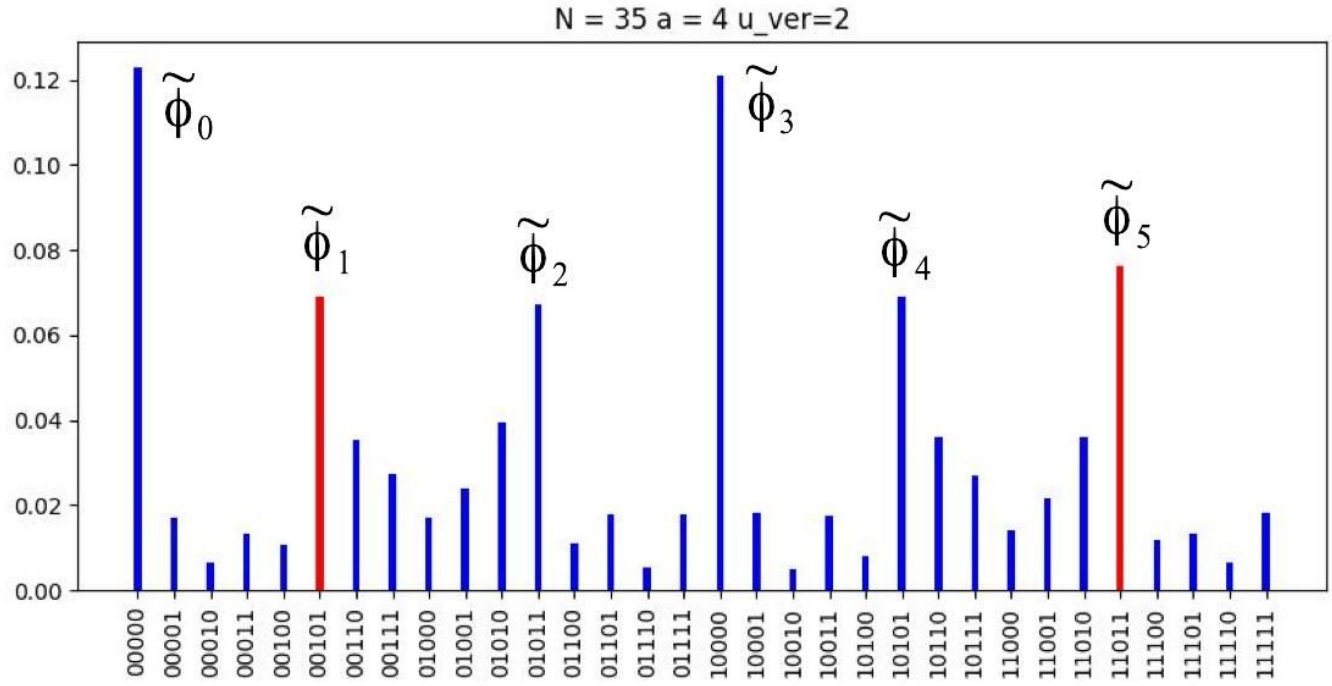


Figure 41: *$N = 35$, $a = 4$, $r = 6$, $m = 5$: Phase histogram for ME operator version* `u_ver` $= 2$.

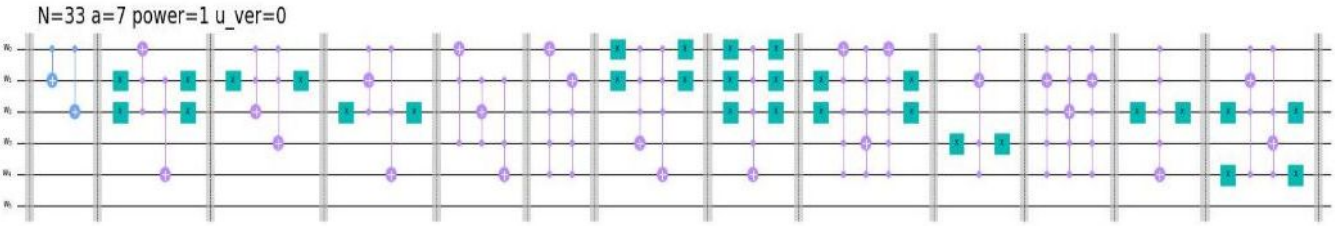
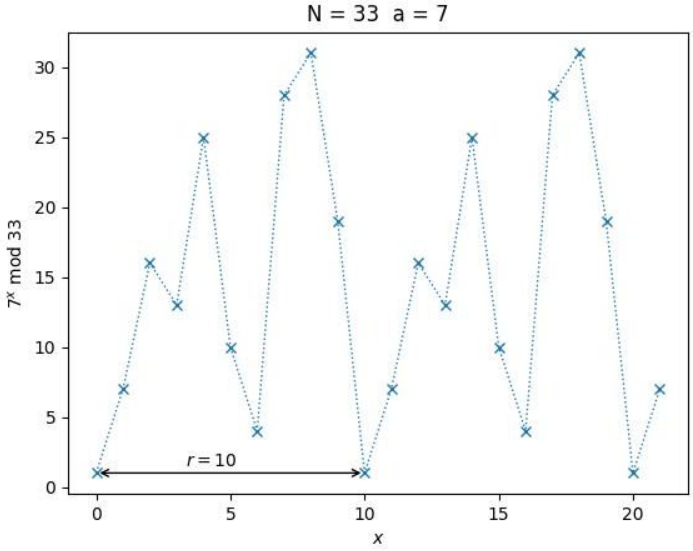


Figure 42: *The circuit formulation of $U_{7,33}$.*



$U\|w\rangle =$	$\|7 \cdot w \pmod{33}\rangle$
$U\|1\rangle = \|7\rangle$	$U\|000001\rangle = \|000111\rangle$
$U\|7\rangle = \|16\rangle$	$U\|000111\rangle = \|010000\rangle$
$U\|16\rangle = \|13\rangle$	$U\|010000\rangle = \|001101\rangle$
$U\|13\rangle = \|25\rangle$	$U\|001101\rangle = \|011001\rangle$
$U\|25\rangle = \|10\rangle$	$U\|011001\rangle = \|001010\rangle$
$U\|10\rangle = \|4\rangle$	$U\|001010\rangle = \|000100\rangle$
$U\|4\rangle = \|28\rangle$	$U\|000100\rangle = \|011100\rangle$
$U\|28\rangle = \|31\rangle$	$U\|011100\rangle = \|011111\rangle$
$U\|31\rangle = \|19\rangle$	$U\|011111\rangle = \|010011\rangle$
$U\|19\rangle = \|1\rangle$	$U\|010011\rangle = \|000001\rangle$

Figure 43: *$N = 33$, $a = 7$, $r = 10$: The left panel gives the modular exponential function $f_{7,33}(x) = 7^x \pmod{33}$, and the right panel shows the action of the ME operator $U_{7,33}$ on the closed sequence* $[1, 7, 16, 13, 25, 10, 4, 28, 31, 19, 1]$.

Note that sub-dominant peaks have appeared, and they too can provide factors. The situation is even more dramatic for $m = 10$, where further sub-dominant peaks emerge, as illustrated in the phase histogram in Fig. 55.

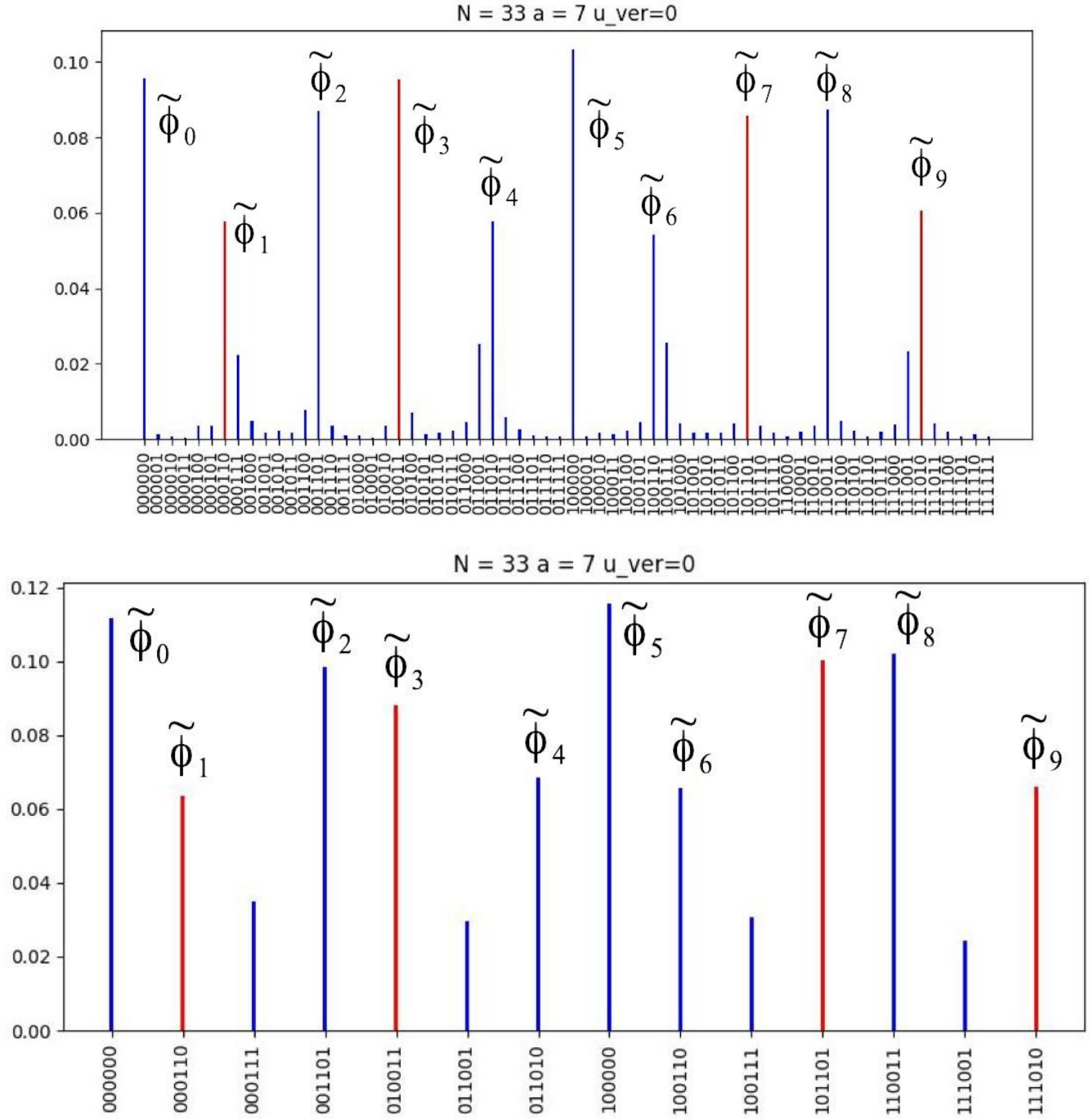


Figure 44: *$N = 33$, $a = 7$, $r = 10$, $m = 6$: Phase histogram for the ME operator of version* `u_ver` $= 0$. *The top panel illustrates the histogram over the full range of phases, while the bottom panel shows only the most frequent peaks.*

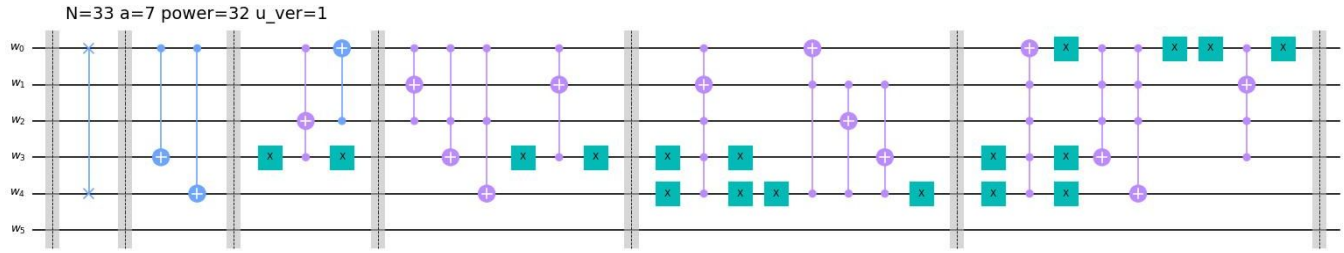


Figure 45: *$N = 33$, $a = 7$, $r = 10$: The ME operators U^{32} for version* `u_ver` $= 1$.

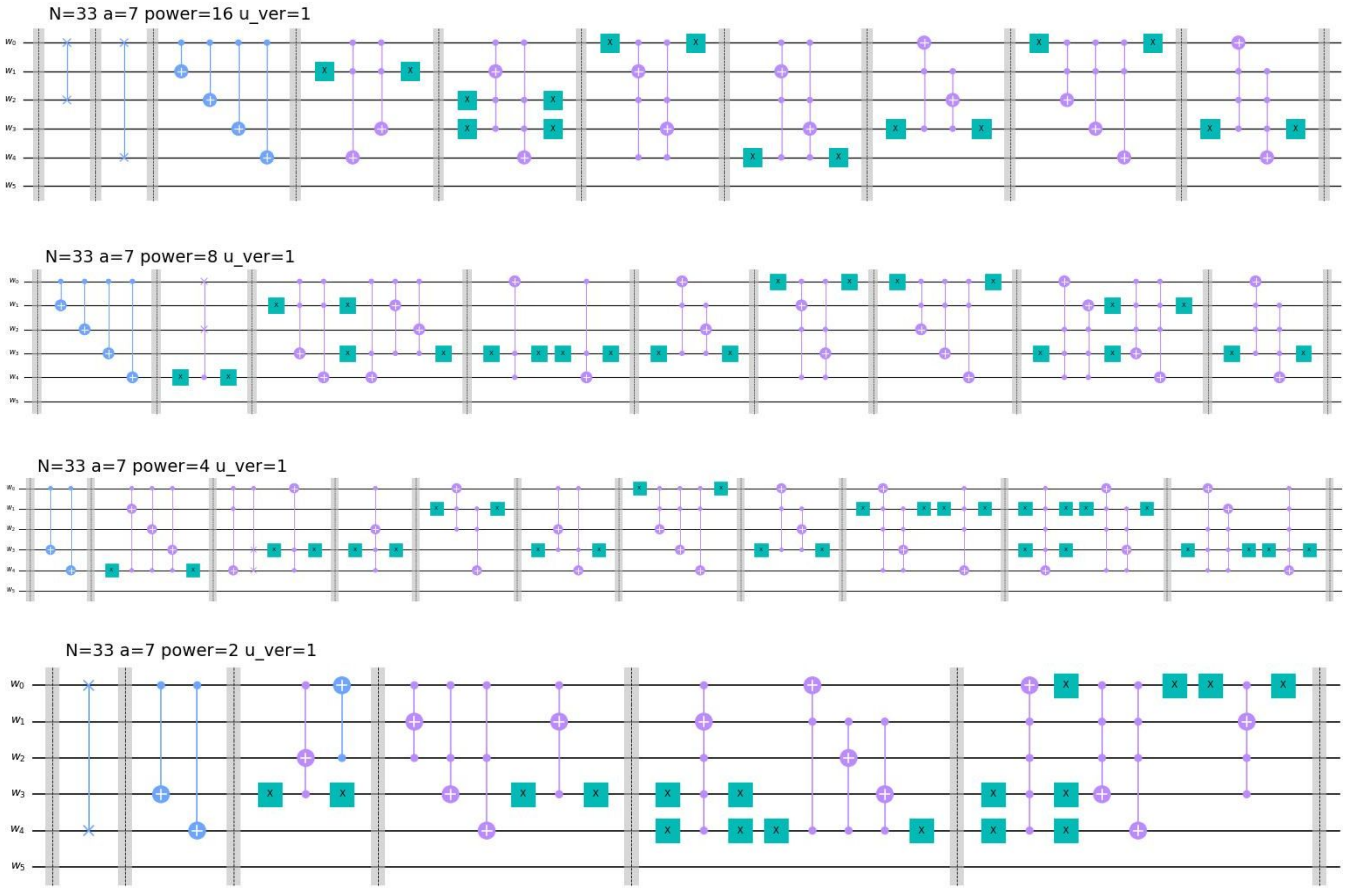


Figure 46: *$N = 33$, $a = 7$, $r = 10$: The ME operators U^2, U^4, U^8 and U^{16} for version* `u_ver` $= 1$.

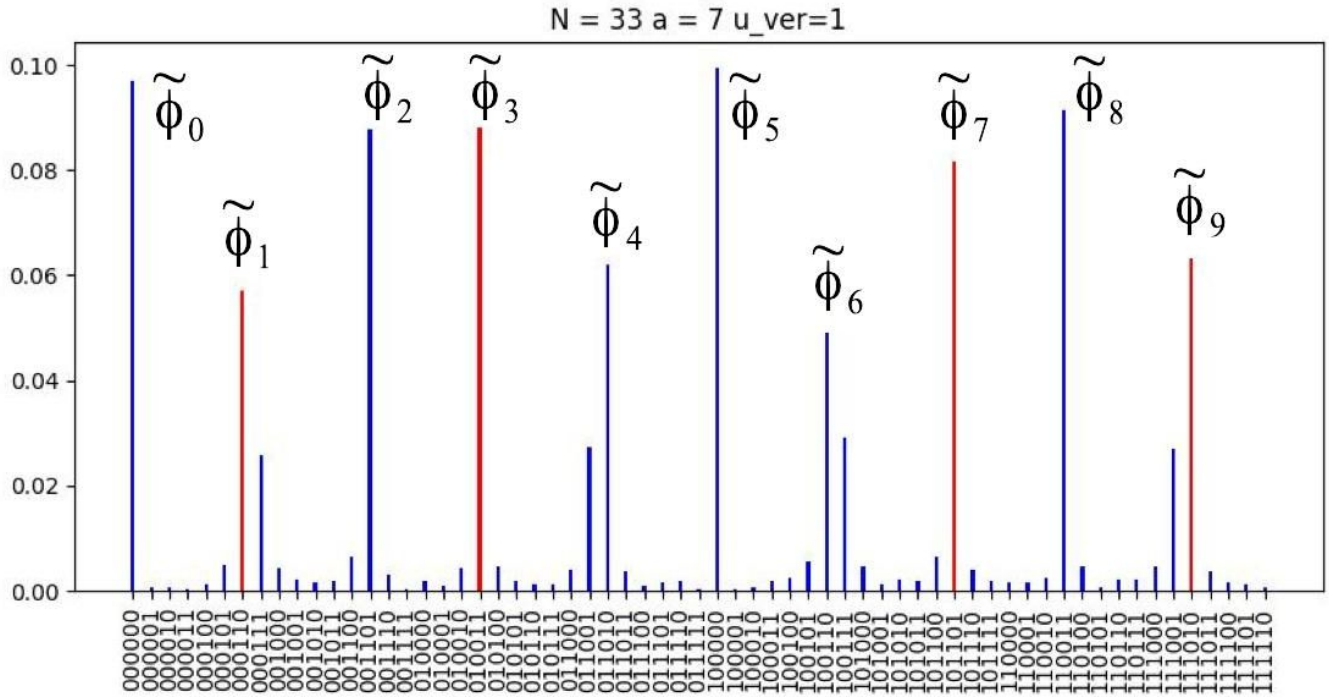


Figure 47: *$N = 33$, $a = 7$, $r = 10$, $m = 6$: Phase histogram for ME operator version* `u_ver` $= 1$.

Finally, let us turn to constructing the composite operators U^p without using concatenation. These operators have

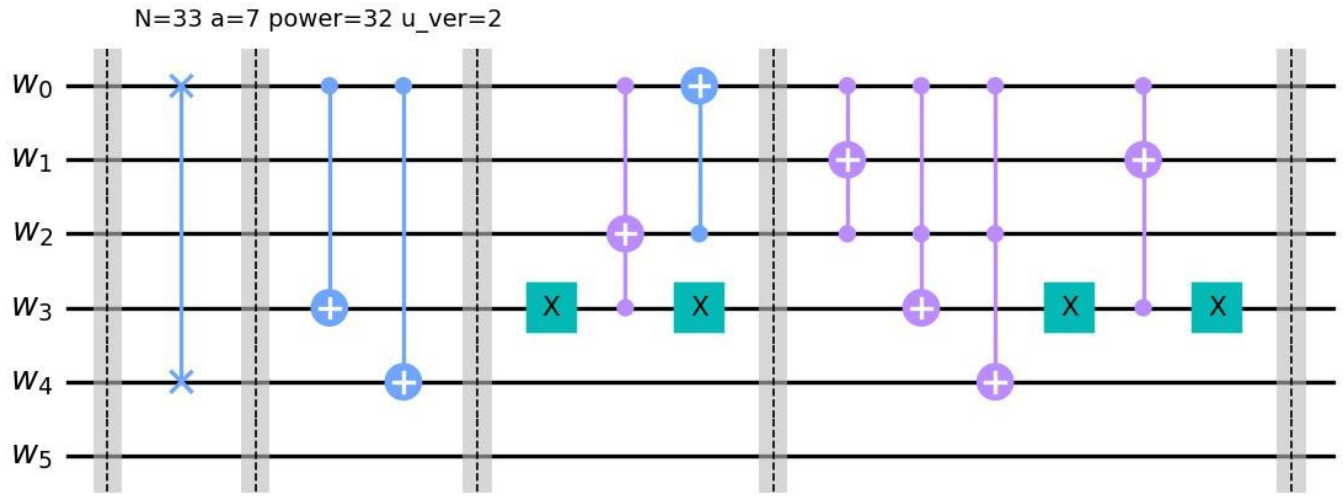


Figure 48: *$N = 33$, $a = 7$, $r = 10$: The ME operators U^{32} for version* `u_ver` $= 2$.

the following pairs of cycles:

$$
\begin{aligned}
U_{5,143} &: [1,5,25,125,53,122,38,47,92,31,12,60,14,70,64,34,27,135,103,86,1] \\
U^2_{5,143} &: [1,25,53,38,92,12,14,64,27,103,1] \text{ and } [5,125,122,47,31,60,70,34,135,86,5] \\
U^4_{5,143} &: [1,53,92,14,27,1] \text{ and } [5,122,31,70,135,5] \\
U^8_{5,143} &: [1,92,27,53,14,1] \text{ and } [5,31,135,122,70,5] \\
U^{16}_{5,143} &: [1,27,14,92,53,1] \text{ and } [5,135,70,31,122,5] \\
U^{32}_{5,143} &: [1,14,53,27,92,1] \text{ and } [5,70,122,135,31,5] \\
U^{64}_{5,143} &: [1,53,92,14,27,1] \text{ and } [5,122,31,70,135,5] \\
U^{128}_{5,143} &: [1,92,27,53,14,1] \text{ and } [5,31,135,122,70,5]\,.
\end{aligned}
\tag{288}
$$

For simplicity we have not included all possible cycles, and we will refer to the procedure given by (288) as version number `u_ver` $= 1$ (so this can also be regarded as a truncated version of the ME operators). The composite operators U^p are given in Figs. 61 and 62 in Appendix A.1, and the corresponding phase histogram from a Qiskit simulation with 4096 runs is given in Fig. 56. As usual, the top panel plots all output phases, and the bottom panel plots only the most frequent ones. Note that the noise in the top Figure has increased significantly, but the signal still dominates.

7.2.4. $N = 247 = 13 \times 19$, $a = 2$, $r = 36$

As our last example, let us factor $N = 247$ into 13 and 19. For the base $a = 2$, the top-left panel of Fig. 57 shows that the period of the modular exponential function $f_{2,247}(x)$ is $r = 36$. For this period, $m = 9$ control qubits is sufficient to resolve the phase difference $\Delta\phi = 1/36$. Also note that we require $n = \lceil \log_2 246 \rceil = 8$ work qubits. The action of the ME operator $U_{2,247}$ on the work state $|1\rangle = |00000001\rangle$ is illustrated top-right panel of Fig. 57, and its circuit representation is given in the bottom panel of the Figure. Since $m = 8$, we shall also require the ME operators $U^p_{2,247}$ for $p = 1, 2, 4, \cdots, 256$. As usual, we can construct these operators by concatenating $U_{2,247}$, and we will refer to this as version number `u_ver` $= 0$. The phase histogram from Shor's algorithm is illustrated in Fig. 58. The phases of the $U_{2,247}$ operator that provide factors are supposed to occur at $\phi_s = s/36$ for $s \in \{0, 1, \cdots, 35\}$, where $r = 36$ and s have no non-trivial common factors. This gives 12 possible phases: $\phi_s = 1/36, 5/36, 7/36, 11/36, 13/36, 17/36, 19/36, 23/36, 25/36, 29/36, 31/36, 35/36$. The phase histogram in Fig. 58 exhibits eight of these phases. As before, the top panel shows every phase from the simulation, while the bottom panel gives only the most frequent phases. However, if we increase the phase resolution to $m = 10$ control qubits, Fig 59 shows that we capture all 12 possible phases.

Let us examine the phase histogram of Fig. 58 in a bit more detail. Note that the phases that produce factors of

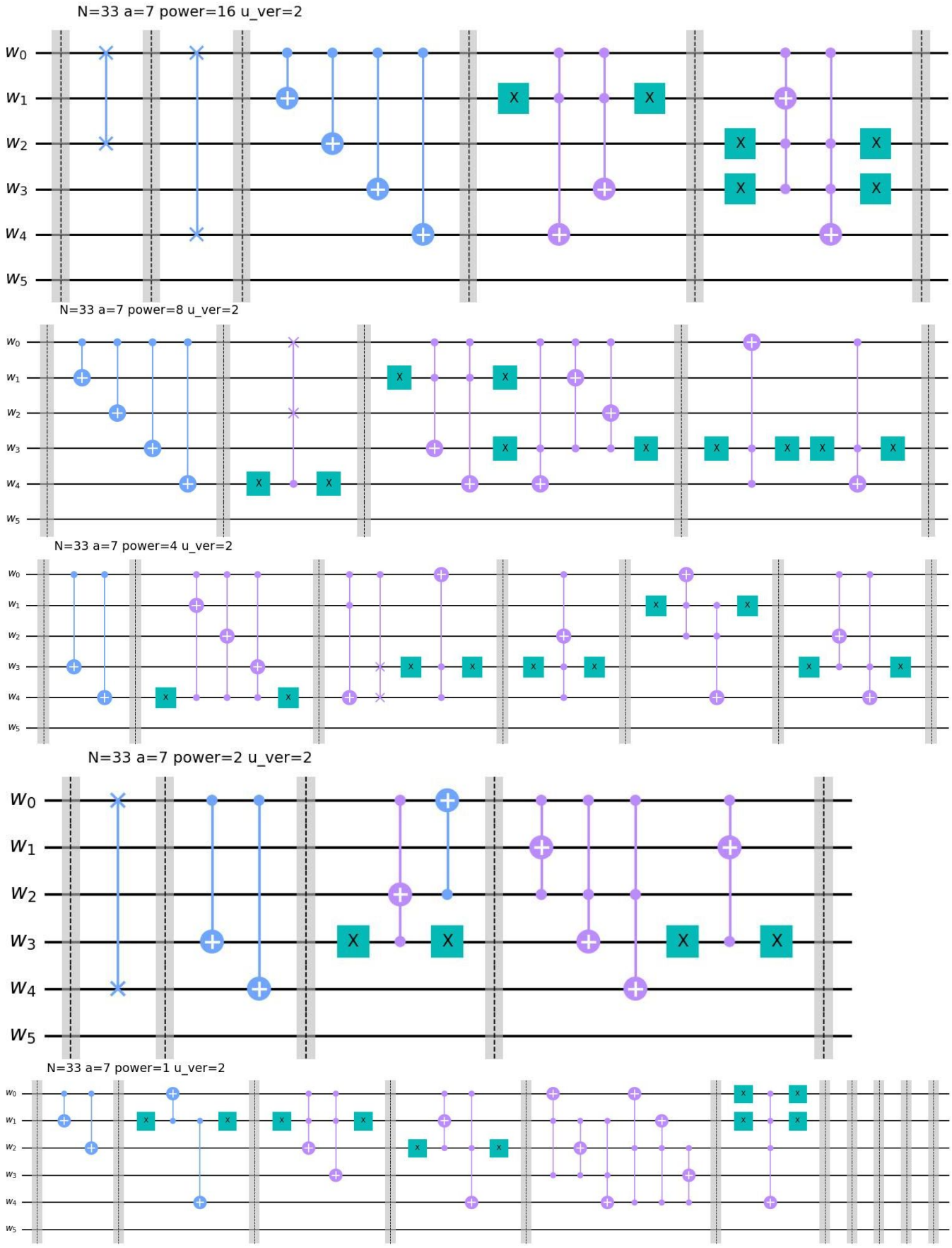


Figure 49: *$N = 33$, $a = 7$, $r = 10$: The ME operators U, U^2, U^4, U^8, U^{16} and U^{32} for version* `u_ver` $= 2$.

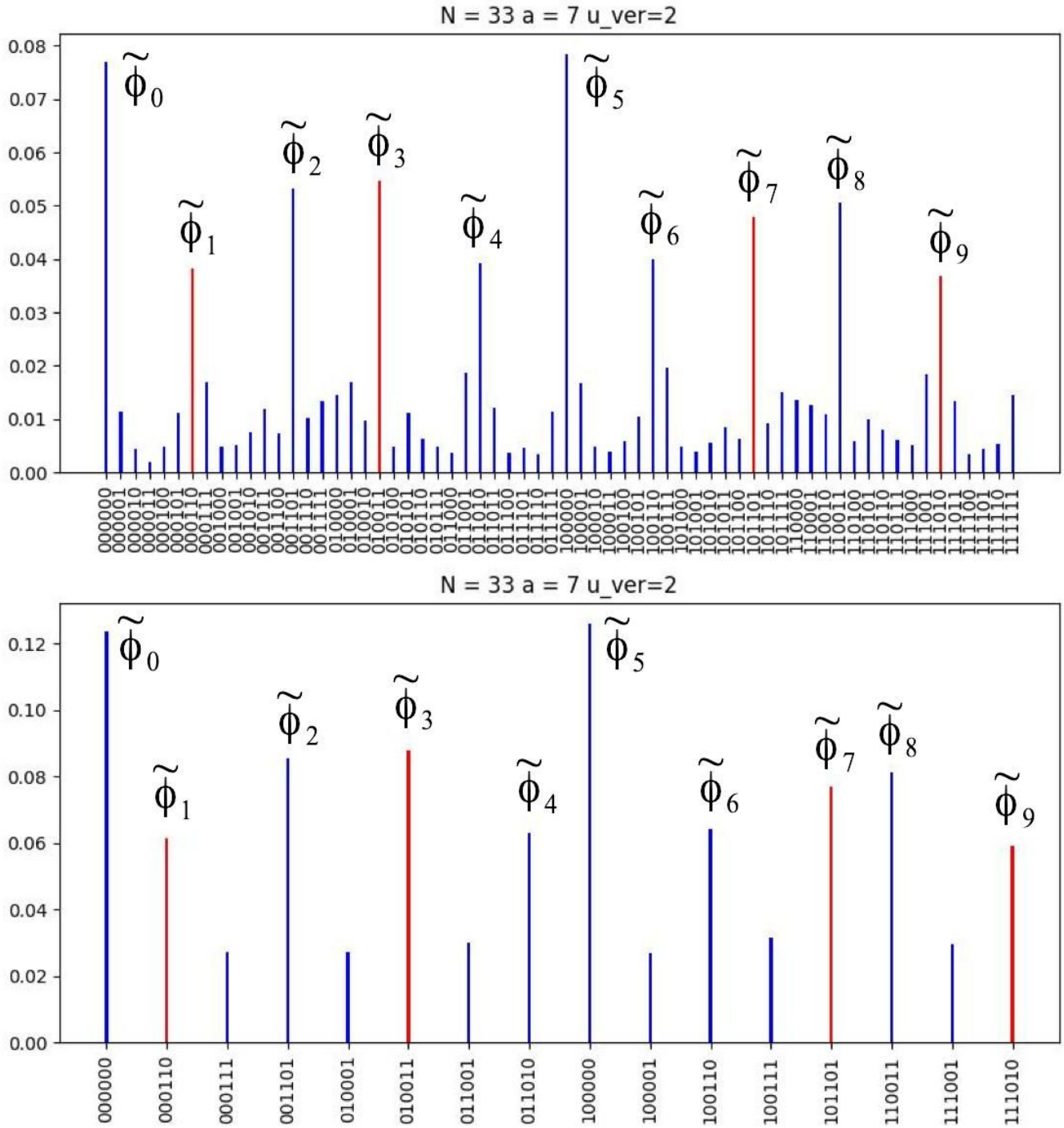
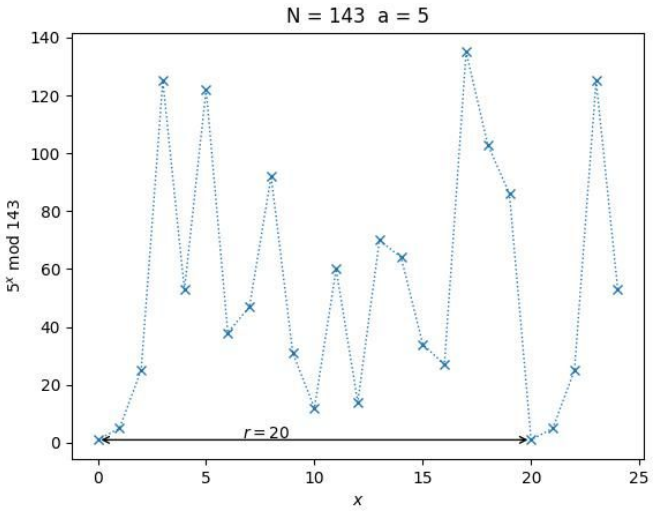


Figure 50: *$N = 33$, $a = 7$, $r = 10$, $m = 6$: Phase histogram for truncated ME operator version* `u_ver` $= 2$.

$N = 247 = 13 \times 19$ are given in red, and take the following values:

$$\begin{aligned}
\tilde{\phi}_1 &= [0.000001110]_2 = 0.027343750 \approx 1/36 \\
\tilde{\phi}_5 &= [0.001000111]_2 = 0.138671875 \approx 5/36 \\
\tilde{\phi}_{13} &= [0.010111001]_2 = 0.361328125 \approx 13/36 \\
\tilde{\phi}_{17} &= [0.011110010]_2 = 0.472656250 \approx 17/36 \\
\tilde{\phi}_{19} &= [0.100001110]_2 = 0.527343750 \approx 19/36 \\
\tilde{\phi}_{23} &= [0.101000111]_2 = 0.638671875 \approx 23/36 \\
\tilde{\phi}_{31} &= [0.110111001]_2 = 0.861328125 \approx 31/36 \\
\tilde{\phi}_{35} &= [0.111110010]_2 = 0.972656250 \approx 35/36 \,.
\end{aligned} \tag{289}$$



$U\|w\rangle =$	$\|5 \cdot w \pmod{143}\rangle$
$U\|1\rangle = \|5\rangle$	$U\|00000001\rangle = \|00000101\rangle$
$U\|5\rangle = \|25\rangle$	$U\|00000101\rangle = \|00011001\rangle$
$U\|25\rangle = \|125\rangle$	$U\|00011001\rangle = \|01111101\rangle$
$U\|125\rangle = \|53\rangle$	$U\|01111101\rangle = \|00110101\rangle$
$U\|53\rangle = \|122\rangle$	$U\|00110101\rangle = \|01111010\rangle$
$U\|122\rangle = \|38\rangle$	$U\|01111010\rangle = \|00100110\rangle$
$U\|38\rangle = \|47\rangle$	$U\|00100110\rangle = \|00101111\rangle$
$U\|47\rangle = \|92\rangle$	$U\|00101111\rangle = \|01011100\rangle$
$U\|92\rangle = \|31\rangle$	$U\|01011100\rangle = \|00011111\rangle$
$U\|31\rangle = \|12\rangle$	$U\|00011111\rangle = \|00001100\rangle$
$U\|12\rangle = \|60\rangle$	$U\|00001100\rangle = \|00111100\rangle$
$U\|60\rangle = \|14\rangle$	$U\|00111100\rangle = \|00001110\rangle$
$U\|14\rangle = \|70\rangle$	$U\|00001110\rangle = \|01000110\rangle$
$U\|70\rangle = \|64\rangle$	$U\|01000110\rangle = \|01000000\rangle$
$U\|64\rangle = \|34\rangle$	$U\|01000000\rangle = \|00100010\rangle$
$U\|34\rangle = \|27\rangle$	$U\|00100010\rangle = \|00011011\rangle$
$U\|27\rangle = \|135\rangle$	$U\|00011011\rangle = \|10000111\rangle$
$U\|135\rangle = \|103\rangle$	$U\|10000111\rangle = \|01100111\rangle$
$U\|103\rangle = \|86\rangle$	$U\|01100111\rangle = \|01010110\rangle$
$U\|86\rangle = \|1\rangle$	$U\|01010110\rangle = \|00000001\rangle$

Figure 51: *$N = 143$, $a = 5$, $r = 20$: The left panel gives the modular exponential function $f_{5,143}(x) = 5^x \pmod{143}$, and the right gives the action of the ME operator $U_{5,143}$ on the closed sequence* $[1, 5, 25, 125, 53, 122, 38, 47, 92, 31, 12, 60, 14, 70, 64, 34, 27, 135, 103, 86, 1]$.

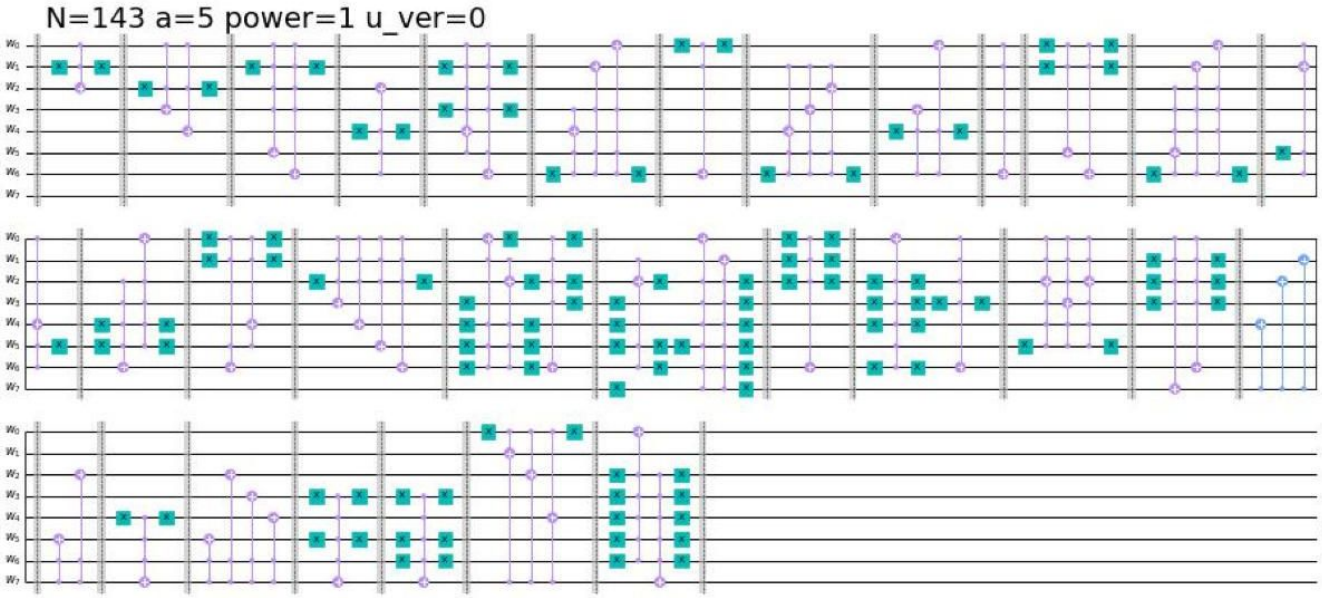


Figure 52: *$N = 143$, $a = 5$, $r = 20$: The modular exponentiation operator $U_{5,143}$.*

This is in agreement with the theoretical predictions of $\phi_s = s/36$ for $\gcd(s, 36) = 1$, except that the phases for $s = 7, 11, 25, 29$ are missing. As we have seen before, we can recover these phases by increasing the resolution of the control register. As noted above, for $m = 10$ we find that all expected phases are observed, as shown in Fig. 59.

Let us now address the composite operator issue for $U^p_{2,247}$ with $p > 1$. Returning to $m = 9$, so that $p = 1, 2, 4, \cdots, 256$, some of the closed cycles are given by

$$
\begin{aligned}
U_{2,247} \quad &: [1, 2, 4, 8, 16, 32, 64, 128, 9, 18, 36, 72, 144, 41, 82, 164, 81, 162, 77, 154, \\
&\quad 61, 122, 244, 241, 235, 223, 199, 151, 55, 110, 220, 193, 139, 31, 62, 124, 1] \qquad (290)\\
U^{128}_{2,247} \quad &: [1, 4, 16, 64, 9, 36, 144, 82, 81, 77, 61, 244, 235, 199, 55, 220, 139, 62, 1] \text{ and} \\
&\quad [2, 8, 32, 128, 18, 72, 41, 164, 162, 154, 122, 241, 223, 151, 110, 193, 31, 124, 2] \\
U^{2}_{2,247} \quad &: [1, 16, 9, 144, 81, 61, 235, 55, 139, 1] \text{ and } [2, 32, 18, 41, 162, 122, 223, 110, 31, 2] \\
&\quad [4, 64, 36, 82, 77, 244, 199, 220, 62, 4] \text{ and } [8, 128, 72, 164, 154, 241, 151, 193, 124, 8] \\
U^{4}_{2,247} \quad &: [1, 16, 9, 144, 81, 61, 235, 55, 139, 1] \text{ and } [2, 32, 18, 41, 162, 122, 223, 110, 31, 2] \\
&\quad [4, 64, 36, 82, 77, 244, 199, 220, 62, 4] \text{ and } [8, 128, 72, 164, 154, 241, 151, 193, 124, 8]
\end{aligned}
$$

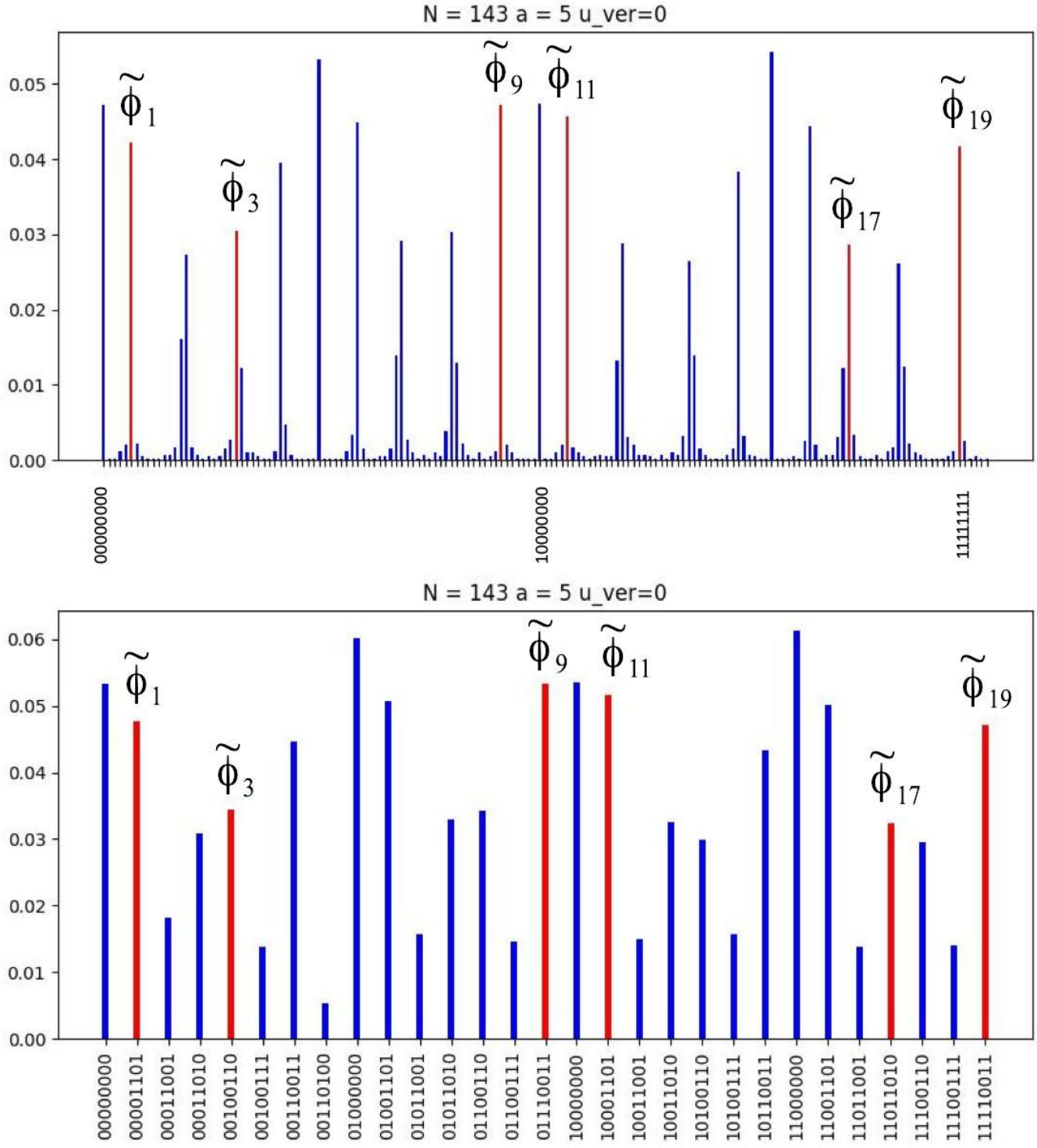


Figure 53: *$N = 143$, $a = 5$, $r = 20$, $m = 8$: Phase histogram for ME operator version* `u_ver` $= 0$ *The peaks in red correspond to the ME phases $\phi_s = s/20$ with $s \in \{0, 1, \cdots, 19\}$ and $gcd(s, 20) = 1$. Thus the eight phases $s = 1, 3, 7, 9, 11, 13, 17, 19$ provide the factors of 11 and 13. Note, however, that the peaks for $s = 7, 13$ are missing. This is because $m = 8$ does not provide sufficient resolution.*

$$
\begin{aligned}
U^{8}_{2,247} \quad &: \ [1, 9, 81, 235, 139, 16, 144, 61, 55, 1] \ \text{and} \ [2, 18, 162, 223, 31, 32, 41, 122, 110, 2] \\
&\quad [4, 36, 77, 199, 62, 64, 82, 244, 220, 4] \ \text{and} \ [8, 72, 154, 151, 124, 128, 164, 241, 193, 8] \\
U^{16}_{2,247} \quad &: \ [1, 81, 139, 144, 55, 9, 235, 16, 61, 1] \ \text{and} \ [2, 162, 31, 41, 110, 18, 223, 32, 122, 2] \\
&\quad [8, 154, 124, 164, 193, 72, 151, 128, 241, 8] \ \text{and} \ [4, 77, 62, 82, 220, 36, 199, 64, 244, 4] \\
U^{32}_{2,247} \quad &: \ [1, 139, 55, 235, 61, 81, 144, 9, 16, 1] \ \text{and} \ [2, 31, 110, 223, 122, 162, 41, 18, 32, 2] \\
&\quad [8, 124, 193, 151, 241, 154, 164, 72, 128, 8] \ \text{and} \ [4, 62, 220, 199, 244, 77, 82, 36, 64, 4]
\end{aligned}
$$

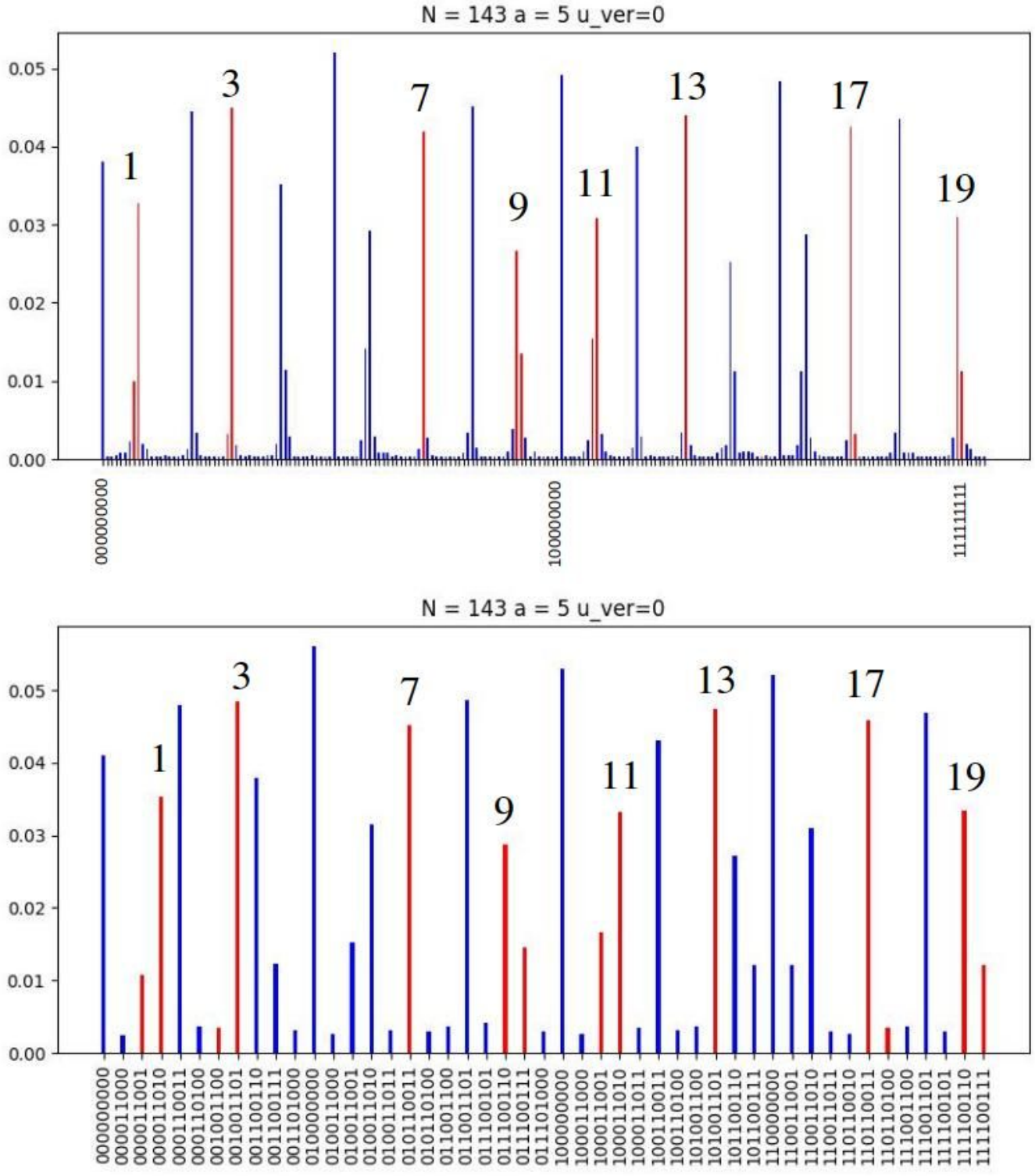


Figure 54: *$N = 143$, $a = 5$, $r = 20$, $m = 9$: Phase histogram for ME operator version* `u_ver` $= 0$. *Increasing the phase resolution to $m = 9$ provides all ten phases associated with factors.*

$U^{64}_{2,247}$: $[1, 55, 61, 144, 16, 139, 235, 81, 9, 1]$ and $[2, 110, 122, 41, 32, 31, 223, 162, 18, 2]$
$[8, 193, 241, 164, 128, 124, 151, 154, 72, 8]$ and $[4, 220, 224, 217, 79, 146, 126, 14, 29, 113, 40, 224, 4]$

$U^{128}_{2,247}$: $[1, 61, 16, 235, 9, 55, 144, 139, 81, 1]$ and $[2, 122, 32, 223, 18, 110, 41, 31, 162, 2]$
$[8, 241, 128, 151, 72, 193, 164, 124, 154, 8]$ and $[4, 244, 64, 199, 36, 220, 82, 62, 77, 4]$

$U^{256}_{2,247}$: $[1, 16, 9, 144, 81, 61, 235, 55, 139, 1]$ and $[2, 32, 18, 41, 162, 122, 223, 110, 31, 2]$
$[4, 64, 36, 82, 77, 244, 199, 220, 62, 4]$ and $[8, 128, 72, 164, 154, 241, 151, 193, 124, 8]$.

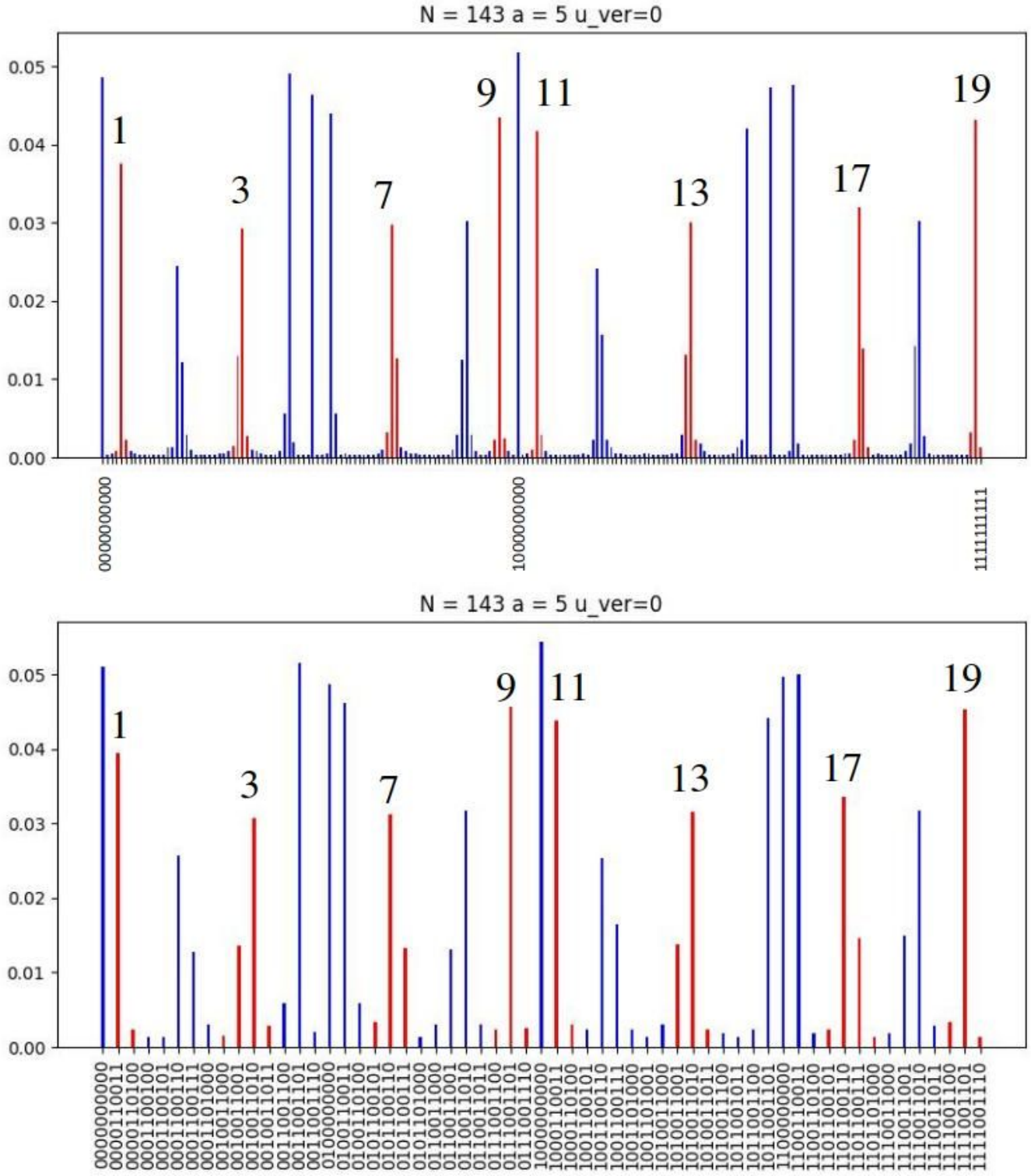


Figure 55: *$N = 143$, $a = 5$, $r = 20$, $m = 10$: Phase histogram for ME operator version* `u_ver` $= 0$.

For simplicity, we have not included all closed sub-cycles, and we will refer to this by version number `u_ver` $= 1$ (this can be regarded as a truncated version of the ME operators). The composite operators U^p are given in Figs. 64, 65, and 66 of Appendix A.2, and the corresponding phase histogram is presented in Fig. 60. We see that the results agree with the previous version, although there is more noise.

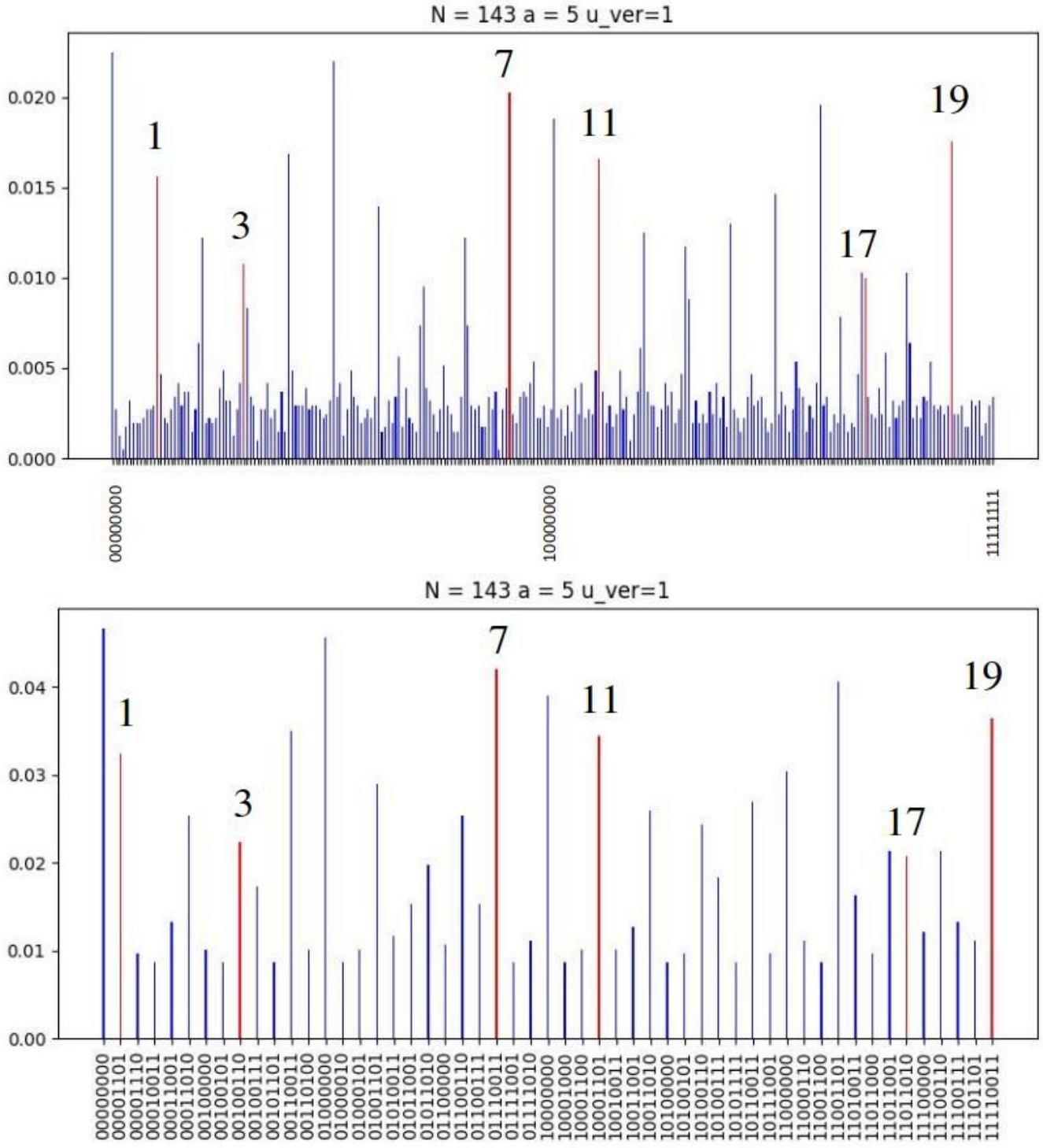
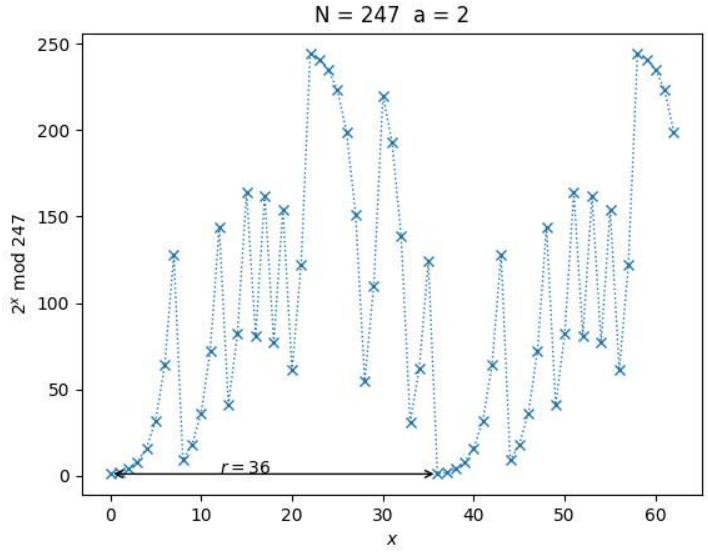


Figure 56: *$N = 143$, $a = 5$, $r = 20$, $m = 8$: Phase histogram for ME operator version* `u_ver` $= 1$. *Despite the noise relative to the previous version* `u_ver` $= 0$, *the signal is quite discernible.*

8. Conclusions and Outlook

It is an interesting mathematical fact that *factoring* is a notoriously difficult problem. That is to say, given an exponentially large integer, it is exceedingly hard to find the corresponding prime factors. In fact, all *known* factorization algorithms that run on a traditional classical (or digital) computer require an exponential time to factor such large numbers: A typical digital computer would take the age of the universe to factor a several thousand bit number used for encryption. Exponential computational costs are incurred because a classical computer must *sequentially* check almost every number less than the one being factored. Indeed, this is the basis upon which the



$U\|w\rangle =$	$\|2 \cdot w \pmod{247}\rangle$
$U\|1\rangle = \|2\rangle$	$U\|00000001\rangle = \|00000010\rangle$
$U\|2\rangle = \|4\rangle$	$U\|00000010\rangle = \|00000100\rangle$
$U\|4\rangle = \|8\rangle$	$U\|00000100\rangle = \|00001000\rangle$
$U\|8\rangle = \|16\rangle$	$U\|00001000\rangle = \|00010000\rangle$
$U\|16\rangle = \|32\rangle$	$U\|00010000\rangle = \|00100000\rangle$
$U\|32\rangle = \|64\rangle$	$U\|00100000\rangle = \|01000000\rangle$
$U\|64\rangle = \|128\rangle$	$U\|01000000\rangle = \|10000000\rangle$
$U\|128\rangle = \|9\rangle$	$U\|10000000\rangle = \|00001001\rangle$
$U\|9\rangle = \|18\rangle$	$U\|00001001\rangle = \|00010010\rangle$
$U\|18\rangle = \|36\rangle$	$U\|00010010\rangle = \|00100100\rangle$
$U\|36\rangle = \|72\rangle$	$U\|00100100\rangle = \|01001000\rangle$
$U\|72\rangle = \|144\rangle$	$U\|01001000\rangle = \|10010000\rangle$
$U\|144\rangle = \|41\rangle$	$U\|10010000\rangle = \|00101001\rangle$
$U\|41\rangle = \|82\rangle$	$U\|00101001\rangle = \|01010010\rangle$
$U\|82\rangle = \|164\rangle$	$U\|01010010\rangle = \|10100100\rangle$
$U\|164\rangle = \|81\rangle$	$U\|10100100\rangle = \|01010001\rangle$
$U\|81\rangle = \|162\rangle$	$U\|01010001\rangle = \|10100010\rangle$
$U\|162\rangle = \|77\rangle$	$U\|10100010\rangle = \|01001101\rangle$
$U\|77\rangle = \|154\rangle$	$U\|01001101\rangle = \|10011010\rangle$
$U\|154\rangle = \|61\rangle$	$U\|10011010\rangle = \|10011010\rangle$
$U\|61\rangle = \|122\rangle$	$U\|10011010\rangle = \|01111010\rangle$
$U\|122\rangle = \|244\rangle$	$U\|01111010\rangle = \|11110100\rangle$
$U\|244\rangle = \|241\rangle$	$U\|11110100\rangle = \|11110001\rangle$
$U\|241\rangle = \|235\rangle$	$U\|11110001\rangle = \|11101011\rangle$
$U\|235\rangle = \|223\rangle$	$U\|11101011\rangle = \|11011111\rangle$
$U\|223\rangle = \|199\rangle$	$U\|11011111\rangle = \|11000111\rangle$
$U\|199\rangle = \|151\rangle$	$U\|11000111\rangle = \|10010111\rangle$
$U\|151\rangle = \|55\rangle$	$U\|10010111\rangle = \|00110111\rangle$
$U\|55\rangle = \|110\rangle$	$U\|00110111\rangle = \|01101110\rangle$
$U\|110\rangle = \|220\rangle$	$U\|01101110\rangle = \|11011100\rangle$
$U\|220\rangle = \|193\rangle$	$U\|11011100\rangle = \|11000001\rangle$
$U\|193\rangle = \|139\rangle$	$U\|11000001\rangle = \|10001011\rangle$
$U\|139\rangle = \|31\rangle$	$U\|10001011\rangle = \|00011111\rangle$
$U\|31\rangle = \|62\rangle$	$U\|00011111\rangle = \|00111110\rangle$
$U\|62\rangle = \|124\rangle$	$U\|00111110\rangle = \|01111100\rangle$
$U\|124\rangle = \|1\rangle$	$U\|01111100\rangle = \|00000001\rangle$

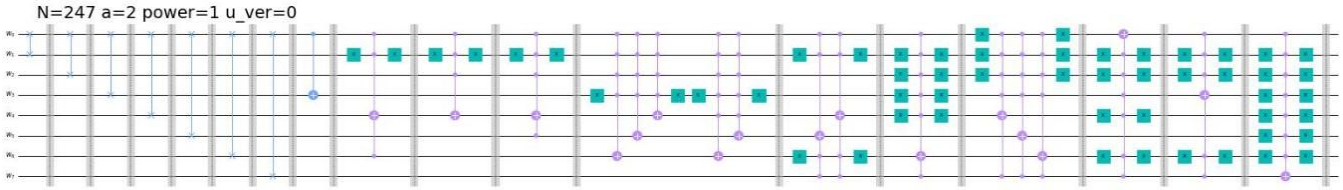


Figure 57: *$N = 247$, $a = 2$, $r = 36$: The top panel is the modular exponential function $f_{2,247}(x) = 2^x \pmod{247}$, while the bottom panel shows the modular exponentiation operator $U_{2,247}$.*

security of many public key cryptographic protocols rests. More specifically, the security of public key cyrto-systems relies on the principle that certain mathematical problems are intrinsically difficult to solve. For example, the RSA [2] system relies on the difficulty in factoring large numbers into their prime constituents, while Diffie–Hellman [3, 4] public key distribution relies on the difficulty in solving the discrete logarithm problem (a problem that is closely related to factoring). However, Shor's algorithm utilizes a quantum circuit that can factor exponentially large numbers in polynomial time (and a variant of the algorithm can also quickly solve the discrete logarithm problem) [1]. This is achieved by exploiting the massive parallelism inherent in quantum mechanics, so that all possibilities can be tested simultaneously rather than sequentially, thereby providing for a polynomial factorization process. Since Shor's algorithm can solve these very hard problems so quickly, the implications are quite sobering for the security of public key cryptography in particular, and digital security in general.

In this work we have presented a rigorous and pedagogical presentation of Shor's factoring algorithm. We have assumed no prior knowledge of the algorithm, except a familiarity with the circuit model of quantum computing, and we have walked the reader through the framework required to understand the algorithm, which is at the same time both elegant and complex. There are a number of moving parts to Shor's algorithm, and we have worked through each of them in turn, culminating in the requisite quantum factoring circuit.

The mathematical basis for Shor's algorithm has no connection with quantum mechanics, but rather rests upon a deep but quite simple result from number theory. Suppose we wish to factor an integer N, and we have a "guess" $a \in \{2, 3, \cdots, N-1\}$. We will usually refer to the guess a as the *base*. Let us further assume that the base a and the number N that we wish to factor contain no *common* factors, so that $\gcd(a, N) = 1$, otherwise $\gcd(a, N)$ is one

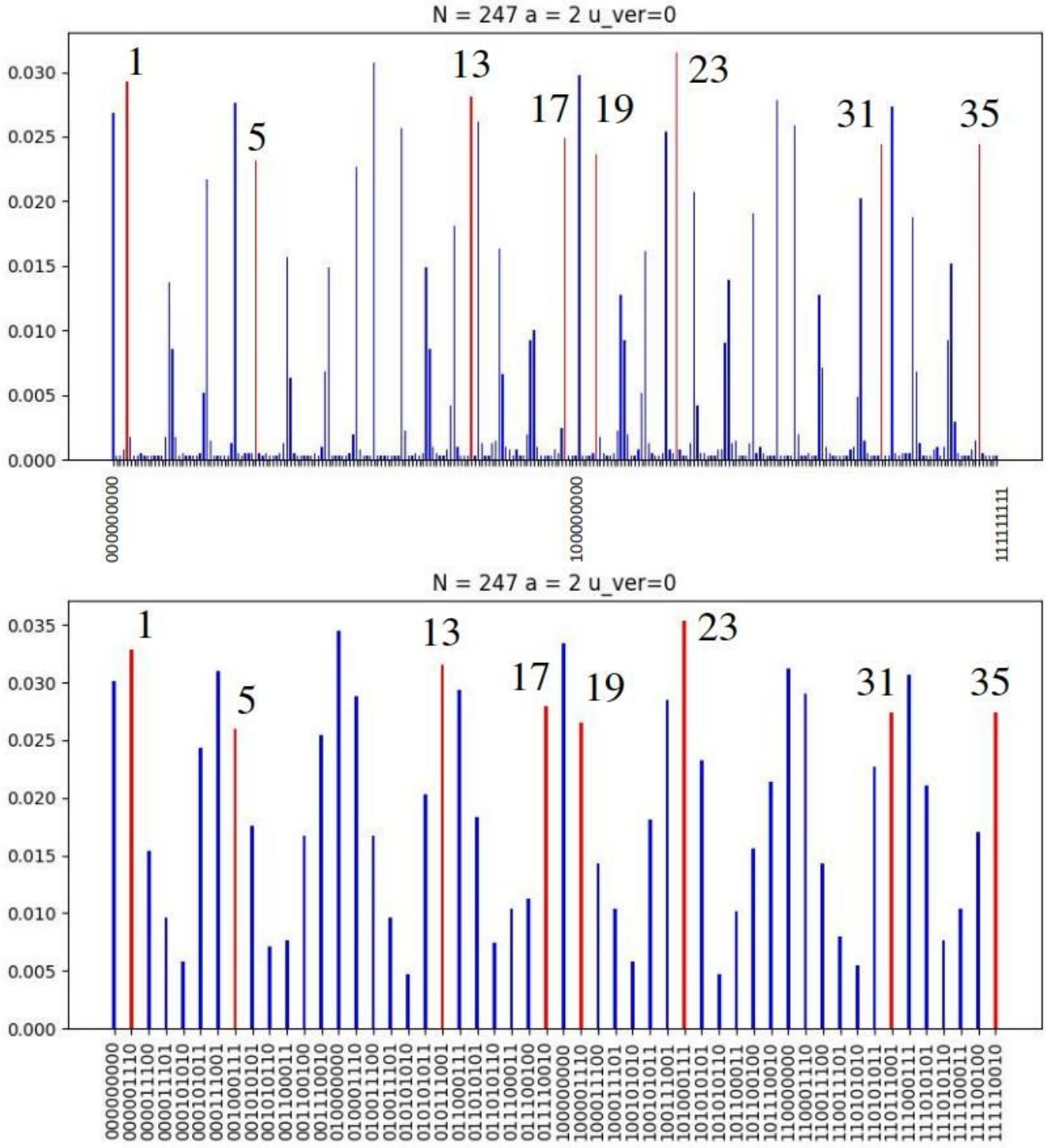


Figure 58: *$N = 247$, $a = 2$, $r = 36$, $m = 9$: Phase histogram for ME operator version* `u_ver` $= 0$.

of the factors of N that we seek (and the problem is solved). Suppose now that we can find a non-trivial *modular square root of unity*, so that $b^2 = 1 \pmod{N}$. The latter condition ensures that $b^2 - 1 = mN$ for some integer m. We can write this expression in the form $(b + 1)(b - 1) = mN$, and we immediately see that factors of N are given by $\gcd(b + 1, N)$ and $\gcd(b - 1, N)$. The greatest common divisor can be computed quickly on a classical computer in polynomial time. We can find b by looking at the modular exponential function $f_{a,N}(x) = a^x \pmod{N}$. This function is periodic with some period r, which means that $a^r = 1 \pmod{N}$. Therefore, $b = a^{r/2}$ is a square root of unity, and the factors of N are thus $\gcd(a^{r/2} \pm 1, N)$. We have now reduced the factoring problem to finding the period of the function $f_{a,N}(x)$! However, there are several caveats: the conditions (214)–(216) must all be met. First, equation (214) requires that the period r be even, so that $b = a^{r/2}$ is an integer. Second, (216) requires that r be a solution

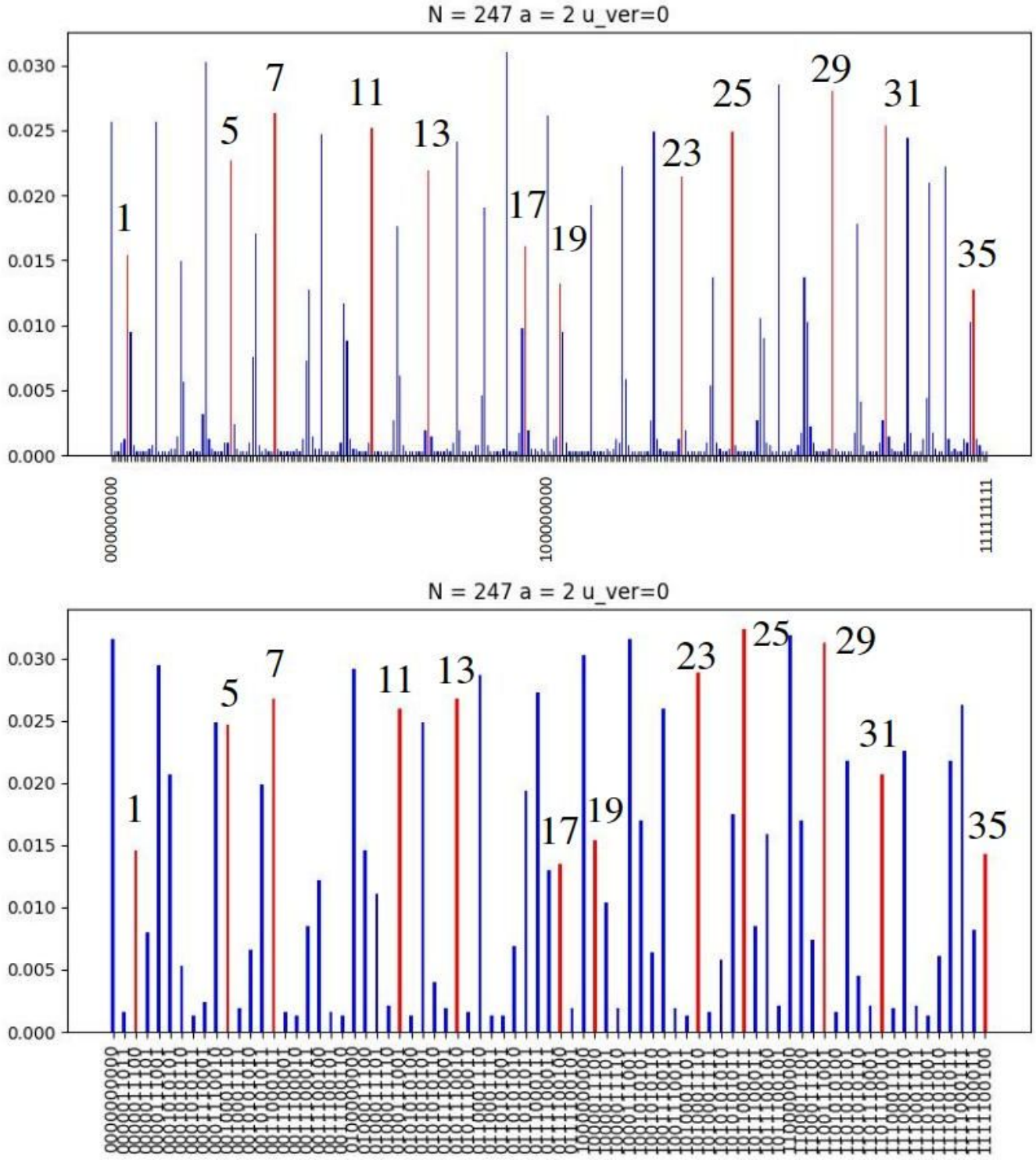


Figure 59: *$N = 247$, $a = 2$, $r = 36$ $m = 10$: Phase histogram for ME operator version* `u_ver` $= 0$.

to $a^r = 1 \pmod N$, so that $b = a^{r/2}$ is indeed a square root of unity. Third, while $b = a^{r/2}$ is a square root of unity, equation (215) prohibits it from being a *trivial* square root, in that $b \neq \pm 1 \pmod N$. In passing, we note that r can in fact be odd, provided that a is a *perfect square*, in which case $b = a^{r/2}$ is still an integer [11].

In contrast, the computational foundation for Shor's algorithm is a bit involved, and rests upon two fundamental quantum algorithms: the quantum Fourier transform (QFT) and quantum phase estimation (QPE). The QFT implements the discrete Fourier transform on a quantum computer, and the QPE algorithm finds the complex phases or the Eigenvalues of an arbitrary unitary linear operator. We spent Section 2 developing the theory of the QFT, and Section 3 covered the QPE, deriving these algorithms from scratch. Shor's algorithm is just a special case

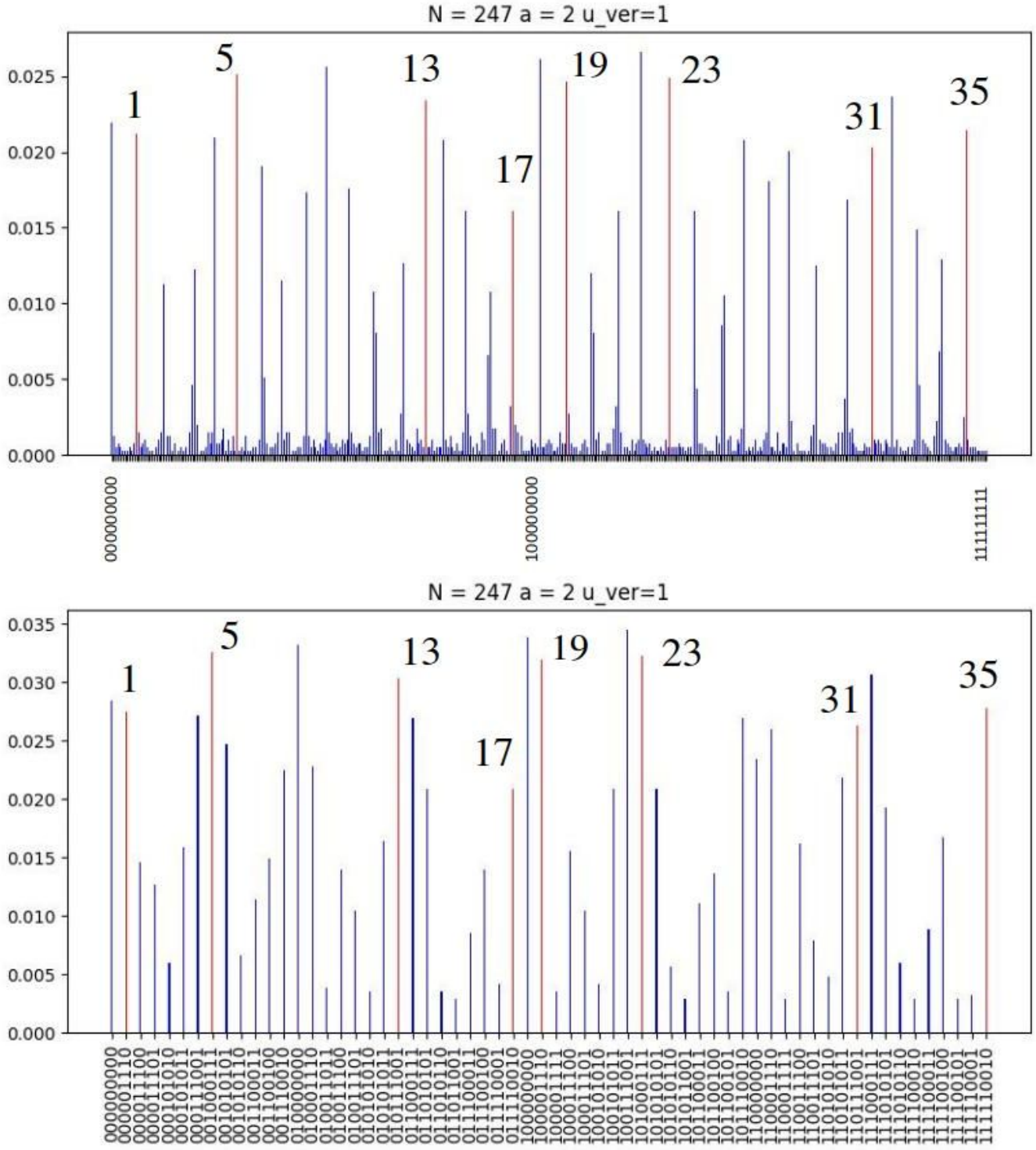


Figure 60: *$N = 247$, $a = 2$, $r = 36$, $m = 9$: Phase histogram for ME operators version* `u_ver` $= 1$.

of the QPE algorithm, with a well-chosen unitary *modular exponentiation* (ME) operator, denoted by U_{aN}. The ME operator is defined by its action on the computational basis by $U_{aN}|w\rangle = |a \cdot w \pmod N\rangle$, and it is related to the modular exponential function by $U^x_{aN}|1\rangle = |f_{aN}(x)\rangle$. The Eigenvalue problem for the ME operator takes the form $U_{aN}|u_s\rangle = e^{2\pi i \phi_s}|u_s\rangle$, where the phases are given by $\phi_s = s/r$ with $s \in \{0, 1, \cdots, r-1\}$. With the ME operators in hand, we combined the QFT and the QPE to construct Shor's factoring circuit in Section 5. The result is a hybrid approach requiring both classical processing and quantum computation for the QPE analysis. In the classical post-processing stage, the method of *continued fractions* allows one to extract the *exact* period r from the *approximately* measured phase $\tilde{\phi}$, thereby obtaining the period of the modular exponential function $f_{aN}(x)$. As we

have seen, this period is directly related to the factors of the number N, and the QPE cleverly extracts r to provide these sought after factors. Since continued fractions might not be familiar to the average reader, we gave a brief introduction to the subject in Section 4, proving a number of fundamental theorems. More specifically, Theorem 3 ensures that the phases $\phi_s = s/r$ that we are seeking will necessarily be one of the convergents of the continued fraction representation of $\tilde{\phi}$. We therefore simply construct all such convergents s_ℓ/r_ℓ, verifying that each value $r = r_\ell$ satisfies the necessary conditions (214)–(216). If so, then the smallest such value of r_ℓ is the exact period that we are seeking, thus permitting us to calculate the factors of N in polynomial time.

In Section 6, we presented a detailed example by factoring the number $N = 15$ using the Qiskit simulator, providing the necessary Python source code to reproduce the results. In particular, we looked more closely at how continued fractions are utilized to extract the exact phase $\phi_s = s/r$ from the approximately measured m-bit phase $\tilde{\phi}$. We also performed a theoretical analysis of the output phase histograms for $N = 15$ with bases $a = 4, 8$, calculating the expected histograms exactly for a general number of control qubits m. The Qiskit simulations agree precisely with the exact calculations.

After verifying the formalism by factoring $N = 15$, the smallest number accessible to Shor's algorithm, in Section 7 we moved on to factoring the more interesting composite numbers $N = 21, 33, 35, 143, 247$. The difficulty in factoring a number with Shor's algorithm does not lie in the size of the number itself, but in the magnitude of the period r of the modular exponential function $f(x)$ [7]. The numbers N have therefore been chosen, along with their respective bases a, to provide a wide range of periods, running from $r = 2$ to $r = 36$. We go on to develop a general procedure that will find the appropriate modular exponentiation operator U for any semi-prime $N = p \times q$, where p and q are prime. The principle behind this technique rests upon the fact that the modular exponential function $f(x)$ creates a sequence of states $|f(x)\rangle$ as we increment the argument x successively over its range of permissible values $x = 0, 1, 2, \cdots, r-1$. These states are the basis elements of an invariant r-dimensional subspace $\mathcal{U}_r$ of the exponentially large work space $\mathcal{W}_n$. To be more precise, note that the ME operator U first acts on the work state $|1\rangle$, and the next operation acts on the output of the first, and so on. Since $U^x|1\rangle = |f(x)\rangle$, this technique encodes the sequence of states generated by $f(x)$ into the ME operator U. One might think that we have gained nothing, since this method is equivalent to knowing the exact period r, and therefore Shor's algorithm would be unnecessary. However, the ME operators are quite forgiving, and they do not require knowing the full sequence of states. We can approximate the ME operator U by a truncated version using only a subset of the states. This is because the continued fraction method does not require knowing the exact phase ϕ_s, but only a sufficiently precise approximate phase $\tilde{\phi}$. This suggests that implementing Shor's algorithm might not be as difficult as first suspected.

In closing this work, we should briefly discuss some practical aspects of realizing Shor's algorithm. References [11, 14–16] have already succeeded in factoring the numbers $N = 15, 21, 35$ on a range of existing quantum computers. However, these authors did not implement complete versions of Shor's algorithm, but rather so called *compiled* versions that take advantage of the specific base a to minimize the qubit count. This is because current machines lack a sufficient number of qubits even for such small numbers. For more realistic cases, to factor a number N with $n = 1024$ bits, we would need $m = 2n + 1 = 2049$ control qubits, with the total number of qubits being $n + m = 3073$. For $n = 4096$ bits, these numbers increase to $m = 2n + 1 = 8193$ control qubits and $n + m = 12289$ total qubits. Breaking RSA therefore requires thousands to tens of thousands of high quality qubits. Modern quantum computers are currently quite far from this domain, although future machines will undoubtedly be able to handle these requirements. The gate count for the ME operators is also problematic. Reference [5] indicates that one would require $72n^3 \sim 8 \times 10^{10}$ gates for $n = 1024$ and $72n^3 \sim 5 \times 10^{12}$ gates for $n = 4096$ work qubits. The technique outlined in this work might well lower this gate count, but the requisite number of ME gates would still be quite large. Clearly, automation would be required for such a large number of gates. Even for the cases considered in this work, we employed a python script to write the Qiskit gates. Finally, implementing the QFT might seem to be the real challenge, as one requires astonishingly small phase angles for large numbers of qubits. Recall that the phase rotation angle is given by $\theta_k = 2\pi/2^k$, and for $k = 1000$, this gives an angle of order $2\pi/2^{1000} \sim 10^{-301}$! However, this problem has already been addressed in Refs. [17–19]. These authors show that for very small phase angles, one can simply ignore the corresponding phase rotation. In other words, we only need to consider phase angles larger than $\theta_{min} = 2\pi/2^{k_{max}}$ for some cutoff $k_{max} \sim 20$ [17]. This is not surprising, as we do not require the *exact* QFT, but only an approximate form that captures sufficient phase accuracy so that the method of continued fractions may be applied during the post-quantum processing. We see that there are indeed very large obstacles to overcome in breaking RSA with Shor's algorithm, but none of them seem insurmountable in the long run. For the immediate future, however, it seems that RSA will remain secure.

Acknowledgments

I would like to thank David Ostby, Mike Rodgers, and Max Singleton of the SavantX Quantum Division. This work was funded by the SavantX Research Center.

Supplementary Materials

The online version of this article contains Python scripts together with README file for their use.

A. Modular Exponentiation Operators

A.1. Composite ME operators for N = 143, a = 5, r = 20, m = 8, u_ver=1

The composite operators $U^{p}_{5,143}$ from (288) for $p = 2, 4, \cdots, 128$.

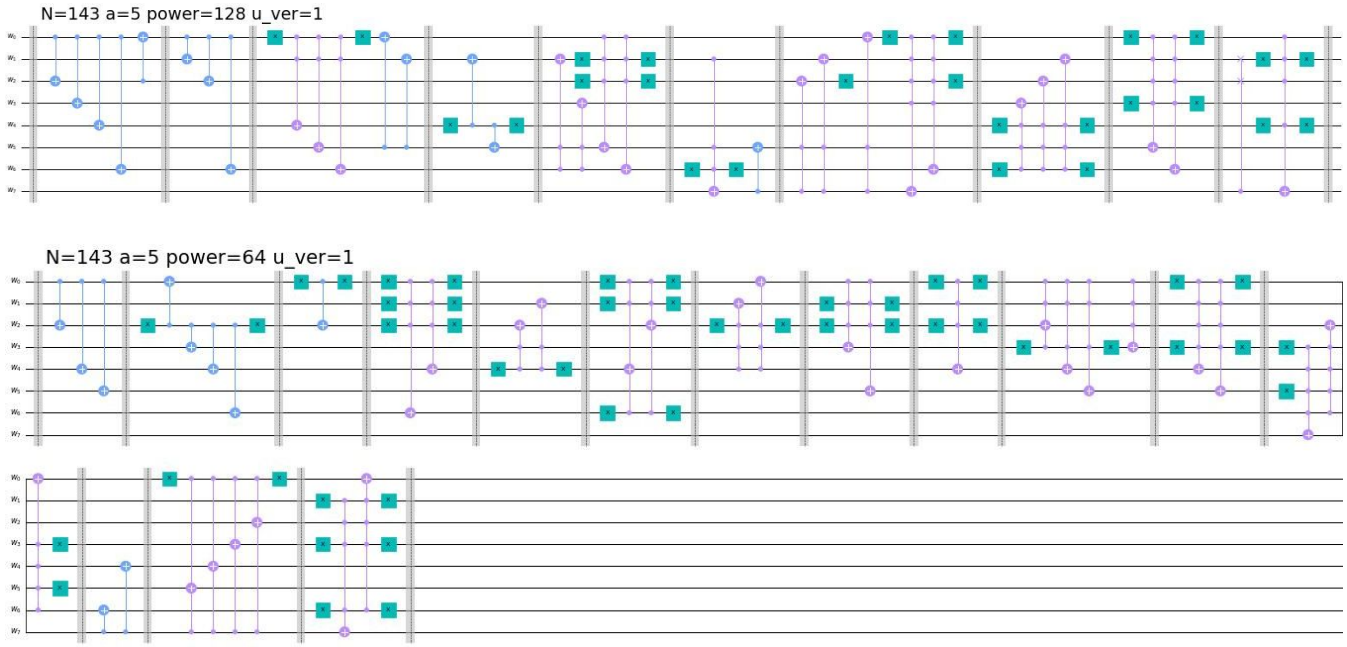


Figure 61: $N = 143$, $a = 5$, $r = 20$, $\texttt{u\_ver} = 1$: $U^{64}_{5,143}, U^{128}_{5,143}$.

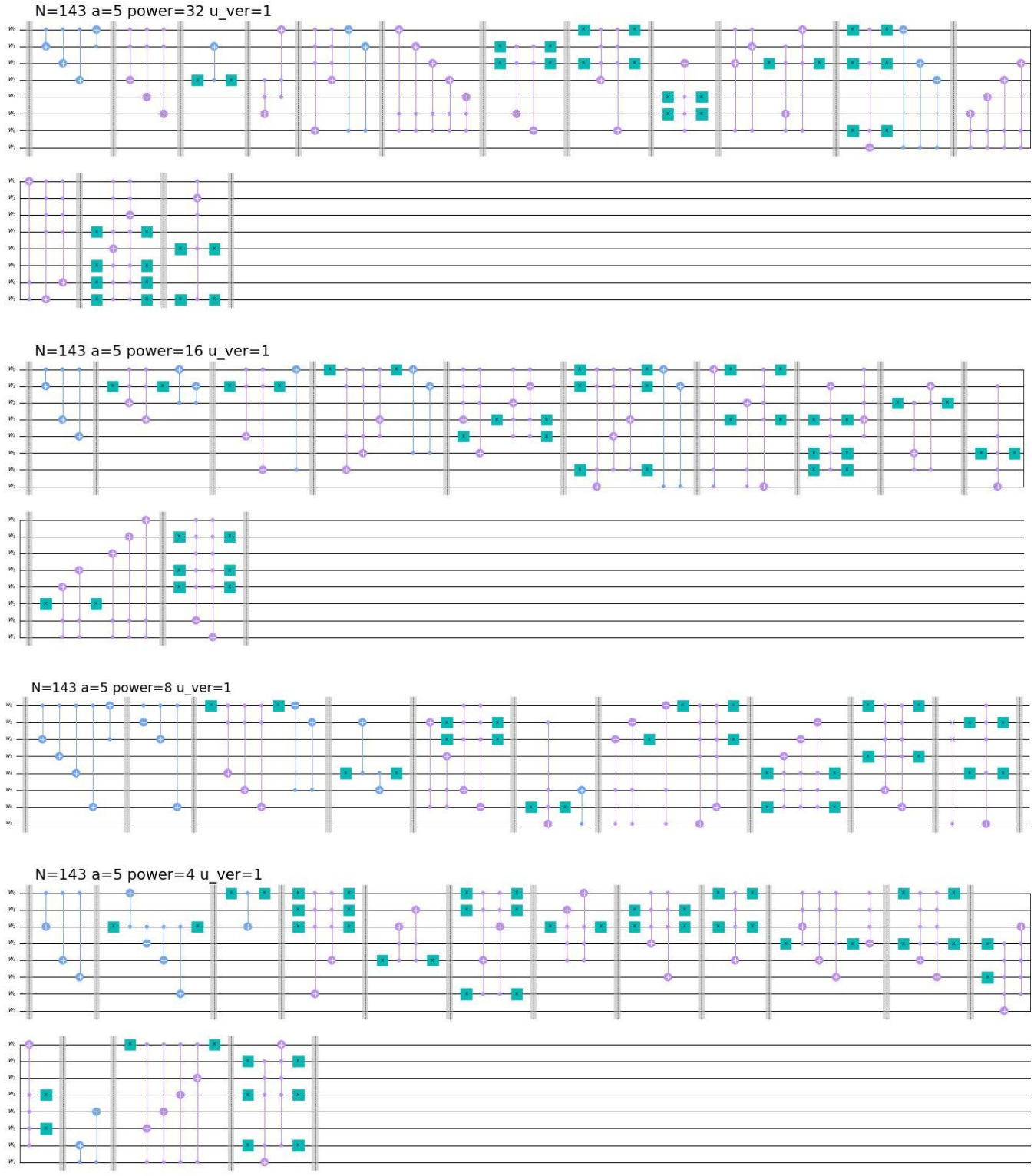


Figure 62: $N = 143$, $a = 5$, $r = 20$, $\texttt{u\_ver} = 1$: $U^4_{5,143}, U^8_{5,143}, U^{16}_{5,143}, U^{32}_{5,143}$.

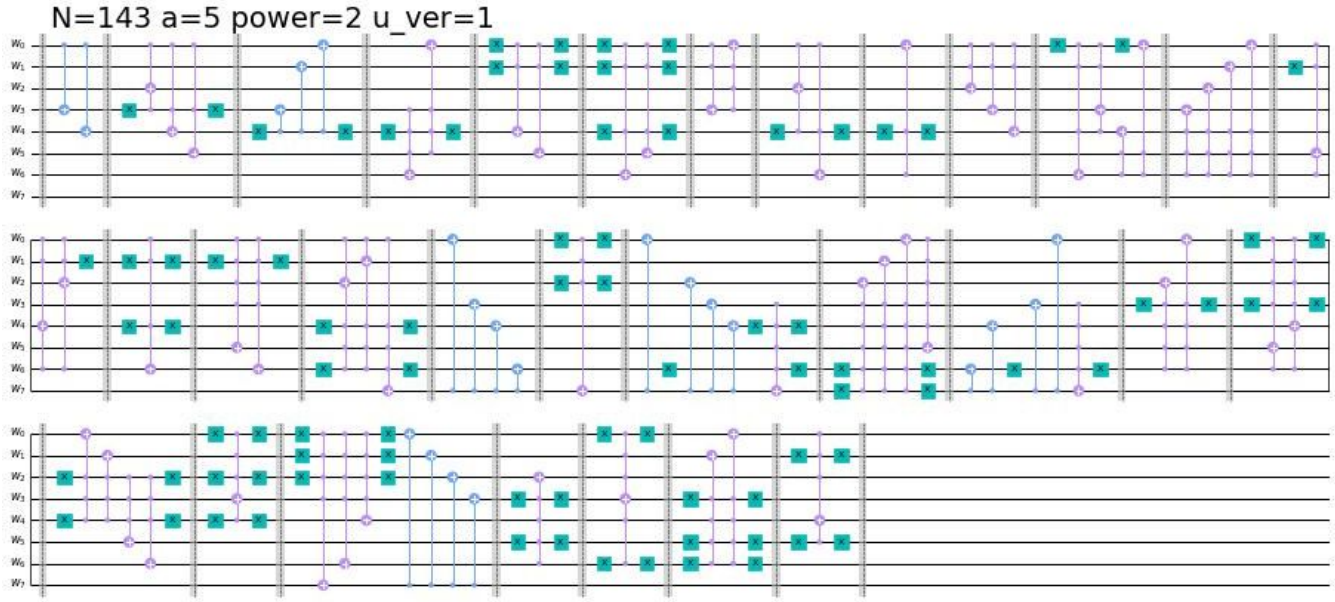


Figure 63: $N = 143$, $a = 5$, $r = 20$, u_ver $= 1$*:* $U^2_{5,143}$.

A.2. Composite ME operators for N = 247, a = 2, r = 36, m = 9, u_ver=1

The composite operators $U^{p}_{2,247}$ from (290) for $p = 2, 4, \cdots, 256$.

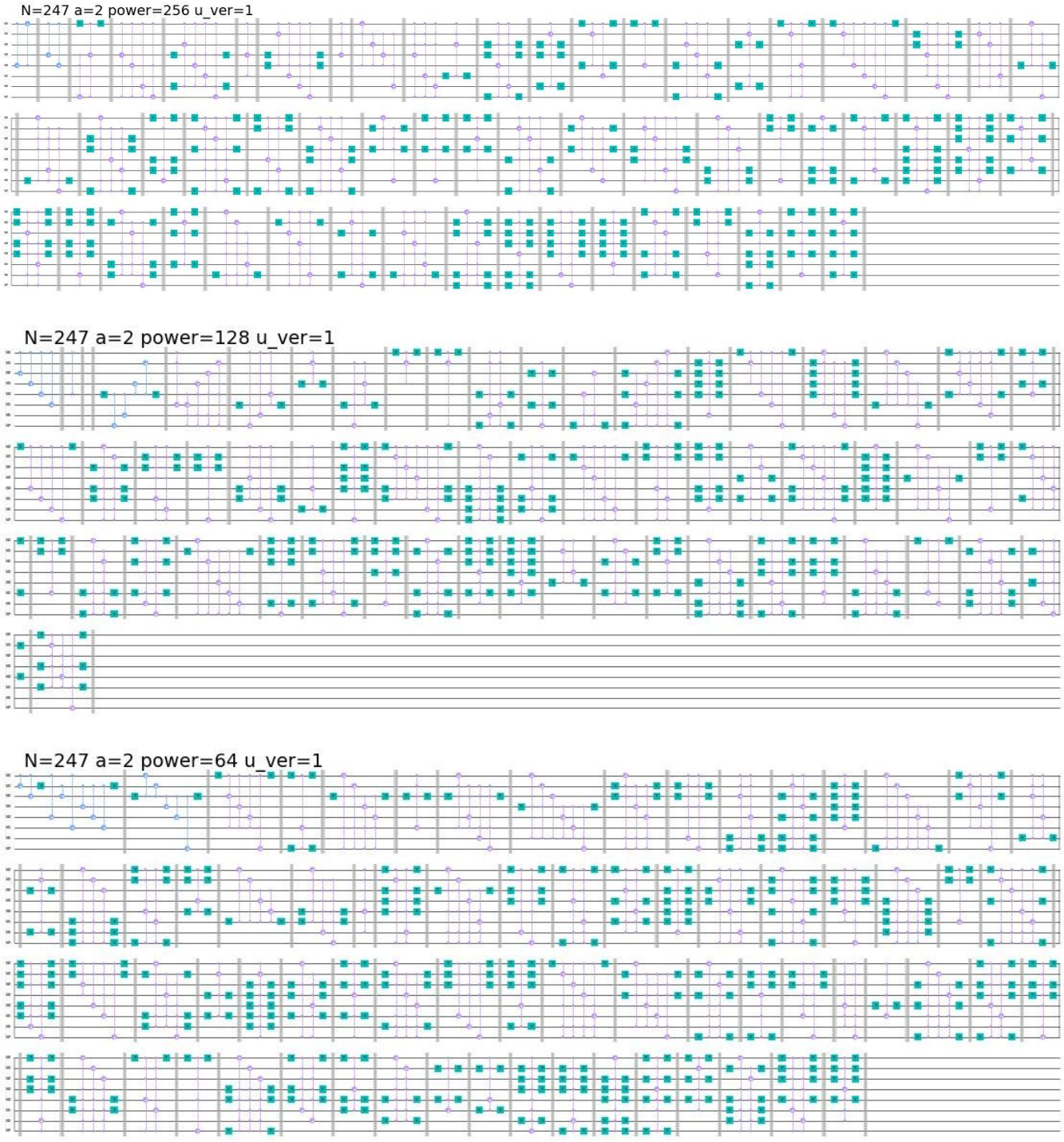


Figure 64: $N = 247$, $a = 2$, $r = 36$, u_ver $= 1$: $U^{64}_{2,247}$, $U^{128}_{2,247}$, $U^{256}_{2,247}$

Figure 65: $N = 247$, $a = 2$, $r = 36$, $\texttt{u\_ver} = 1$: $U^{8}_{2,247}$, $U^{16}_{2,247}$, $U^{32}_{2,247}$.

Figure 66: $N = 247$, $a = 2$, $r = 36$, $\texttt{u\_ver} = 1$*:* $U^2_{2,247}$, $U^4_{2,247}$.

References

[1] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing* 1997; **26**(5):1484–1509. arXiv:quant-ph/9508027. doi:10.1137/S0097539795293172.

[2] R. L. Rivest, A. Shamir, L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM* 1978; **21**(2):120–126. doi:10.1145/359340.359342.

[3] W. Diffie, M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory* 1976; **22**(6):644–654. doi:10.1109/TIT.1976.1055638.

[4] R. C. Merkle. Secure communications over insecure channels. *Communications of the ACM* 1978; **21**(4):294–299. doi:10.1145/359460.359473.

[5] D. Beckman, A. N. Chari, S. Devabhaktuni, J. Preskill. Efficient networks for quantum factoring. *Physical Review A* 1996; **54**(2):1034–1063. arXiv:quant-ph/9602016. doi:10.1103/PhysRevA.54.1034.

[6] D. R. Simon. On the power of quantum computation. *SIAM Journal on Computing* 1997; **26**(5):1474–1483. doi:10.1137/S0097539796298637.

[7] J. A. Smolin, G. Smith, A. Vargo. Oversimplifying quantum factoring. *Nature* 2013; **499**(7457):163–165. arXiv:1301.7007. doi:10.1038/nature12290.

[8] M. A. Nielsen, I. L. Chuang. Quantum Computation and Quantum Information. Cambridge University Press, Cambridge, 2010. doi:10.1017/cbo9780511976667.

[9] E. Desurvire. Classical and Quantum Information Theory: An Introduction for the Telecom Scientist. Cambridge University Press, Cambridge, 2009. doi:10.1017/cbo9780511803758.

[10] A. W. Cross, L. S. Bishop, J. A. Smolin, J. M. Gambetta. Open quantum assembly language 2017; arXiv:1707.03429.

[11] U. Skosana, M. Tame. Demonstration of Shor's factoring algorithm for $n = 21$ on IBM quantum processors. *Scientific Reports* 2021; **11**(1):16599. doi:10.1038/s41598-021-95973-w.

[12] Q. D. Team. Shor's algorithm. in: Qiskit Textbook. 2023. https://learn.qiskit.org/course/ch-algorithms/shors-algorithm.

[13] M. Guštin. Continued Fractions Python Module. 2019. https://github.com/TheMatjaz/contfrac.

[14] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, I. L. Chuang. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature* 2001; **414**(6866):883–887. doi:10.1038/414883a.

[15] E. Martín-López, A. Laing, T. Lawson, R. Alvarez, X.-Q. Zhou, J. L. O'Brien. Experimental realization of Shor's quantum factoring algorithm using qubit recycling. *Nature Photonics* 2012; **6**(11):773–776. doi:10.1038/nphoton.2012.259.

[16] M. Amico, Z. H. Saleem, M. Kumph. Experimental study of Shor's factoring algorithm using the IBM Q Experience. *Physical Review A* 2019; **100**(1):012305. doi:10.1103/PhysRevA.100.012305.

[17] D. Coppersmith. An approximate Fourier transform useful in quantum factoring. IBM Research Division, 1994. arXiv:quant-ph/0201067.

[18] A. G. Fowler, L. C. L. Hollenberg. Scalability of Shor's algorithm with a limited set of rotation gates. *Physical Review A* 2004; **70**(3):032329. doi:10.1103/PhysRevA.70.032329.

[19] Y. S. Nam, R. Blümel. Scaling laws for Shor's algorithm with a banded quantum Fourier transform. *Physical Review A* 2013; **87**(3):032333. doi:10.1103/PhysRevA.87.032333.