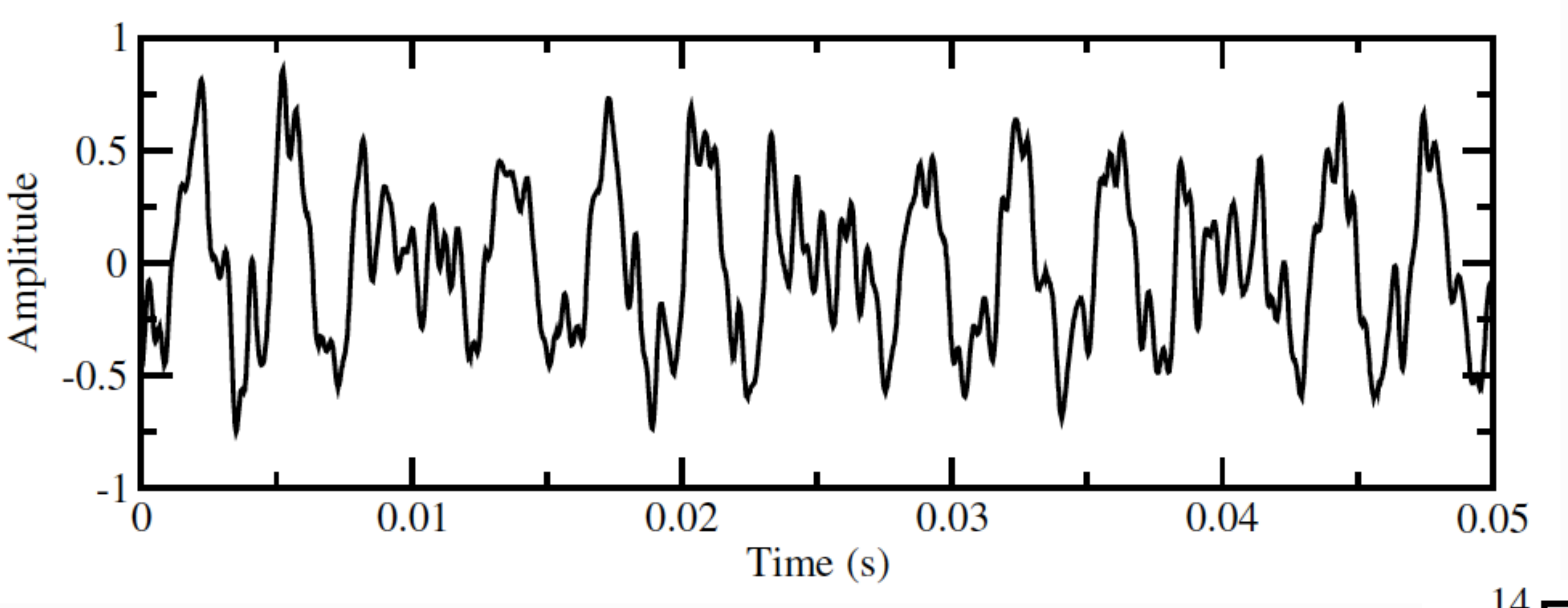
Quantum Fourier transform: motivation

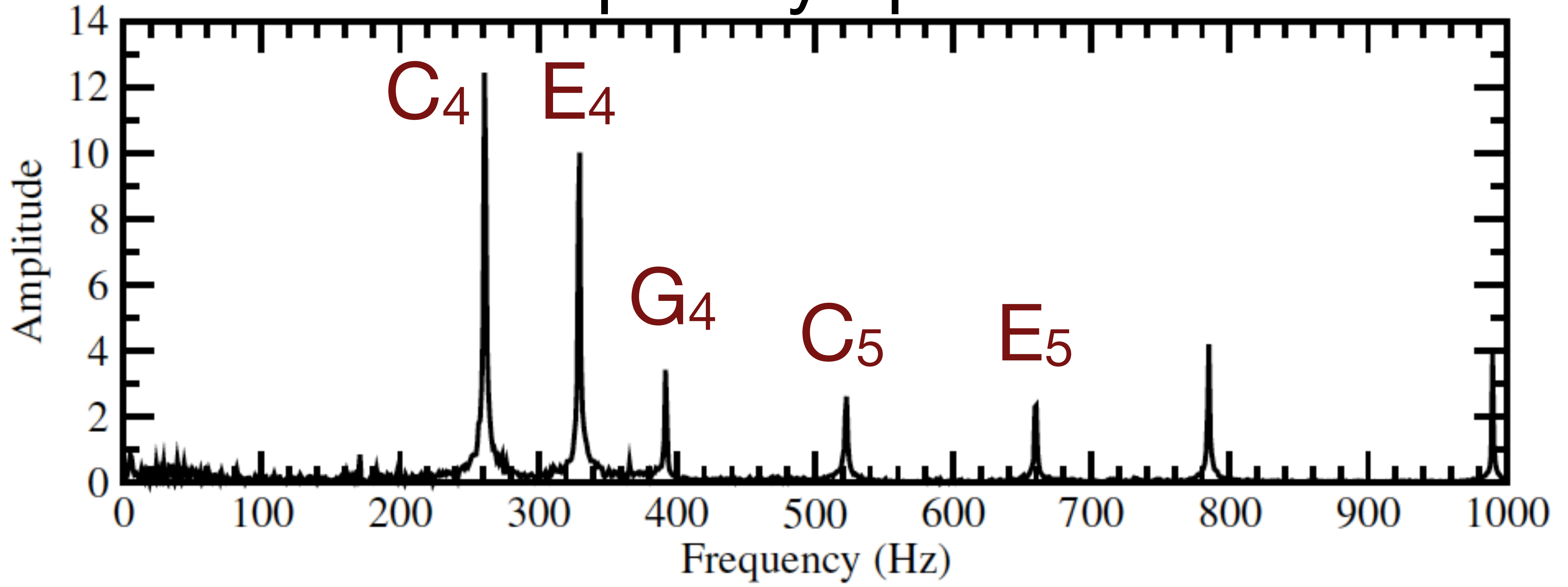
Classical solution

Discrete Fourier Transform (DFT) is used for data processing analysis.



Waveform of a piano playing a C major chord.

Frequency spectrum



With DFT it is possible to discover which frequencies are composing the chord.

C_4 (C middle) corresponds to 262 Hz

[1] T. G. Wong, Introduction to Classical and Quantum Computing (2022). https://www.thomaswong.net/

Discrete Fourier Transform

The discrete Fourier transform is

$$\phi_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} a_j \omega^{jk}$$

$k \in \{0,1,2,\ldots,N-1\}$

$$\omega = e^{i2\pi/N}$$

More explicitly

$$\phi_0 = \frac{1}{\sqrt{N}}\left(a_0 + a_1 + a_2 + \cdots + a_{N-1}\right),$$

$$\phi_1 = \frac{1}{\sqrt{N}}\left(a_0 + a_1\omega + a_2\omega^2 + \cdots + a_{N-1}\omega^{N-1}\right),$$

$$\phi_2 = \frac{1}{\sqrt{N}}\left(a_0 + a_1\omega^2 + a_2\omega^4 + \cdots + a_{N-1}\omega^{2(N-1)}\right),$$

$$\vdots$$

$$\phi_{N-1} = \frac{1}{\sqrt{N}}\left(a_0 + a_1\omega^{N-1} + a_2\omega^{2(N-1)} + \cdots + a_{N-1}\omega^{(N-1)^2}\right)$$

DFT matrix

$$\begin{pmatrix} \phi_0 \\ \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{N-1} \end{pmatrix} = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & \omega & \omega^2 & \ldots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \ldots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \ldots & \omega^{(N-1)^2} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{N-1} \end{pmatrix}$$

It is necessary to compute $O(N^2)$ terms.

Fast Fourier transform implements in $O(N \log N)$ steps.

Quantum solution

Using the quantum formalism, the state corresponding to the sound amplitudes is

$$|\phi\rangle = \begin{pmatrix} \phi_0 \\ \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{N-1} \end{pmatrix} = \phi_0|0\rangle + \cdots + \phi_{N-1}|N-1\rangle$$

While the transformed state is

$$|\psi\rangle = \sum_{j=0}^{N-1} a_j|j\rangle \longrightarrow |\phi\rangle = \sum_{k=0}^{N-1} \phi_k|k\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} a_j e^{2\pi i jk/N}|k\rangle$$

The matrix used to implement the DFT can be used here. *It is unitary!*

Quantum Fourier tranform

So, the quantum Fourier transform (QFT) gate is

$$\mathrm{QFT} = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \dots & \omega^{(N-1)^2} \end{pmatrix}$$

$$\omega = e^{i2\pi/N}$$
$$N = 2^n$$

$n = \#$ of qubits

Its action on the basis states is

$$|j\rangle \longrightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i jk/N} |k\rangle$$

Exercise: Show by matrix multiplication that the QFT gate is unitary.

Quantum circuit of QFT

Lets decompose the QFT into single-qubit and two-qubit quantum gates. However, first we need rearrange the argument of the exponentials.

Representing j as a n-binary number

$$\begin{aligned} j &= j_{n-1} j_{n-2} \cdots j_1 j_0 \\ &= j_{n-1} 2^{n-1} + j_{n-2} 2^{n-2} + \cdots + j_1 2 + j_0 \end{aligned}$$

Then, j/N can be represented using a *binary point* as

$$\begin{aligned} \frac{j}{N} &= \frac{j_{n-1} 2^{n-1} + j_{n-2} 2^{n-2} + \cdots + j_1 2 + j_0}{2^n} \\ &= \frac{j_{n-1}}{2} + \frac{j_{n-2}}{2^2} + \cdots + \frac{j_1}{2^{n-1}} + \frac{j_0}{2^n} \\ &= 0.j_{n-1} j_{n-2} \ldots j_1 j_0. \end{aligned}$$

Expressing k as an n-bit binary number

$$\begin{aligned} k &= k_{n-1}k_{n-2}\ldots k_1k_0 \\ &= k_{n-1}2^{n-1} + k_{n-2}2^{n-2} + \cdots + k_1 2 + k_0 \end{aligned}$$

we obtain

$$\begin{aligned} e^{2\pi i jk/N} &= e^{2\pi i (j/N)k} \\ &= e^{2\pi i (0.j_{n-1}j_{n-2}\cdots j_1 j_0)(k_{n-1}2^{n-1}+k_{n-2}2^{n-2}+\cdots+k_1 2+k_0)} \\ &= e^{2\pi i (0.j_{n-1}j_{n-2}\cdots j_1 j_0)k_{n-1}2^{n-1}} e^{2\pi i (0.j_{n-1}j_{n-2}\cdots j_1 j_0)k_{n-2}2^{n-2}} \ldots \\ &\quad \times e^{2\pi i (0.j_{n-1}j_{n-2}\cdots j_1 j_0)k_1 2} e^{2\pi i (0.j_{n-1}j_{n-2}\cdots j_1 j_0)k_0} \\ &= e^{2\pi i (j_{n-1}j_{n-2}\cdots j_1 . j_0)k_{n-1}} e^{2\pi i (j_{n-1}j_{n-2}\cdots j_2 . j_1 j_0)k_{n-2}} \ldots \\ &\quad \times e^{2\pi i (j_{n-1} . j_{n-2}\cdots j_1 j_0)k_1} e^{2\pi i (0.j_{n-1}j_{n-2}\cdots j_1 j_0)k_0} . \end{aligned}$$

We can drop all the bits to the left of the binary point. Example:

$$
\begin{aligned}
e^{2\pi i (j_{n-1} j_{n-2} \cdots j_1 . j_0) k_{n-1}} &= e^{2\pi i (j_{n-1} 2^{n-2} + j_{n-2} 2^{n-3} \cdots j_1 + j_0/2) k_{n-1}} \\
&= \underbrace{e^{2\pi i j_{n-1} 2^{n-2} k_{n-1}}}_{1} \underbrace{e^{2\pi i j_{n-2} 2^{n-3} k_{n-1}}}_{1} \cdots \underbrace{e^{2\pi i j_1 k_{n-1}}}_{1} e^{2\pi i j_0/2 k_{n-1}} \\
&= e^{2\pi i 0.j_0 k_{n-1}} .
\end{aligned}
$$

Then, we get

$$
\begin{aligned}
e^{2\pi i jk/N} = & \; e^{2\pi i (0.j_0) k_{n-1}} e^{2\pi i (0.j_1 j_0) k_{n-2}} \cdots \\
& \times e^{2\pi i (0.j_{n-2} \cdots j_1 j_0) k_1} e^{2\pi i (0.j_{n-1} j_{n-2} \cdots j_1 j_0) k_0} .
\end{aligned}
$$

The application of the QFT on a basis state can be written as

$$
\begin{aligned}
|j\rangle \rightarrow & \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i jk/N} |k\rangle \\
&= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i (0.j_0) k_{n-1}} e^{2\pi i (0.j_1 j_0) k_{n-2}} \dots \\
& \qquad \times e^{2\pi i (0.j_{n-2} \dots j_1 j_0) k_1} e^{2\pi i (0.j_{n-1} j_{n-2} \dots j_1 j_0) k_0} |k\rangle .
\end{aligned}
$$

The sum over the binary numbers k is equivalent to the sum over each bit,

$$
\begin{aligned}
& \frac{1}{\sqrt{N}} \sum_{k_{n-1}=0}^{1} \dots \sum_{k_0=0}^{1} e^{2\pi i (0.j_0) k_{n-1}} e^{2\pi i (0.j_1 j_0) k_{n-2}} \dots \\
& \qquad \times e^{2\pi i (0.j_{n-2} \dots j_1 j_0) k_1} e^{2\pi i (0.j_{n-1} j_{n-2} \dots j_1 j_0) k_0} |k_{n-1} \dots k_0\rangle
\end{aligned}
$$

As $|k_{n-1}\dots k_0\rangle = |k_{n-1}\rangle \otimes \dots \otimes |k_0\rangle$, moving the summations, we get

$$\frac{1}{\sqrt{N}} \sum_{k_{n-1}=0}^{1} e^{2\pi i (0.j_0)k_{n-1}} |k_{n-1}\rangle \sum_{k_{n-2}=0}^{1} e^{2\pi i (0.j_1 j_0)k_{n-2}} |k_{n-2}\rangle \dots$$
$$\times \sum_{k_1=0}^{1} e^{2\pi i (0.j_{n-2}\dots j_1 j_0)k_1} |k_1\rangle \sum_{k_0=0}^{1} e^{2\pi i (0.j_{n-1} j_{n-2}\dots j_1 j_0)k_0} |k_0\rangle$$

or

$|j_{n-1}\rangle$ $|j_{n-2}\rangle$

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i (0.j_0)}|1\rangle\right) \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i (0.j_1 j_0)}|1\rangle\right)\dots$$
$$\times \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i (0.j_{n-2}\dots j_1 j_0)}|1\rangle\right) \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i (0.j_{n-1} j_{n-2}\dots j_1 j_0)}|1\rangle\right)$$

$|j_1\rangle$ $|j_0\rangle$

Now, lets create the quantum circuit using Hadamard and controlled-rotations. Starting with state $|j_{n-1}\rangle$

$$H|j_{n-1}\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^{j_{n-1}}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + (e^{i\pi})^{j_{n-1}}|1\rangle\right)$$
$$= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i j_{n-1}/2}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(0.j_{n-1})}|1\rangle\right)$$

Consider a single-qubit gate that rotates about the z-axis of the Bloch sphere by $2\pi/2^r$ radians

$$R_r = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^r} \end{pmatrix} \Rightarrow \begin{aligned} R_r|0\rangle &= |0\rangle, \\ R_r|1\rangle &= e^{2\pi i/2^r}|1\rangle \end{aligned}$$

Applying R_2 to qubit n-1 controlled by qubit n-2,

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(0.j_{n-1})}|1\rangle\right) \to \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(0.j_{n-1})}(e^{2\pi i/2^2})^{j_{n-2}}|1\rangle\right)$$
$$= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(0.j_{n-1})}e^{2\pi i(0.0j_{n-2})}|1\rangle\right)$$
$$= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(0.j_{n-1}j_{n-2})}|1\rangle\right).$$

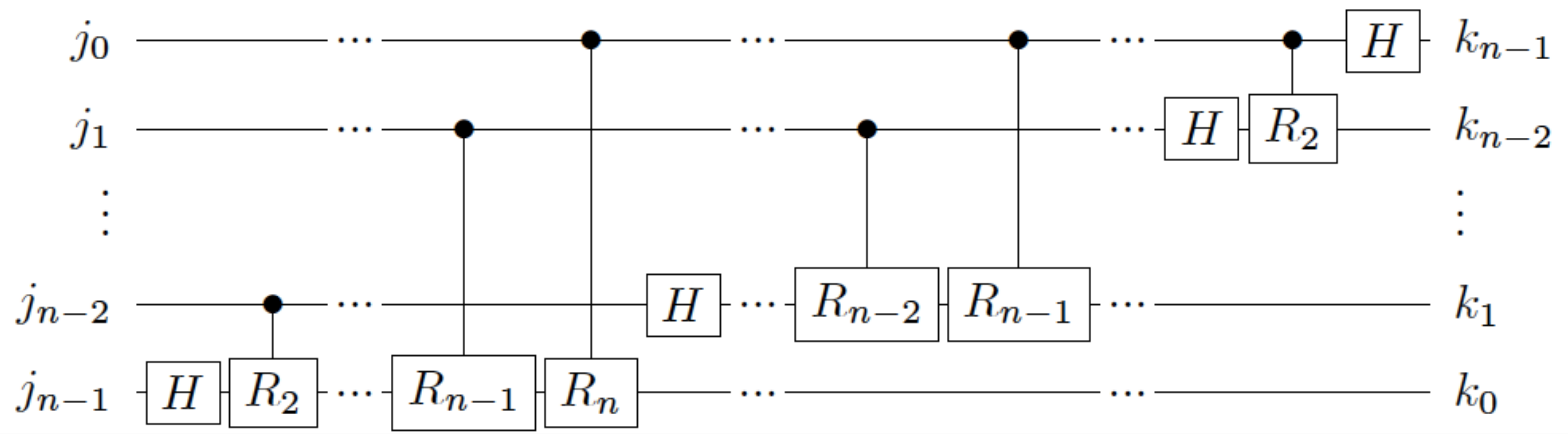
Similarly, applying R_3 to qubit n-1 controlled by qubit n-3,

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(0.j_{n-1}j_{n-2}j_{n-3})}|1\rangle\right)$$

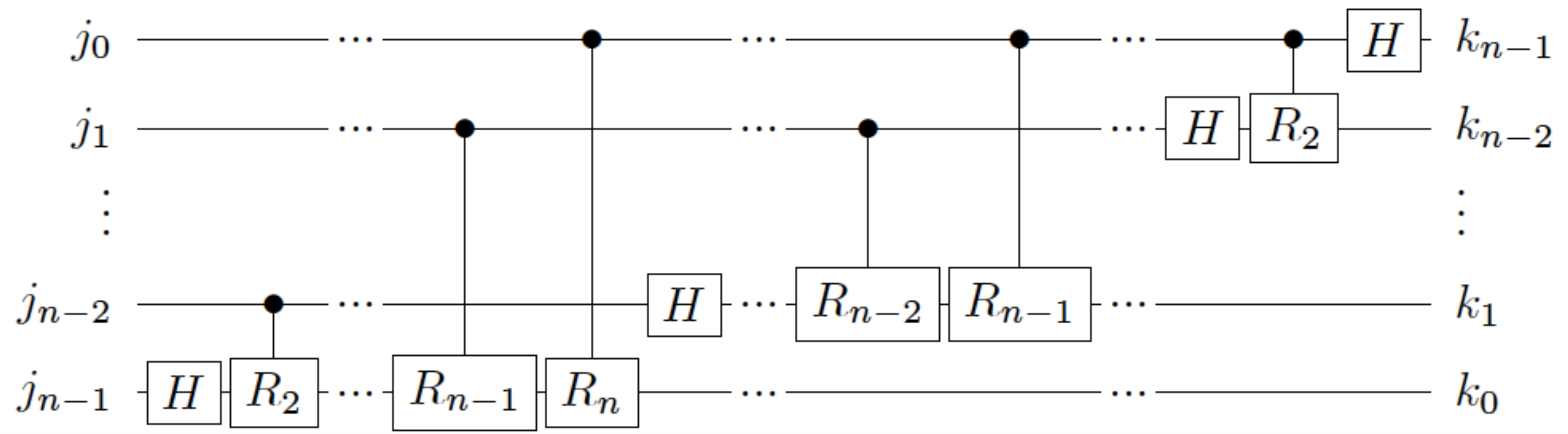
Continuing this through R_n, controlled by qubit 0, the state of qubit n-1 is

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(0.j_{n-1}j_{n-2}j_{n-3}\cdots j_0)}|1\rangle\right)$$

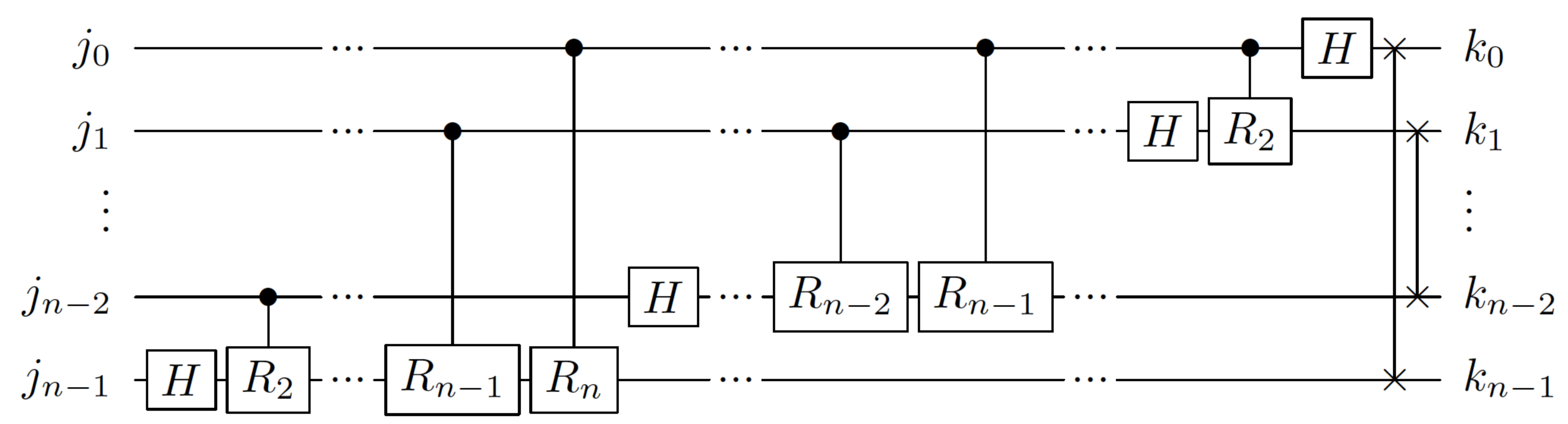
Repeating this procedure to construct the other factors, we get



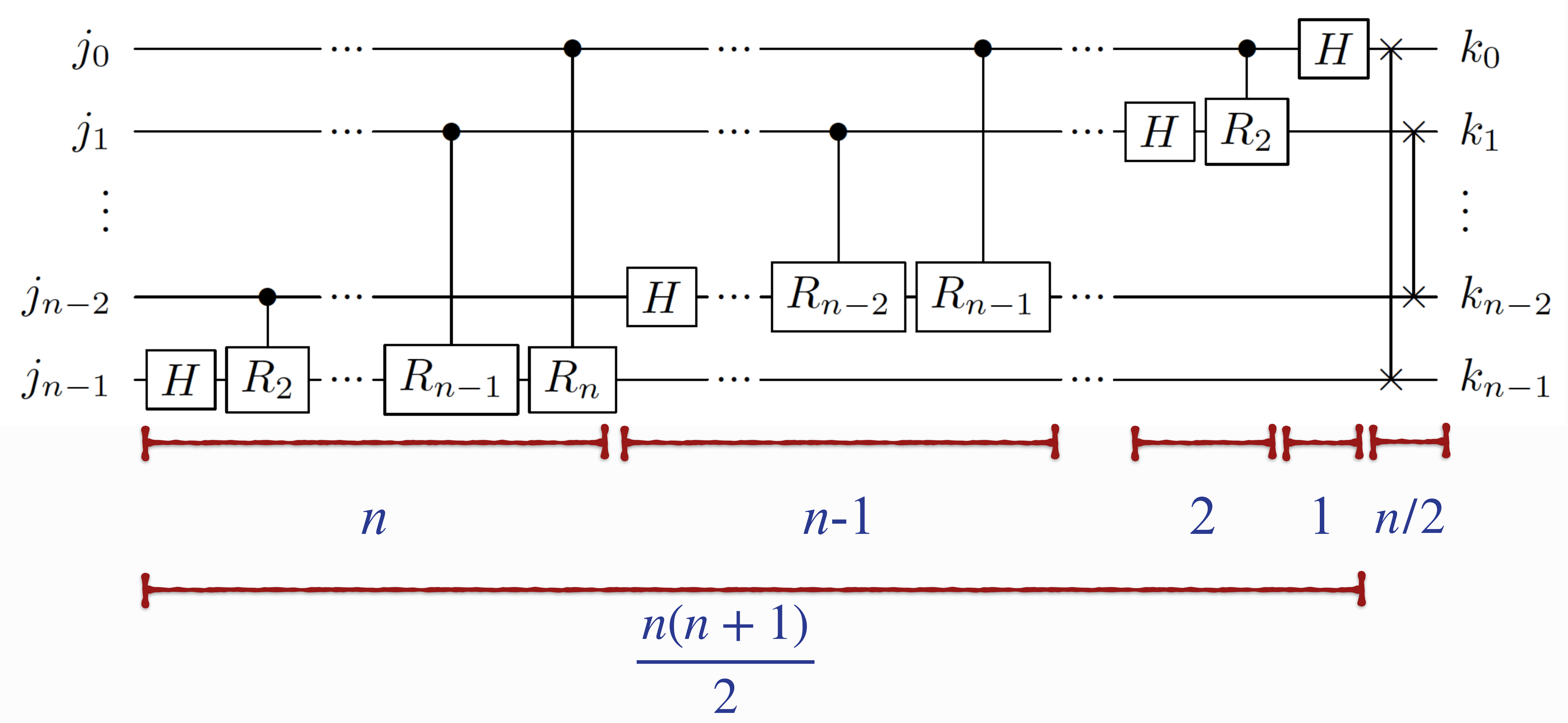
the order of the outputs is reversed.



Just apply SWAP gates



What is the number of quantum gates to implement the QFT?



QFT $\dfrac{n(n+1)}{2} + \dfrac{n}{2} = O(n^2) = O(\log^2 N)$

Classical fast Fourier transform $O(N \log N)$

Exponential speedup

Important differences

FFT

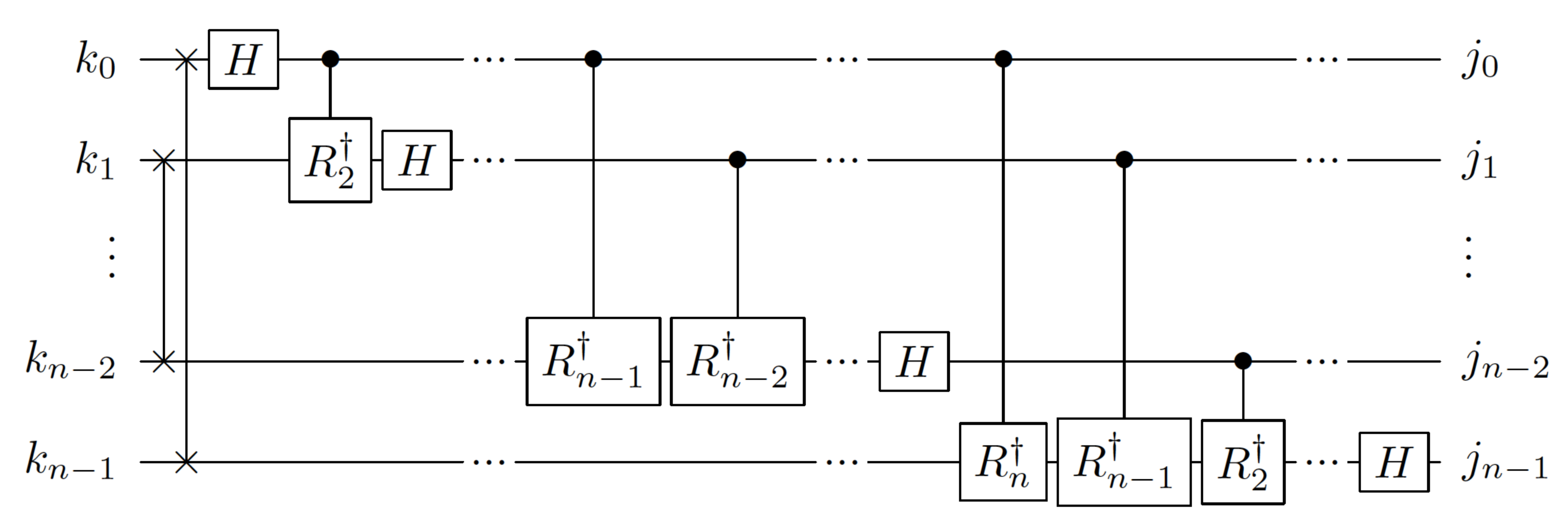
- we have access to all terms of the DFT.

X

QFT

- the result is a superposition quantum state, so we do not have access to these probability amplitudes all at once.
- Measurement in the computational basis returns just the norm-square of the amplitudes.

Exercise: The inverse QFT (IQFT) does the reverse of QFT. Show that its circuit is given by



Quantum phase estimation

Problem: Given a *unitary* matrix U and one of its eigenvectors $|\nu\rangle$, find or estimate its eigenvalue.

The eigenvalue equation for unitary operators takes the form

$$U|v\rangle = e^{i\theta}|v\rangle \qquad \theta \in \mathbb{R}$$

Therefore, estimate its eigenvalues is equivalent to determine the phase θ.

In the case in which the unitary operator has the form

$$U(t,0) = e^{-i\frac{Ht}{\hbar}} \qquad \Rightarrow \theta = \frac{-Ht}{\hbar}$$

it is possible to obtain the Hamiltonian eigenenergies. These phases can contain solutions to problems of interest.

Classical solution

For an N-dimensional space

$$\begin{pmatrix} U_{11} & U_{12} & \dots & U_{1N} \\ U_{21} & U_{22} & \dots & U_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ U_{N1} & U_{N2} & \dots & U_{NN} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix} = e^{i\theta} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix} \Rightarrow \begin{pmatrix} U_{11}v_1 + U_{12}v_2 + \cdots + U_{1N}v_N \\ U_{21}v_1 + U_{22}v_2 + \cdots + U_{2N}v_N \\ \vdots \\ U_{N1}v_1 + U_{N2}v_2 + \cdots + U_{NN}v_N \end{pmatrix} = e^{i\theta} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix}$$

Thus the phase can be obtained

$$e^{i\theta} = \frac{U_{11}v_1 + U_{12}v_2 + \cdots + U_{1N}v_N}{v_1}$$

after the application of N multiplications, N-1 additions and one division,

$O(N)$ steps are necessary to solve the problem classically.

Quantum solution

To describe the eigenvectors of the system of dimension N, we will use n qubits, such that $N = 2^n$,

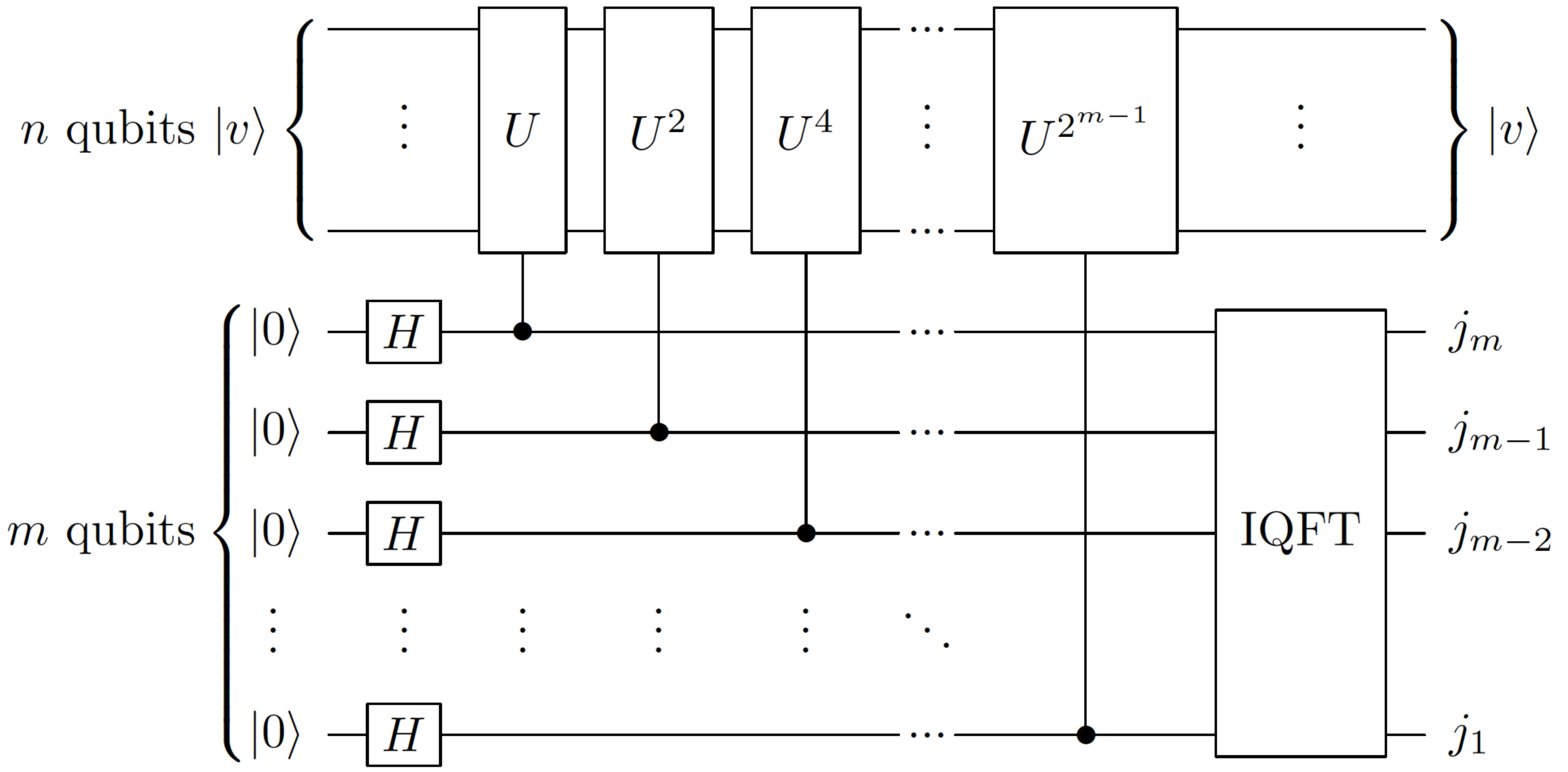
$|\nu\rangle$ - n qubit state

The value of the phase $0 \leq \theta < 2\pi$ will be approximated using m precision-qubits,

$$\theta = 2\pi j \quad \text{s.t.} \quad j = 0.j_1 j_2 \ldots j_m \qquad 0 \leq j < 1$$
$$= \frac{j_1}{2} + \frac{j_2}{4} + \cdots + \frac{j_m}{2^m}$$

Therefore, our task is to find the values of the bits $j_1, j_2, \ldots, j_m$.

The quantum circuit for the phase estimation algorithm is



Step 0 - Creating uniform superposition state.

$$
\begin{aligned}
|+ + \cdots +\rangle|v\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)\ldots\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)|v\rangle \\
&= \frac{1}{\sqrt{2^m}}\left(|0\rangle + |1\rangle\right)\left(|0\rangle + |1\rangle\right)\ldots\left(|0\rangle + |1\rangle\right)|v\rangle.
\end{aligned}
$$

Step 1 - Given the eigenvalue register are ordered as

$$|j_1 j_2 \cdots j_m\rangle$$

the action of the gate U controlled by state $|j_m\rangle$ results

$$\frac{1}{\sqrt{2^m}}\left(|0\rangle+|1\rangle\right)\ldots\left(|0\rangle+|1\rangle\right)\left(|0\rangle+|1\rangle\right)\left(|0\rangle+e^{i\theta}|1\rangle\right)|v\rangle$$

Now, acting with the remaining controlled unitaries, we obtain

$$\frac{1}{\sqrt{2^m}}\left(|0\rangle+e^{2^{m-1}i\theta}|1\rangle\right)\ldots\left(|0\rangle+e^{4i\theta}|1\rangle\right)\left(|0\rangle+e^{2i\theta}|1\rangle\right)\left(|0\rangle+e^{i\theta}|1\rangle\right)|v\rangle$$

Substituting $\theta = 2\pi j$ and $j = 0.j_1 j_2 \ldots j_m$, we find

$$\frac{1}{\sqrt{2^m}}\left(|0\rangle+e^{2\pi i(j_1 j_2 \cdots j_{m-1}.j_m)}|1\rangle\right)\ldots\left(|0\rangle+e^{2\pi i(j_1 j_2.j_3 \cdots j_m)}|1\rangle\right)$$

$$\times\left(|0\rangle+e^{2\pi i(j_1.j_2 \cdots j_m)}|1\rangle\right)\left(|0\rangle+e^{2\pi i(0.j_1 \cdots j_m)}|1\rangle\right)|v\rangle.$$

The bits to the left of the binary point must be ignored because they contribute with integer multiples of 2π. Then,

$$\frac{1}{\sqrt{2^m}}\left(|0\rangle + e^{2\pi i(0.j_m)}|1\rangle\right)\ldots\left(|0\rangle + e^{2\pi i(0.j_3\ldots j_m)}|1\rangle\right)$$
$$\times\left(|0\rangle + e^{2\pi i(0.j_2\ldots j_m)}|1\rangle\right)\left(|0\rangle + e^{2\pi i(0.j_1\ldots j_m)}|1\rangle\right)|v\rangle$$

This is exactly

$$QFT|j_1j_2\cdots j_m\rangle$$

So, if we apply the inverse of QFT (IQFT) to this state, we obtain

$$|j_1j_2\cdots j_m\rangle$$

Now, we are able to estimate the phase to a give precision

$$\theta = 2\pi j = 2\pi\left(\frac{j_1}{2} + \frac{j_2}{2^2} + \ldots + \frac{j_m}{2^m}\right)$$

Quantum computational complexity

Number of quantum gates to estimate the eigenvalue to m bits of precision:

- m Hadamard gates;
- m controlled-U^P operations;
- IQFT on m bits - $O(m^2)$

→ **"Total" cost $O(m^2)$**

Hypotesis:

- We are able to prepare the initial state $|\nu\rangle$;
- We are able to execute m controlled-U^P operations.

Speedup?

Classical cost $N = 2^n$ → It depends on the values of m and n.

Period of Modular Exponentiation

Modular exponentiation is taking powers of a number modulo some other number.

"mod" refers to *modulus* = the remainder of a division

$$
\begin{aligned}
2^0 \bmod 7 &= 1 \bmod 7, \\
2^1 \bmod 7 &= 2 \bmod 7, \\
2^2 \bmod 7 &= 4 \bmod 7, \\
2^3 \bmod 7 &= 8 \bmod 7 = 1 \bmod 7, \\
2^4 \bmod 7 &= 16 \bmod 7 = 2 \bmod 7, \\
2^5 \bmod 7 &= 32 \bmod 7 = 4 \bmod 7, \\
2^6 \bmod 7 &= 64 \bmod 7 = 1 \bmod 7, \\
2^7 \bmod 7 &= 128 \bmod 7 = 2 \bmod 7, \\
2^8 \bmod 7 &= 256 \bmod 7 = 4 \bmod 7,
\end{aligned}
$$

Period $r = 3$

$$
\begin{aligned}
3^0 \bmod 10 &= 1 \bmod 10, \\
3^1 \bmod 10 &= 3 \bmod 10, \\
3^2 \bmod 10 &= 9 \bmod 10, \\
3^3 \bmod 10 &= 27 \bmod 10 = 7 \bmod 10, \\
3^4 \bmod 10 &= 81 \bmod 10 = 1 \bmod 10, \\
3^5 \bmod 10 &= 243 \bmod 10 = 3 \bmod 10, \\
3^6 \bmod 10 &= 729 \bmod 10 = 9 \bmod 10, \\
3^7 \bmod 10 &= 2187 \bmod 10 = 7 \bmod 10, \\
3^8 \bmod 10 &= 6561 \bmod 10 = 1 \bmod 10,
\end{aligned}
$$

Period $r = 4$

The *period* or *order* is the smallest positive exponent r such that

$$a^r \bmod N = 1 \bmod N$$

If a and N are *relatively prime* (they share no common factors except 1), the repeated pattern always comes out.

Classical solution

- The total number of elementary binary arithmetic operations for an individual modular exponentiation is $O(n^2)$. n is the number of bits used to write the power in binary representation.
- However, there are several individual modular exponentials. This turns this method expensive.
- See pages 329-331 of Ref. [Wong] for a detailed analysis.

T. G. Wong, Introduction to Classical and Quantum Computing (2022). https://www.thomaswong.net/

Quantum solution

Lets consider a quantum gate U that performs *modular multiplication*

$$U|y\rangle = |ay \bmod N\rangle \qquad 0 \le y \le N-1$$

Applying U repeatedly on the state $|1\rangle$

$$\begin{aligned}
U^0|1\rangle &= |1 \bmod N\rangle = |a^0 \bmod N\rangle, \\
U^1|1\rangle &= |a \bmod N\rangle = |a^1 \bmod N\rangle, \\
U^2|1\rangle &= |a^2 \bmod N\rangle, \\
U^3|1\rangle &= |a^3 \bmod N\rangle, \\
&\vdots \\
U^r|1\rangle &= |a^r \bmod N\rangle = |a^0 \bmod N\rangle.
\end{aligned}$$

U implements exactly the modular exponential $a^x \bmod N$

r is the period

Due to the cyclic character of the states $|a^x \mod N\rangle$, they can be superposed to create an eigenstate of U

$$|v_s\rangle = \frac{1}{\sqrt{r}}\Big(e^{-2\pi i s(0)/r}\big|a^0 \bmod N\big\rangle + e^{-2\pi i s(1)/r}\big|a^1 \bmod N\big\rangle + \dots$$

$$+ e^{-2\pi i s(r-2)/r}\big|a^{r-2} \bmod N\big\rangle + e^{-2\pi i s(r-1)/r}\big|a^{r-1} \bmod N\big\rangle\Big) \qquad 0 \leq s \leq r-1$$

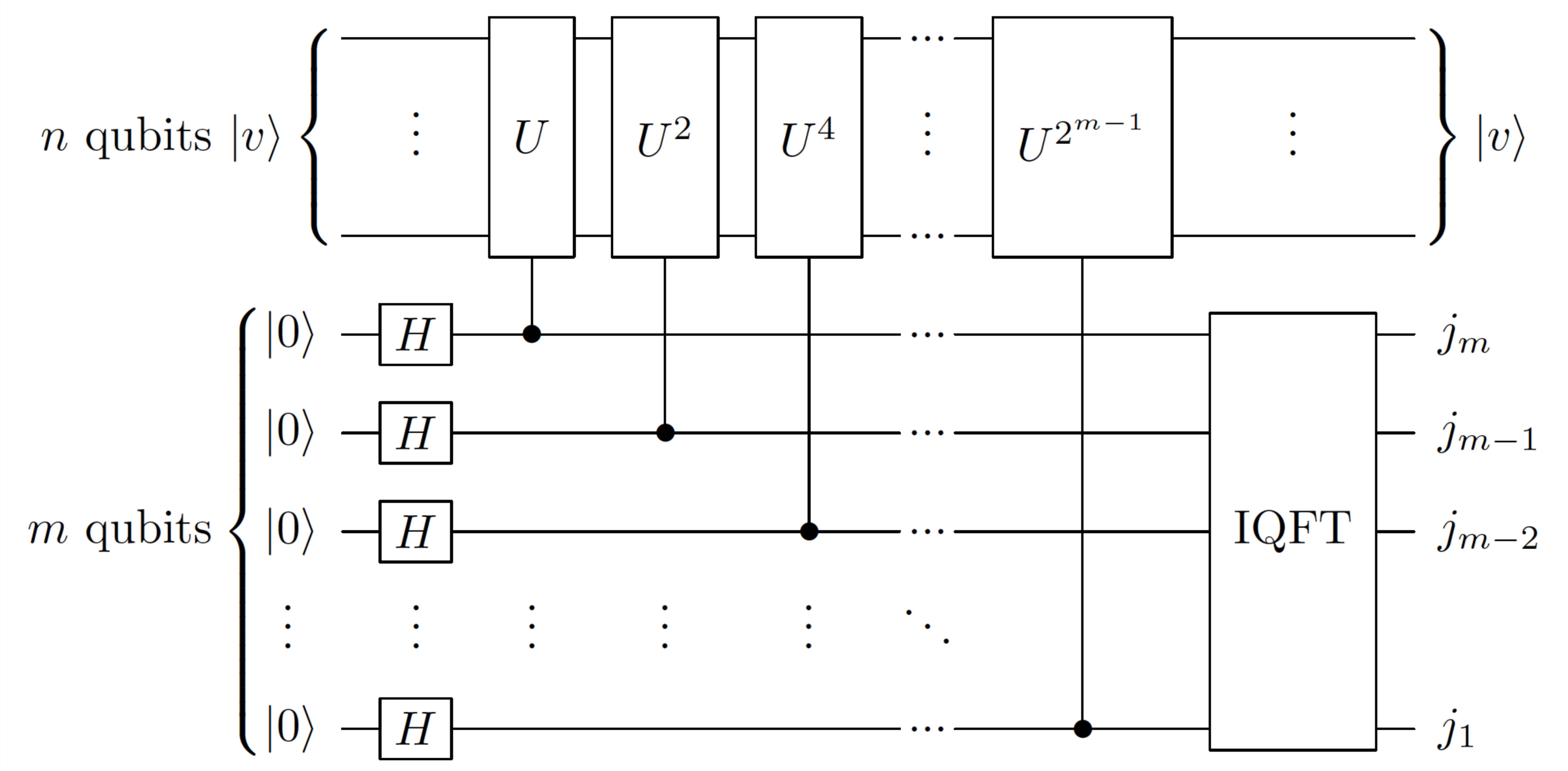
$$= \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{-2\pi i s k/r}\Big|a^k \bmod N\Big\rangle.$$

Exercise: Show that

$$U|\nu_s\rangle = e^{i2\pi s/r}|\nu_s\rangle\,.$$

Observe that r is registered in the phase $e^{i2\pi s/r}$ for some value of s, then we can use the *phase estimation algorithm* to obtain r.

As seen before, the circuit for phase estimation is



How to construct the eigenstates $|\nu_s\rangle$ of U?

The action of the controlled U gates CU^{2^j} are determined by

$$CU^{2^j}|z\rangle|y\rangle = |z\rangle\left|a^{z2^j} y \bmod N\right\rangle$$

$|z\rangle$ – control qubit

Trick: instead of preparing a single eigenvector of U, we prepare the following superposition of them

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|v_s\rangle$$

Exercise: Show that

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|v_s\rangle = |1 \bmod N\rangle$$

This state is easy to prepare

$$|00\ldots01\rangle = |1 \bmod N\rangle$$

However, when we measure the phase of this state, there is a probability $1/r$ of obtaining one of the eigenvalues $e^{i2\pi s/r}$.

The phase is approximated using m bits

$$2\pi\frac{s}{r} \approx 2\pi 0.j_1 j_2 \ldots j_m$$

How to obtain r?

Continued fraction method

It is a method to approach a real number by its closest rational

$$0.j_1 j_2 \ldots j_m \approx \frac{s}{r}$$

Considering the number of precision qubits $m = O(n)$, the continued fraction method takes $O(n^3)$.

We can discuss more about this method during the exercise classes.

Total cost of the modular exponentiation

- 1 X gate
- m Hadamard gates
- $O(n^3)$ controlled U^{power}
- $IQFT\, O(n^2)$
- Continued fraction $O(n^3)$

Total # of gates: $O(n^3)$

It is efficient!!

Factoring algorithm

The goal of factoring is to find prime numbers p and q such that

$$N = pq\,.$$

N is an n-bit number

Classical solution: the best known classical algorithm is the *number field sieve*. To factor an n-bit number, its runtime is roughly

$$O\left(e^{n^{1/3}}\right)$$

Shor's factoring algorithm

1 - Pick any number $1 < a < N$.

2 - Calculate the $\gcd(a, N)$.

If $\gcd(a, N) \neq 1$, then we have found $p = \gcd(a, N)$. So, $q = N/p$ and we are done factoring.

Else $\gcd(a, N) = 1$ continue to the next step.

3 - Find the period r of $a^x \mod N$. ← Quantum advantage $O(n^3)$

If r is odd, go back to step 1 and pick a different a.

Else r is even, calculate $a^{r/2} \mod N$.

If $a^{r/2} \mod N = N - 1$ go back to step 1 and pick a different a.

Else we have found r.

4 - Then we have factored

$$p = \gcd(a^{r/2} - 1, N)$$
$$q = \gcd(a^{r/2} + 1, N)$$

P. W. Shor, *Proceedings 35th Annual Symposium on Foundations of Computer Science*. IEEE Comput. Soc. Press: 124 (1994).

Exercise: Use "Shor's" algorithm (previous slide) to find the factors of $N = 35$.