Advanced Quantum Computing, Assignment 2

Pradyot Pritam Sahoo

May 4, 2025

Q1.

The continued fraction expansion of $\frac{77}{65}$ is computed as follows:

$$\begin{aligned}
\frac{77}{65} &= 1 + \frac{12}{65} \\
&= 1 + \frac{1}{\frac{65}{12}} \\
&= 1 + \frac{1}{5 + \frac{5}{12}} \\
&= 1 + \frac{1}{5 + \frac{1}{\frac{12}{5}}} \\
&= 1 + \frac{1}{5 + \frac{1}{2+\frac{2}{5}}} \\
&= 1 + \frac{1}{5 + \frac{1}{2+\frac{1}{\frac{5}{2}}}} \\
&= 1 + \frac{1}{5 + \frac{1}{2+\frac{1}{2+\frac{1}{2}}}}
\end{aligned}$$

We now have to compute the convergents of the continued fraction, which give approximations $\frac{s}{r} \approx \phi$. Then choose the r in which $min(|s/r - \phi|)$.

Q2.

Factoring N = 63 using Shor's Algorithm with x = 8:

- Choose $x = 8$, which is coprime to 63: $\gcd(8, 63) = 1$.
- Use the order-finding subroutine to find the smallest r such that:

  $$8^r \equiv 1 \mod 63$$

  We find $r = 2$ because $8^2 = 64 \equiv 1 \mod 63$.
- Since r is even, compute:

  $$\gcd(8^{r/2} - 1, 63) = \gcd(7, 63) = 7$$
  $$\gcd(8^{r/2} + 1, 63) = \gcd(9, 63) = 9$$
- So, 63 factors as 7×9, and since $9 = 3^2$, we get:

  $$63 = 3^2 \cdot 7$$

Q3.

Given the unitary operator U defined by:

$$U|k\rangle = |xk \bmod N\rangle,$$

where x is coprime to N. For any positive integer z, repeated application of U yields:

$$U^z|k\rangle = |x^z k \bmod N\rangle.$$

Thus, U^z implements *modular multiplication* by x^z.

Stepwise Process to Find U^z for t Clock Qubits

- **Initialization:** Prepare the system with t clock qubits in the superposition state $\frac{1}{\sqrt{t}}\sum_{j=0}^{t-1}|j\rangle$ and a second register in the state $|1\rangle$. So the combined state is:

$$|\psi_0\rangle = \frac{1}{\sqrt{t}}\sum_{j=0}^{t-1}|j\rangle|1\rangle$$

- **Apply the Controlled-U^{2^j}:** For each qubit j in the first register, apply a controlled-U^{2^j} operation:

$$|\psi_1\rangle = \frac{1}{\sqrt{t}}\sum_{j=0}^{t-1}|j\rangle U^{2^j}|1\rangle = \frac{1}{\sqrt{t}}\sum_{j=0}^{t-1}|j\rangle|x^{2^j} \mod N\rangle.$$

  Here, U^z (where $z = 2^j$) is implemented via **modular exponentiation**:

- **Inverse-QFT :** After phase estimation, apply the inverse QFT to the first register to extract the phase (related to the order r of $x \mod N$).

Time Complexity

- Each modular multiplication circuit can be built using $O((\log N)^2)$ elementary gates.
- To compute U^z, we apply a sequence of controlled-U^{2^j} operations, one for each of the t clock qubits.
- Each of these operations involves modular multiplication by $x^{2^j} \mod N$, which costs $O((\log N)^2)$ gates.
- Since there are $t = O(\log N)$ such operations in Shor's algorithm, the total time complexity is: $O((\log N)^3)$.