Technical description of Shor's order-finding circuits

qBitTensor Labs

1 Background

Rivest-Shamir-Adleman (RSA) is a ubiquitous, standardized system used to send and receive cryptographically secure messages. Notably, the security of RSA relies fundamentally on the practical difficulty of recovering two prime numbers p and q from their product $N = pq$. Peter Shor discovered that a quantum computer can factor N in time polynomial in $\log N$ by reducing the problem to that of *group-theoretic order finding* and extracting that order with a technique known as phase estimation [1,2]. This is not a niche algorithm: It is the paradigmatic example where quantum interference translates a periodic structure in arithmetic into a signal recoverable via (inverse) Fourier transform. Learning this pathway – from modular multiplication to eigenphases to rational approximants – is foundational for both cryptographic literacy and for understanding why coherent quantum devices are different from classical ones.

Here we describe the mathematical underpinnings of how Shor's algorithm works, including some basic number theory and group theory, and how quantum computation is applied to the general problem space.

1.1 Number theory and the integer multiplicative group modulo N

Some mathematical theory is required to understand what's going on here. We'll start off with the definition of a group – while some group theory was relevant for our hidden stabilizers circuit, proper understanding of Shor's algorithm will require somewhat more formal treatment.

Definition 1. (Group)
A *group* $\mathcal{G} = (G, \times)$ is a set of elements G together with a binary operation $\times$ such that:

- G is *closed* under $\times$: $u \times v \in G \quad \forall u, v \in G$;
- $\times$ is *associative*: $(u \times v) \times w = u \times (v \times w) \quad \forall u, v, w \in G$;
- G has an *identity element*: there exists an $e \in G$ such that $u \times e = u = e \times u \quad \forall u \in G$
- Elements are *invertible*: $\exists a^{-1} \in G : a \times a^{-1} = e = a^{-1} \times a \quad \forall a \in G$

Groups find a remarkably wide range of applications in, for example:

- computer graphics, where they describe rotations in N-dimensional space;
- physics, where they permit description of invariant quantities like color charge in particle physics or phases of matter in condensed matter physics through Noether's theorem;
- number theory and cryptography, where they describe specialized numerical encodings of information and their properties through Fermat's little theorem;
- quantum information, where they model the space of all unitary transformations that can be applied to a quantum state as well as invariants in quantum error correction.

In this section, we'll deal with a single, specific group, which is the multiplicative group of integers, modulo some non-negative N. This group, which we will denote $\mathcal{G}_N$ for brevity, consists of $|\mathcal{G}_N| = N$ unique elements with group operation $\times$ set to ordinary integer multiplication modulo N. This group is obviously finite, and an important feature of every such group is that every element $a \in G$

has an *order* r such that $a^{\times r} = e$ – that is, r-fold application of a through the group operation is equal to the group identity e. In the case of $\mathcal{G}_N$, this amounts to the statement

$$a^r \equiv 1 \bmod N. \tag{1}$$

Every element a has an order in a finite group, and we will denote this order as $\mathrm{ord}_N(a)$. Here, we'll focus on the special case of some odd, composite N and some a that is coprime with N; i.e. $\gcd(a, N) = 1$. One relevant insight from number theory is that when r is even and $a^{r/2} \not\equiv -1 \bmod N$, the numbers

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

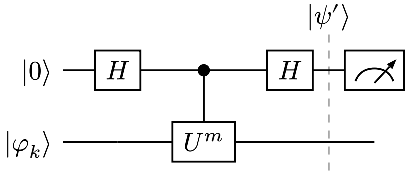
are nontrivial factors of N [1,2]. Through this fact, we are thus provided a route to factoring N, provided we can find a and, more importantly, $\mathrm{ord}_N(a)$. In the context of RSA encryption, we are already given some semiprime $N = pq$, so any $a \neq p, q$ is valid, but the problem of finding $\mathrm{ord}_N(a)$ unfortunately remains hard (for classical computers, that is). Shor's algorithm solves the problem of factoring N along this line of reasoning by first applying this (well-known) piece of number theory, and then using a quantum computer to solve the sub-problem of group-theoretic order-finding.

1.2 Phase estimation for eigenvalues of a unitary operator

On an operational level, all of quantum computing deals with *unitary* linear transformations with the sole exception of measurement. Concretely, every such transformation can be represented as a square complex-valued matrix U with the additional properties $\det(U) = 1$ and $U^\dagger U = UU^\dagger = I$, where $(\cdot)^\dagger$ denotes the conjugate transpose.

These properties have an important corollary: Given a Hilbert space $\mathcal{H} = (H, \langle\cdot,\cdot\rangle)$ on which U acts with elements $x, y \in \mathcal{H}$, we have $\langle Ux, Uy\rangle = \langle x, y\rangle$. Thus we can say that for any $v \in \mathcal{H}$, U *preserves the (standard inner product-induced) norm of* v, and hence that every eigenvalue λ_k of U has unit magnitude, $|\lambda_k| = 1$. Since we can further note that all eigenspaces of U are orthogonal, we then arrive at the final important fact, which is that every eigenvalue takes the form of a complex exponential with a distinct phase: $\lambda_k \equiv e^{i2\pi\theta_k}\ \forall k \in \{1, ..., \dim(\mathcal{H})\}, 0 \leq \theta_k < 1$.

Quantum phase estimation (QPE) is a technique that can be used to measure the phases θ_k for some such given U. The working principle behind it is based on, minimally, a toy circuit known as the Hadamard test:



In this circuit, we can calculate $|\psi'\rangle$ by hand to find

$$|\psi'\rangle = e^{i2\pi\frac{m\theta_k}{2}}\left(\cos\left(2\pi\frac{m\theta_k}{2}\right)|0\rangle - i\sin\left(2\pi\frac{m\theta_k}{2}\right)|1\rangle\right)|\varphi_k\rangle \tag{3}$$

and hence that the probability $P(1)$ of the final measurement yielding a 1 has *fringes* that depend on the value of θ_k as

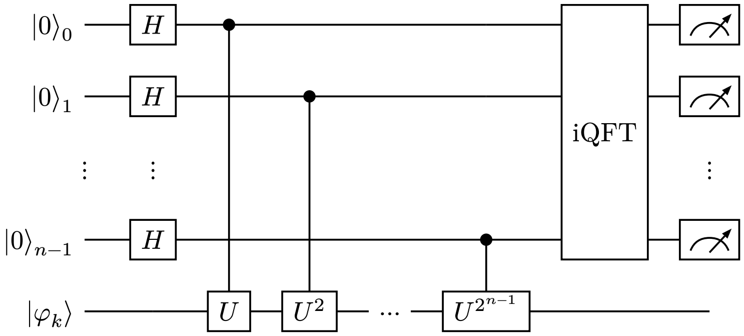
$$\Pr(1) = |\langle 1|\psi'\rangle|^2 = \frac{1 - \cos(2\pi m\theta_k)}{2}. \tag{4}$$

More specifically, we can see that $P(1)$ reaches local maxima where $2m\theta_k$ is an odd integer. If we then consider the binary expansion of θ_k, i.e.

$$\theta_k = 0 + \frac{b_1}{2} + \frac{b_2}{4} + \cdots + \frac{b_n}{2^n} \equiv 0.b_1b_2...b_n \tag{5}$$

then the Hadamard test, when run for $m = 2^j$, estimates b_j.

To then estimate the full binary expansion, the canonical picture of QPE simply performs simultaneous Hadamard tests using enough qubits to achieve a desired precision:



In this full QPE circuit, the iQFT is known as the inverse quantum Fourier transform. Its implementation is not so important here, and can be seen as merely a generalization of the second Hadamard gate in the Hadamard test circuit above – although it is more complicated, it achieves the same purpose as a classical Fourier transform, which is to map the phases accrued through controlled applications of U to computational basis states. Through the iQFT, the phase θ_k is mapped to a peak in the final measurement probabilities, indicating the bits of the binary expansion of θ_k. The height of the peak follows the well-known bound

$$\Pr\left(\left|\frac{s}{2^n} - \theta_k\right| \leq \frac{1}{2^{n+1}}\right) \geq \frac{4}{\pi^2} \tag{6}$$

where $s = \lfloor \theta_k 2^n \rceil$.

1.3 Shor's algorithm

Shor's algorithm solves the integer factoring problem by using quantum phase estimation as a subroutine to efficiently find the order of an element $a \in \mathcal{G}_N$. Broadly, the algorithm goes like this: Given some semiprime N, choose some $1 < a < N$ – because N is semiprime, we are guaranteed that Equation 2 applies to any such a.

Next, define a unitary, linear operator $U : |x\rangle \mapsto |ax \bmod N\rangle$ on an n-qubit work register. In the computational basis, U is then a permutation matrix whose eigenpairs can be written $\{(|\psi_k\rangle, \exp(i2\pi k/r))\}$ for $k \in \{0, ..., 2^{n-1}\}$. Notably, U *encodes* $\mathrm{ord}_N(a)$ in its eigenvalues: r is exactly the order of a in $\mathcal{G}_N$, which means it can be elucidated through application of QPE on U.

This is the key innovation in Shor's algorithm – although there is no known method to classically compute the order of a given group element short of simply exponentiating the group element (which has an exponential runtime bound), QPE demands only a polynomial-depth circuit. Since U can also be implemented using a quantum ripple-carry adder [3,4] in only polynomial depth (and both it and QPE require only polynomially many qubits in this case), we hence have an efficient method to factor N.

More concretely, a naive implementation of Shor's algorithm uses a work register of $n = \lceil \log_2 N \rceil$ qubits to compute the modular exponentiation of a, controlled by a counting register of t qubits. Application of the full QPE procedure then produces a measurement outcome $s \in \{0, ..., 2^t - 1\}$ drawn from a probability distribution that is sharply concentrated near $s = \lfloor 2^t k/r \rceil$, such that $\frac{s}{2^t}$ is a good rational approximation to k/r [5]. Since Shor's original formulation, however, improvements to the quantum step of the algorithm have been proposed, making use of semi-classical means that produce equivalent output using fewer qubits via interleaved single-qubit Fourier steps with classical feed-forward operations [6].

The final step in the algorithm is to recover the actual period r of a using a rational approximation algorithm based on continued fractions. Given a sampled output state s as above, we take $x = s/2^t$, and compute a best continued-fraction convergent u/v with $v \leq 2^t$, and accept it if $|x - u/v| \leq 1/2^{t+1}$. Each accepted v is equal to $r/\gcd(k, r)$ for some k; thus the estimator

$$\hat{r} = \operatorname{lcm}\{v_1, v_2, ...\} \tag{7}$$

formed from several such sampled v converges rapidly in practice. An additional refinement step divides $\hat{r}$ by small primes p whenever $a^{\hat{r}/p} \equiv 1 \bmod N$. After an estimate for the order r is recovered through some means, Equation 2 can be applied to finally return a factor of N.

2 Problem definition

Unlike peaked circuits and hidden stabilizers, these order-finding circuits will be Subnet 63′s first contact with "real-world" quantum computing problems. From the perspective of the subnet, these phase-estimation/order-finding circuits are an excellent addition to the current family of circuits because quantum phase estimation is very well-ordered, which means they are fast to generate in a way that is notably only weakly dependent on the content of the problem − in contrast to hidden stabilizers, which required obfuscation, and peaked circuits, which required solution of a large tensor network optimization problem. Rather, order-finding circuits are much more freely scaled to larger numbers of qubits.

Additionally, these circuits begin to mix in a small amount of classical post-processing on the output of the proper quantum circuits in the form of analyzing the final probability distribution (or measurement outcomes for sampling-based approaches) in a way that is much more typical of real-world problems. As always, we will provide sample code with a basic implementation of a full solution, but miners should feel free to innovate on both the quantum and classical steps of the underlying algorithm.

Problem 1. (Shor's order-finding)
Given a quantum circuit implementing quantum phase estimation on multiplication by some a modulo some N with $\gcd(a, N) = 1$, execute the circuit and compute the order r of the multiplication.

3 Additional theory for miners

This section collects some standard facts and strategies that can yield a practical edge for ambitious miners.

Single-sample and multi-sample recovery

A classical criterion from Diophantine approximation states that if $\alpha - p/q| \leq 1/(2q^2)$, then p/q is a convergent of the continued fraction of α. Taking $\alpha = k/r$ for some positive k where r is order of the underlying $a \in \mathcal{G}_N$ as specified above, and using $s/2^t$ as our estimate of α (s an outcome drawn from the appropriate t-qubit output register), the bound $|s/2^t - k/r| \leq 1/2^{t+1}$ suffices whenever $q \leq 2^t$. Thus a single good outcome yields a divisor $q \mid r$. For multiple outcomes with k effectively uniform on $\{0, ..., r-1\}$, each denominator equals $r/\gcd(k, r)$. After m independent samples, the probability that the least common multiple of accepted denominators equals r obeys

$$\Pr(\operatorname{lcm} = r) \geq \prod_{p \mid r} (1 - p^{-m}), \tag{8}$$

which increases quickly with m (for example, each factor in the right-hand side is at least $1 - 1/8$ for $m = 3$).

Peak spacing via gcd

Let $\mathcal{S}$ be a set of high-probability samples (integers in the range $[0, 2^t)$). In the output of the QPE subroutine, peaks lie close to $k2^t/r$ for some k, and the spacings between the peaks obey $\Delta s \approx (k' - k)2^t/r$. Rounding to the nearest integer (and ignoring wraparound for large t), the quantity

$g = \gcd\{\Delta s\}$ concentrates near $2^t/r$; taking $\hat{r} = \lfloor 2^t/g \rceil$ often succeeds and is robust to a modest fraction of spurious hits. A practical refinement computes several pairwise gcds among the largest peaks and takes a median of the implied $2^t/g$ values.

Lattice-based simultaneous approximation

With multiple samples $s_i/2^t \approx k_i/r$, seek r minimizing $\sum_i |rs_i - k_i 2^t|$ over integers k_i. This can be cast as a search for short vectors in low-rank lattices; even 2D Gaussian reduction on carefully chosen pairs can recover r at moderate t, and LLL-style heuristics perform well as the sample set grows. Empirically, this competes with the lcm-from-continued-fractions method and can be less sensitive to outliers [5].

Bayesian and MLE estimators

Treat the k_i as latent and score each candidate r by a product of Dirichlet-kernel values at the sampled s_i. One may hill-climb over plausible r and validate by checking $a^r \equiv 1 \bmod N$ classically. Alternatively, maintain a discrete posterior over r updated by each new measurement. These approaches can outperform purely number-theoretic recovery when the iQFT is aggressively approximated or shot budgets are tight [6]

Robustness to noise and approximations

Use an acceptance window of half-width on the order of $2^t/2r$ around each hypothesized peak to include counts; weight denominators by probability; and trim outliers before lcm/gcd aggregation. Approximate or semiclassical QFT chiefly broadens peaks but leaves their centers essentially unbiased [6]. Increasing t by one qubit roughly halves the admissible window.

References

[1] P. W. Shor, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring." In Proceedings 35th Annual Symposium on Foundations of Computer Science 124 (1994).

[2] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer." SIAM Journal on Computing **26**, 1484 (1997).

[3] S. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. P. Moulton, "A New Quantum Ripple-Carry Addition Circuit." arXiv:0410184 (2004).

[4] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter, "Elementary gates for quantum computation." Phys. Rev. A **52**, 3457 (1995).

[5] A. Y. Kitaev, "Quantum Measurements and the Abelian Stabilizer Problem." arXiv:9511026 (1995).

[6] R. B. Griffiths, and C.-S. Niu, "Semiclassical Fourier Transform for Quantum Computation." Phys. Rev. Lett. **76**, 3228 (1996).