Regev's factoring algorithm

See how several modular directions and classical lattice processing can reduce the size of a factoring circuit per run.

Research 18 min lesson Dr. Charith Jayasekara

By the end, you'll be able to

- Trace a small modular relation into a lattice constraint.
- Separate per-run circuit cost from repetition and classical work.
- Distinguish the original heuristic from subsequent rigorous variants.

Before you begin: Shor's algorithm, modular arithmetic, Fourier sampling and an introduction to lattices.

From a period to a lattice of relations

Shor's factoring method finds repetition in modular powers. Regev's approach uses several small bases simultaneously and looks for a multidimensional collection of relations. A lattice is an orderly set of integer combinations of basis vectors. Here its points describe exponent choices that make a modular product equal one.

The worked relation below uses ordinary multiplication and greatest common divisors. The full proof requires advanced number theory.

Let N be an odd composite integer, after straightforward cases such as prime powers have been handled classically. Choose small integers $b_1, \ldots, b_d$ coprime to N, and set $a_i = b_i^2$. Consider

$$f(\mathbf{z}) = \prod_{i=1}^{d} a_i^{z_i} \pmod{N}.$$

Negative exponents mean modular inverses, which exist because the chosen bases are coprime to N. The relation lattice is

$$L = \{\mathbf{z} \in \mathbb{Z}^d : f(\mathbf{z}) = 1 \pmod{N}\}.$$

For a relation $\mathbf{z}$, define $u = \prod_i b_i^{z_i} \pmod{N}$. Since the a_i are squares, $u^2 = 1 \pmod{N}$. If u is a nontrivial square root of one, the classical calculation $\gcd(u-1, N)$ gives a nontrivial factor.

Checkpoint. Does every relation guarantee a factor? No. A relation producing $u = 1$ or $u = -1 \pmod{N}$ is unhelpful for this step. The post-processing must obtain a useful relation.

Worked example: a complete small relation

Use $N = 15$, $b_1 = 2$ and $b_2 = 7$. Both are coprime to 15, and

$$a_1 = 4, \qquad a_2 = 49 \equiv 4 \pmod{15}.$$

The vector $(1, -1)$ is a relation because $4 \cdot 4^{-1} = 1 \pmod{15}$. Since $7^{-1} = 13 \pmod{15}$, its corresponding square root is

$$u = 2 \cdot 7^{-1} \equiv 2 \cdot 13 \equiv 11 \pmod{15}.$$

Check $11^2 = 121 \equiv 1 \pmod{15}$. Then

$$\gcd(11 - 1, 15) = 5, \qquad \gcd(11 + 1, 15) = 3.$$

This relation was chosen by inspection. The quantum and lattice algorithms obtain suitable information for large, unknown instances.

In this toy example, $f(z_1, z_2) = 4^{z_1 + z_2} \pmod{15}$, and 4 has order two. Therefore L consists of integer points with $z_1 + z_2$ even. One basis is $(1, 1), (1, -1)$. The relation lattice can be drawn as a checkerboard subset of the integer grid.

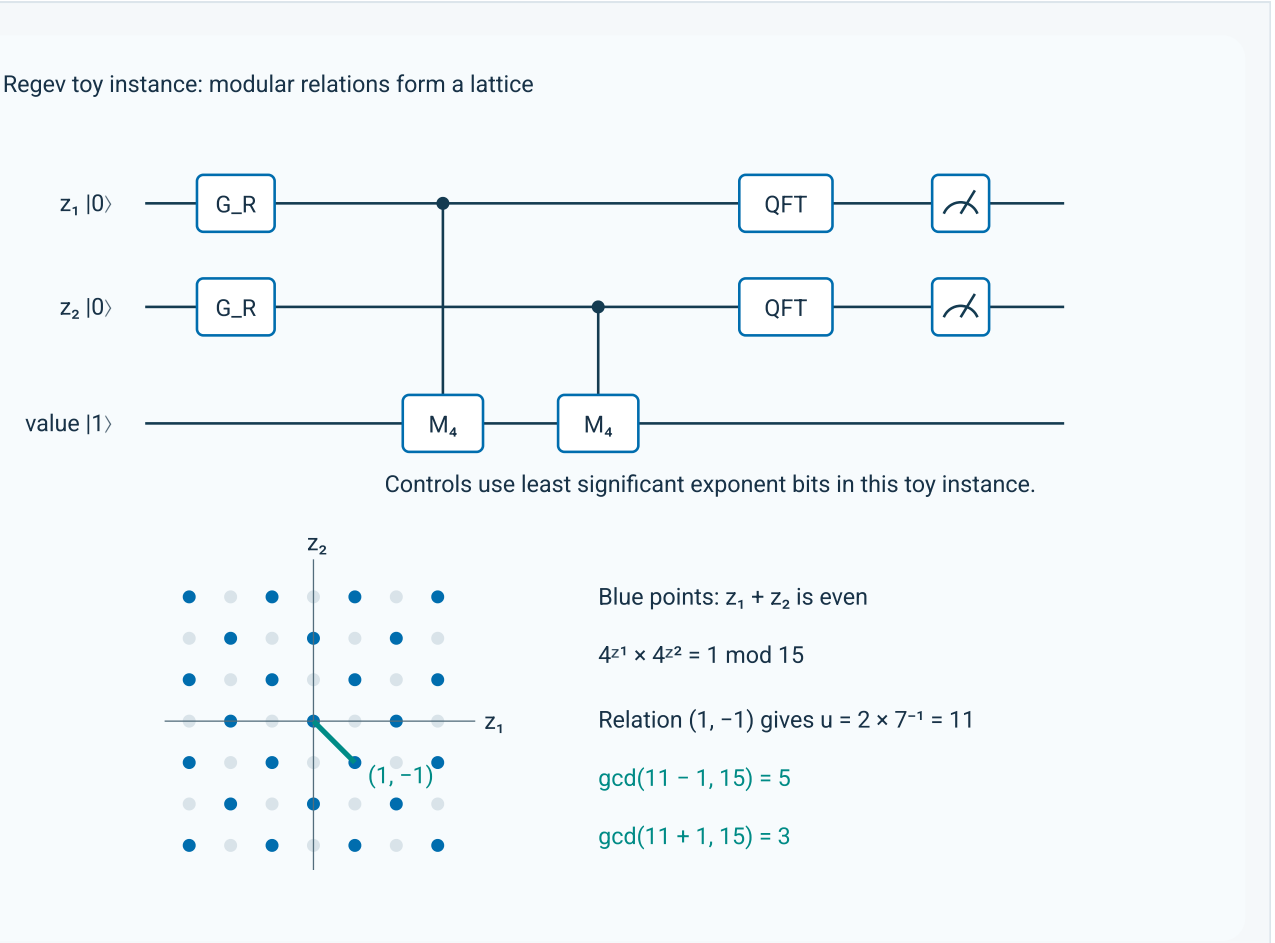
What the quantum circuit samples

The algorithm prepares a smoothly weighted superposition over exponent vectors, coherently computes $f(\mathbf{z})$ into a value register, then Fourier transforms the exponent registers and measures them. Ignoring or measuring the value register groups exponents with the same modular product. Differences between such exponents lie in L.

Fourier sampling gives approximate information about the dual lattice,

$$L^* = \{\mathbf{y} \in \mathbb{R}^d : \mathbf{y} \cdot \mathbf{z} \in \mathbb{Z} \text{ for every } \mathbf{z} \in L\}.$$

Several independent runs provide samples for classical lattice processing. The processing reconstructs enough relation information to attempt the square-root and gcd step. The measurement output is therefore an input to a classical algorithm, rather than a list of factors by itself.



A toy modular-multiplication circuit and the even-sum relation lattice, with factors recovered from a useful relation.

The toy circuit's G_R operations prepare smoothly weighted exponent registers. Here each modular multiplier M_4 is controlled by the least significant exponent bit: higher bits contribute identity because $4^2 = 1 \pmod{15}$. The four-qubit multiplier fixes the unused basis state 15, completing a reversible permutation. The grid marks the relation lattice in blue and the useful vector $(1, -1)$ in teal.

Interpret the improvement precisely

Regev's original 2023 result uses $\widetilde{O}(n^{3/2})$ gates per quantum run for an n-bit input, with $O(\sqrt{n})$ independent runs and polynomial classical processing. The tilde suppresses logarithmic factors. Multiplying the per-run cost by the number of runs gives a different total-work quantity. Reducing the size or depth of an individual coherent circuit can still be valuable when hardware reliability constrains each run.

The original analysis used a number-theoretic heuristic. Subsequent space-efficient work reduces the required qubit count, and Pilatte establishes unconditional correctness for slightly modified variants, preserving the improvement up to logarithmic factors. These are mathematical algorithm results. A practical RSA resource comparison additionally needs concrete arithmetic, error correction, connectivity, runtime and success targets.

Exercises

Draw the toy lattice and test its basis vectors freely. For $(1, 1)$, compute $u = 14 \equiv -1 \pmod{15}$: this relation is unhelpful. For $(1, -1)$, recover the useful value 11. The contrast shows why finding a relation and finding a factor are separate tasks.

1. Is $(2, 0)$ a relation, and is it useful?
2. What should a resource table separate?

Worked answers

1. yes: $4^2 = 1 \pmod{15}$. It gives $u = 2^2 = 4$, so $\gcd(3, 15) = 3$.
2. qubits, gates/depth per run, run count, classical work and physical implementation assumptions.

Check your understanding

PAUSE & THINK

What does the headline gate improvement describe?

- [ ] The size of a quantum circuit used in each run
- [ ] A completed RSA-2048 attack on existing hardware
- [ ] Every measured Fourier sample is already a factor

Answer: The size of a quantum circuit used in each run. Repetition and classical processing remain part of the full algorithm.

Regev's approach replaces one-dimensional period finding with multidimensional relation finding. The useful output emerges from quantum samples, classical geometry and modular arithmetic working together.

Sources

Regev, original algorithm; Ragavan and Vaikuntanathan, space-efficient variant; Pilatte, unconditional variants, revised August 2026. The $N = 15$ relation, grid and circuit figure above are original instructional calculations.