UNIVERSIDADE FEDERAL FLUMINENSE

FÁBIO GOMES DOS SANTOS

OTIMIZANDO O ALGORITMO QUÂNTICO DE FATORAÇÃO DO SHOR COM USO DE RETICULADO

NITERÓI

2026

FÁBIO GOMES DOS SANTOS

OTIMIZANDO O ALGORITMO QUÂNTICO DE FATORAÇÃO DO SHOR COM USO DE RETICULADO

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Doutor em Computação. Área de concentração: Ciência da Computação

NITERÓI

2026

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

S237o Santos, Fábio Gomes dos
OTIMIZANDO O ALGORITMO QUÂNTICO DE FATORAÇÃO DO SHOR COM USO DE RETICULADO / Fábio Gomes dos Santos. - 2026.
71 f.: il.

Orientador: Luis Antonio Brasil Kowada.
Tese (doutorado)-Universidade Federal Fluminense, Instituto de Computação, Niterói, 2026.

1. Criptografia. 2. Computador quântico. 3. Reticulado. 4. Algoritmo. 5. Produção intelectual. I. Kowada, Luis Antonio Brasil, orientador. II. Universidade Federal Fluminense. Instituto de Computação. III. Título.

CDD - XXX

Bibliotecário responsável: Debora do Nascimento - CRB7/6368

FÁBIO GOMES DOS SANTOS

OTIMIZANDO O ALGORITMO QUÂNTICO DE FATORAÇÃO DO SHOR COM USO DE RETICULADO

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Doutor em Computação. Área de concentração: Ciência da Computação

Aprovada em julho de 2026.

BANCA EXAMINADORA

Prof. Luis Antonio Kowada - Orientador, UFF

Profa. Celina Miraglia Herrera de Figueiredo, UFRJ

Prof. Daniel Chicayban Bastos, UFRJ

Prof. Franklin Lima Marquezino, UFRJ

Prof. Luís Felipe Ignácio Cunha, UFF

Niterói

2026

Ao meu filho Miguel, para que você cresça sabendo que nenhum sonho é grande demais e que a dedicação e resiliência podem transformar qualquer plano em realidade.

Agradecimentos

A jornada do doutorado é, por natureza, um caminho repleto de desafios e superações, que jamais poderia ser trilhado de forma solitária. Por isso, expresso aqui a minha mais profunda gratidão a todos que, direta ou indiretamente, contribuíram para a concretização deste trabalho.

A Deus, pela graça, pela saúde e por renovar minhas energias a cada amanhecer, permitindo-me concluir esta jornada.

Ao meu orientador, Prof. Dr. Luis Kowada, pela paciência, confiança e sabedoria. Obrigado por me guiar com excelência, por seus apontamentos sempre precisos e por ter sido uma verdadeira inspiração acadêmica e profissional ao longo de todos esses anos.

Aos meus colegas de pós graduação que em conjunto com nosso orientador me apoiaram na apresentação em congressos e na escrita de artigos.

Ao meu ex-chefe, Marcelo Contursi, pelo apoio fundamental durante essa trajetória. Agradeço pelas oportunidades profissionais que me concedeu, pelo incentivo constante aos meus estudos e por ter compreendido as exigências e os momentos de ausência que a pesquisa me demandou.

Aos meus pais, Ivan (*in memoriam*) e Ana Maria, que nunca mediram esforços para que eu pudesse alcançar os meus sonhos. Obrigado por me transmitirem os valores essenciais da vida, pelo incentivo contínuo aos meus estudos desde os primeiros anos e pelo amor infinito. Esta conquista também é de vocês.

À minha esposa, Ludimila. Obrigado por compreender as horas dedicadas ao estudo, por cuidar do nosso filho enquanto eu me dedicava a escrita dessa tese e por me encorajar nos momentos de dúvida. Seu amor, sua parceria e sua paciência foram indispensáveis para que eu chegasse até aqui. Obrigado por compartilhar comigo cada pequena vitória deste longo caminho.

Resumo

Os algoritmos de Shor representam um dos principais avanços da computação quântica ao demonstrar que a fatoração de inteiros e o cálculo de logaritmos discretos podem ser realizados em tempo polinomial, ameaçando a segurança de sistemas criptográficos amplamente utilizados, como RSA e ECDSA. Apesar de seu impacto teórico, a execução prática do algoritmo em larga escala ainda é inviável, principalmente devido às limitações do hardware quântico atual, incluindo ruído, necessidade de correção de erros e elevado custo em número de qubits. Além disso, a simulação clássica do algoritmo enfrenta desafios significativos relacionados ao consumo exponencial de memória e processamento.

Neste trabalho, investigamos melhorias no algoritmo de Shor com foco na redução dos requisitos de hardware, particularmente no número de qubits necessários. Propomos uma abordagem baseada em técnicas de reticulados no pós-processamento, permitindo a recuperação da ordem a partir de múltiplas execuções da versão original do algoritmo. Essa estratégia estabelece um *trade-off* entre a redução de recursos quânticos e o aumento no número de execuções.

Adicionalmente, desenvolvemos um simulador para resolver o problema da fatoração, possibilitando a análise do comportamento do algoritmo em diferentes cenários e a validação das otimizações propostas. Os resultados obtidos indicam que é possível reduzir significativamente os requisitos de hardware sem comprometer a corretude do algoritmo, contribuindo para aproximar sua implementação prática em dispositivos quânticos reais.

Palavras-chave: Criptografia, RSA, SHOR, Computação Quântica, Simulação.

Abstract

Shor's algorithm represents a major breakthrough in quantum computing by demonstrating that integer factorization and discrete logarithms can be solved in polynomial time, thereby threatening widely used cryptographic systems such as RSA and ECDSA. Despite its theoretical impact, the practical execution of the algorithm at scale remains infeasible, primarily due to limitations of current quantum hardware, including noise, the need for error correction, and the high cost in terms of qubit resources. Additionally, classical simulation of the algorithm faces significant challenges related to exponential memory and computational requirements.

In this work, we investigate improvements to Shor's algorithm with a focus on reducing hardware requirements, particularly the number of qubits. We propose an approach based on *lattice* techniques in the post-processing stage, enabling order recovery from multiple executions of the original version of the algorithm. This strategy introduces a trade-off between reduced quantum resources and an increased number of executions.

Furthermore, we develop a simulator for the integer factorization problem, allowing us to analyze the behavior of the algorithm under different scenarios and to validate the proposed optimizations. The results indicate that it is possible to significantly reduce hardware requirements without compromising correctness, contributing toward the practical realization of quantum factoring on real devices.

Keywords: Cryptography, RSA, SHOR, Quantum computing, Simulation.

Lista de Figuras

Lista de Tabelas

Sumário

1 Introdução

O algoritmo de Shor para a fatoração de inteiros foi um marco na ciência da computação. Ele demonstrou como um computador quântico poderia ser usado para fatorar inteiros em tempo polinomial. Desenvolvido pelo cientista da computação Peter Shor, este algoritmo revolucionou a criptografia ao demonstrar a capacidade que um computador quântico tem de resolver problemas que, para os computadores clássicos, seriam praticamente impossíveis em um tempo razoável. Em particular, ele aborda dois desafios matemáticos fundamentais: a fatoração de grandes números inteiros e o problema do logaritmo discreto. Ambos os problemas constituem o alicerce da segurança de diversos sistemas criptográficos fundamentais na atualidade, como o RSA (Rivest–Shamir–Adleman) e o ECDSA (Elliptic Curve Digital Signature Algorithm). Para dimensionar a relevância desses protocolos, basta observar que o RSA é o padrão utilizado pelo SSH (Secure Shell), essencial para o acesso e gerenciamento remoto de sistemas. Enquanto o ECDSA é a base de segurança do Bitcoin (Nakamoto, 2008) (a criptomoeda pioneira e de maior valor de mercado até esta data). Consequentemente, qualquer vulnerabilidade ou avanço que impacte a credibilidade desses métodos pode causar uma disrupção sem precedentes no funcionamento de instituições e na infraestrutura digital global. Desde a publicação de sua primeira versão em 1994, o algoritmo de Shor (Shor, 1999) tem sido objeto de diversos estudos que visam tanto elucidar seu funcionamento quanto propor otimizações. A literatura sobre o tema abrange desde abordagens estritamente teóricas, focadas em formulações matemáticas como em (Leander, 2002), até vertentes mais práticas. Estas últimas incluem simulações em arquiteturas clássicas, como exemplo os trabalhos do Ekerå (Ekerå, 2024) e (Ekerå, 2021), e implementações experimentais em hardware quântico real, como o trabalho pioneiro de (Vandersypen *et al.*, 2001).

Uma das maneiras mais utilizadas para entender o funcionamento do algoritmo de Shor e também para validar mudanças propostas é através de simulações. Muitas simulações do algoritmo de Shor já foram apresentadas (Bastos; Kowada, 2021; Wang; Hill; Hollenberg, 2017; Tankasala; Ilatikhameneh, 2019; Silva; Santos; Kowada, 2025). No en-

tanto, a grande maioria falha ao tentar simular o algoritmo com números inteiros grandes (Wang; Hill; Hollenberg, 2017; Tankasala; Ilatikhameneh, 2019). Isso acontece porque, ao simular o algoritmo, o hardware é sobrecarregado em termos de memória e processamento. É necessário alocar uma quantidade exorbitante de memória para armazenar o número de estados exigidos pelo algoritmo. Em uma simulação desse tipo, os estados são representados por diferentes posições de memória. O número de estados proposto por Shor (Shor, 1999) é uma potência de 2 entre N^2 e $2N^2$. Para fatorar um número de 2048 bits, que é o tamanho de chave RSA recomendado, alcançaremos 2^{4096} estados. Isso definitivamente não é suportado por computadores clássicos, pois a maioria dos computadores tem apenas 2^{64} endereços de memória. Mesmo para outras especificações RSA menos seguras, como RSA com 512 bits, seria impossível, pois esse cenário exigiria 2^{1024} estados. Dependendo de como é feita a simulação, pode-se ter também problema relacionado à capacidade de processamento. Algumas abordagens de simulação precisam saber a ordem de um grupo multiplicativo Z_N^*. No entanto, não é conhecido um algoritmo de tempo polinomial para recuperar a ordem do elemento em um grupo Z_N^*, por exemplo. Por outro lado, existem alguns trabalhos que não se deparam com esse problema porque o foco é uma parte mais do algoritmo, como por exemplo, avaliar o pós processamento, como vemos em (Bastos; Kowada, 2021). Neste trabalho, os autores propõem resolver esse problema considerando toda a rotina quântica como uma "caixa preta", com o objetivo de simular o algoritmo com números grandes.

Embora o algoritmo de Shor (Shor, 1999) tenha sido proposto em 1994, decorridas mais de três décadas, sua execução para números de grande magnitude permanece inviável em hardware real, uma limitação que decorre da dificuldade em construir dispositivos quânticos confiáveis e capazes de processar informações sem que o ruído, inerente ao meio físico, comprometa o resultado final. Esta inviabilidade prática está profundamente ligada à eficiência das operações lógicas reversíveis, um conceito fundamental para a computação quântica estabelecido por (Bennett, 1973), que demonstrou a possibilidade de realizar computações sem a dissipação inevitável de energia, desde que o sistema não descarte informações. No contexto de Shor, tal reversibilidade exige que operações aritméticas clássicas, como a exponenciação modular, sejam traduzidas em circuitos que preservem a coerência; entretanto, a implementação dessas rotinas em hardware limitado impõe um custo elevado em termos de qubits auxiliares e profundidade de circuito. Diferente dos sistemas clássicos, esse hardware quântico é extremamente suscetível a falhas (Barends *et al.*, 2014; Kim *et al.*, 2014; Schroeder; Pinheiro; Weber, 2011), tornando as técnicas de correção de erros não apenas essenciais, mas consideravelmente mais onerosas (Campbell;

Khurana; Montanaro, 2019; Fowler *et al.*, 2012). Para mitigar esses efeitos, múltiplos qubits físicos são tipicamente agrupados para formar um único qubit lógico, abordagem que amplia drasticamente a demanda por recursos devido à redundância significativa exigida. Nesse cenário, otimizar a estrutura lógica proposta por (Bennett, 1973) para minimizar a sobrecarga de memória e simplificar os circuitos torna-se um passo fundamental para reduzir o abismo entre a teoria algorítmica e as restrições físicas dos processadores atuais. Dessa forma, torna-se evidente que a evolução do algoritmo de Shor depende da combinação de avanços no hardware e melhorias algorítmicas que reduzam sua complexidade prática, sendo esta estratégia essencial para aproximar a computação quântica de aplicações reais na criptografia, onde a fatoração de grandes inteiros representa tanto uma ameaça quanto uma oportunidade para o desenvolvimento de novos sistemas pós-quânticos.

Nesta tese, investigamos possíveis melhorias no algoritmo de fatoração de Shor, explorando otimizações que visam acelerar sua execução e aumentar sua eficiência. Nosso principal objetivo é avaliar em que medida é possível reduzir os requisitos de hardware, mais especificamente o número de qubits, e ainda assim fatorar um inteiro arbitrário N. Para isso, propomos o uso de técnicas baseadas em reticulados no pós-processamento, de modo que, a partir dos resultados de múltiplas execuções (n) da proposta original do algoritmo de Shor (Shor, 1999), seja possível extrair a ordem associada aos valores retornados pelo QOFA (Quantum Order Finding Algorithm). A abordagem consiste em utilizar menos qubits, assumindo como *trade-off* a necessidade de um maior número de execuções do algoritmo. Além disso, desenvolvemos um simulador para o problema de fatoração, o que nos permite visualizar e analisar o comportamento do algoritmo em diferentes cenários, bem como validar a eficácia das melhorias propostas. Ressalta-se, entretanto, que o objetivo desta tese não consiste em propor um algoritmo universalmente superior ao algoritmo original de Shor para a fatoração de inteiros arbitrários. O foco deste trabalho está na demonstração, associado com investigação experimental, do impacto de técnicas de pós-processamento baseadas em reticulados sobre a redução dos requisitos quânticos, particularmente no número de qubits necessários para a recuperação da ordem em simulações do QOFA.

Este trabalho inicia-se com uma revisão sucinta dos conceitos fundamentais necessários à compreensão do tema (capítulo 2), seguida por uma análise dos trabalhos relacionados e de como estes se conectam à nossa proposta (capítulo 3). Em seguida, descrevemos a proposta de otimização e detalhamos o simulador desenvolvido (capítulo 4) e as execuções realizadas (capítulo 5). Por fim, apresentamos e discutimos os resultados obtidos por meio das simulações, comparando-os com o estado da arte (capítulo 6).

2 Fundamentação Teórica

Nesta seção, nós apresentamos os conceitos fundamentais de computação quântica, computação clássica e teoria dos números que estão de alguma maneira relacionados a este trabalho, fornecendo assim ao leitor as informações necessárias para sua plena compreensão.

2.1 Conceitos Fundamentais de Computação Quântica

A unidade básica de informação na computação quântica é o qubit. Diferente do bit clássico, um qubit $|\psi\rangle$ pode existir em uma superposição de estados, sendo representado matematicamente como um vetor em um espaço de Hilbert bidimensional:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{2.1}$$

onde $\alpha, \beta \in \mathbb{C}$ são as amplitudes dos estados. A relação entre amplitude e probabilidade é definida pelo postulado da medição, onde a probabilidade de colapsar o estado para $|0\rangle$ ou $|1\rangle$ é dada por $|\alpha|^2$ e $|\beta|^2$, respectivamente, satisfazendo a condição de normalização $|\alpha|^2 + |\beta|^2 = 1$ (Nielsen; Chuang, 2010, Cap. 1).

Além da **superposição**, que permite ao sistema processar múltiplas combinações de estados simultaneamente, destaca-se o **emaranhamento**. Este fenômeno descreve correlações não-locais entre qubits, onde o estado quântico de uma partícula não pode ser descrito independentemente do estado da outra, mesmo quando separadas por grandes distâncias. Esse comportamento é essencial para o funcionamento de algoritmos quânticos (Nielsen; Chuang, 2010, Cap. 1) .

Outro conceito fundamental é a **evolução unitária** dos estados quânticos. De acordo com os postulados da mecânica quântica, a evolução de um sistema fechado é descrita por operadores unitários U, ou seja, operadores tais que $U^\dagger U = I$, garantindo a reversibilidade das operações (Nielsen; Chuang, 2010, Cap. 1).

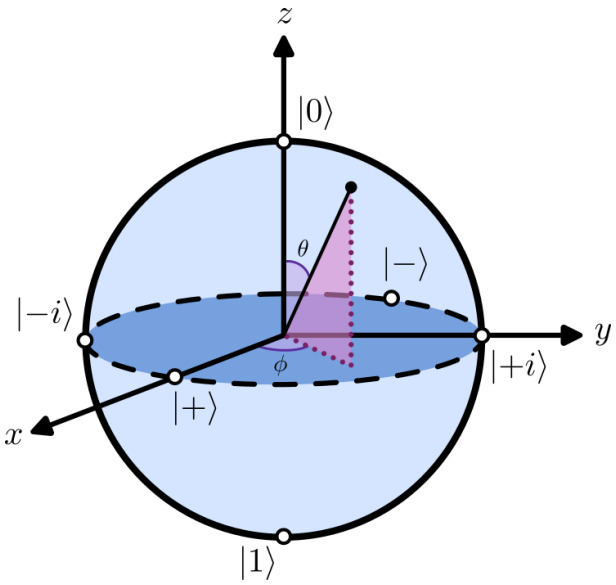

Figura 1: Representação de um qubit

As operações sobre qubits são implementadas por meio de portas quânticas, que correspondem a essas transformações unitárias. As portas de um único qubit são representadas por matrizes 2×2, dentre as quais destacam-se as portas de Pauli:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.2}$$

A porta X atua como um operador de inversão de bits, enquanto as portas Y e Z introduzem mudanças de fase no estado quântico (Nielsen; Chuang, 2010, Cap. 1). Outra porta fundamental é a porta de Hadamard:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{2.3}$$

que permite criar superposições a partir de estados base, sendo amplamente utilizada na inicialização de algoritmos quânticos (Nielsen; Chuang, 2010, Cap. 1).

Para sistemas compostos, utiliza-se o produto tensorial. Um sistema com n qubits é descrito em um espaço de dimensão 2^n, permitindo representar uma quantidade exponencial de estados com apenas n qubits físicos (Nielsen; Chuang, 2010, Cap. 1).

As **portas de múltiplos qubits** são essenciais para explorar o emaranhamento. A porta CNOT (*Controlled-NOT*) atua sobre dois qubits e inverte o qubit alvo quando o qubit de controle está no estado $|1\rangle$ (Nielsen; Chuang, 2010, Cap. 1). Sua representação

matricial é:

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{2.4}$$

A combinação de portas de um e dois qubits permite construir **circuitos quânticos**, que são sequências ordenadas de operações aplicadas aos qubits. Esses circuitos constituem o modelo computacional padrão da computação quântica (Nielsen; Chuang, 2010, Cap. 1). Cada linha horizontal representa um qubit, e as portas são aplicadas ao longo do tempo, culminando em uma etapa final de medição. Um resultado fundamental é que

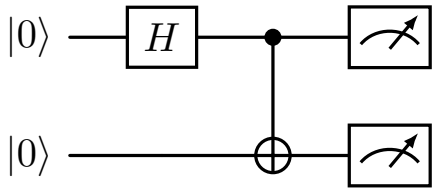

Figura 2: Circuito para geração de um estado de Bell $|\Phi^+\rangle$.

o conjunto formado por portas de um qubit juntamente com a porta CNOT é **universal**, permitindo aproximar qualquer transformação unitária com precisão arbitrária (Nielsen; Chuang, 2010, Cap. 1).

A medição de estados quânticos introduz um comportamento probabilístico, causando o colapso da superposição para um dos estados da base computacional. Esse processo impõe desafios no desenvolvimento de algoritmos, pois a informação quântica pode ser irreversivelmente perdida (Nielsen; Chuang, 2010, Cap. 1).

Outro princípio importante é o **teorema da não-clonagem**, que estabelece a impossibilidade de copiar um estado quântico arbitrário desconhecido, restringindo a manipulação da informação quântica (Nielsen; Chuang, 2010, Cap. 12).

Além disso, a computação quântica explora a **interferência quântica**, permitindo amplificar amplitudes de estados desejados e cancelar estados indesejados, sendo esse mecanismo central para o ganho de desempenho em algoritmos quânticos.

Por fim, sistemas quânticos reais estão sujeitos à **decoerência** e ao ruído, o que pode degradar rapidamente a informação armazenada. Como consequência, a profundidade dos

circuitos e o número de portas tornam-se fatores críticos, motivando o desenvolvimento de técnicas de correção de erros e otimização de circuitos (Nielsen; Chuang, 2010, Cap. 7).

2.2 Grupos Matemáticos e o Grupo $\mathbb{Z}_N^*$

Formalmente, um **grupo** é um par $(G, *)$, onde G é um conjunto não vazio e $* : G \times G \to G$ é uma operação binária que satisfaz os seguintes axiomas:

- **Fechamento:** para todos $a, b \in G$, tem-se $a * b \in G$;
- **Associatividade:** para todos $a, b, c \in G$, vale $(a * b) * c = a * (b * c)$;
- **Elemento neutro:** existe $e \in G$ tal que $e * a = a * e = a$ para todo $a \in G$;
- **Elemento inverso:** para todo $a \in G$, existe $a^{-1} \in G$ tal que $a * a^{-1} = a^{-1} * a = e$.

Quando a operação também é comutativa, isto é, $a * b = b * a$ para todos $a, b \in G$, o grupo é dito **abeliano**. A quantidade de elementos de G é denominada **ordem do grupo**, denotada por $|G|$. Um subconjunto $H \subseteq G$ que é, ele próprio, um grupo sob a mesma operação é chamado de **subgrupo** de G. Um resultado central nesse contexto é o **Teorema de Lagrange**, que estabelece que a ordem de qualquer subgrupo H divide a ordem do grupo G, isto é, $|H|$ divide $|G|$.

O fundamento algébrico do algoritmo de Shor reside na teoria de grupos, não somente ao grupo multiplicativo de inteiros módulo N, denotado por $\mathbb{Z}_N^*$, mas este é o grupo de interesse para este trabalho. Este grupo consiste no conjunto de inteiros $\{a \in \mathbb{Z} : 1 \leq a < N \text{ e } \text{mdc}(a, N) = 1\}$, sob a operação de multiplicação modular.

As propriedades fundamentais deste grupo para a fatoração são:

- **Estrutura Cíclica:** Para qualquer elemento $a \in \mathbb{Z}_N^*$, o conjunto de potências $\{a^0, a^1, a^2, \ldots \pmod N\}$ forma um subgrupo cíclico dos subgrupos gerados por a em $\mathbb{Z}_N^*$.
- **Ordem do Elemento:** A ordem r é o menor inteiro positivo tal que $a^r \equiv 1 \pmod N$. O algoritmo de Shor é eficiente justamente por transformar a busca exaustiva por esse valor em um problema de estimativa de fase quântica.

A estrutura de $\mathbb{Z}_N^*$ determina a distribuição dos picos de probabilidade na simulação quântica. Se o grupo possui uma estrutura muito regular ou degenerada, o pós-processamento via frações continuadas pode ser facilitado ou dificultado, dependendo da cardinalidade do grupo, dada pela função totiente de Euler $\phi(N)$.

2.3 Safe Primes e sua Relevância Criptográfica

Na seleção dos parâmetros para sistemas como o RSA, o uso de *safe primes* (primos seguros) é uma prática comum para aumentar a resistência contra algoritmos clássicos de fatoração, como é o caso do Método de Pollard $p-1$. Um número primo p é considerado *safe prime* se for da forma:

$$p = 2u + 1, \tag{2.5}$$

onde u também é um número primo (denominado primo de Sophie Germain).

No contexto da simulação do algoritmo de Shor, a escolha de $N = p \cdot q$ onde p e q são *safe primes* tem um impacto direto. A utilização de *safe primes* introduz uma propriedade algébrica fundamental que otimiza a obtenção do valor da ordem usado na execução da simulação. Pelo Teorema de Lagrange, a ordem r de qualquer elemento $x \in \mathbb{Z}_N^*$ deve ser um divisor da ordem do grupo, dada pela função totiente de Euler $\phi(N)$.

No caso específico de $N = p \cdot q$, onde p e q são primos seguros, os divisores de $p-1$ e $q-1$ são restritos aos conjuntos $\{1, 2, p', 2p'\}$ e $\{1, 2, q', 2q'\}$, respectivamente. Como a ordem global r em $\mathbb{Z}_N^*$ é definida pelo mínimo múltiplo comum das ordens locais, $r = \text{lcm}(r_p, r_q)$, o espaço de busca para o valor correto de r é drasticamente reduzido ao conjunto:

$$\mathcal{R}_{candidatos} = \{1, 2, p', q', 2p', 2q', p'q', 2p'q'\}. \tag{2.6}$$

Esta característica permite uma simulação com números grandes dado que em vez de uma busca exaustiva, o simulador pode validar se o resultado do pós processamento para o pico de probabilidade recuperado no QOFA está de acordo com o esperado. Simular a fatoração de produtos de *safe primes* permite avaliar o desempenho do algoritmo em instâncias que são classicamente difíceis, garantindo que as melhorias propostas mantenham a eficácia em cenários de alta complexidade.

2.4 Fast Fourier Transform

A Transformada Discreta de Fourier (DFT) é uma das ferramentas matemáticas mais importantes para análise de sinais, permitindo decompor uma sequência discreta em suas componentes de frequência. Dada uma sequência $x_0, x_1, \ldots, x_{N-1}$, sua DFT é definida por

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i k n/N}, \quad k = 0, \ldots, N-1. \tag{2.7}$$

Essa transformação converte uma representação no domínio do tempo (ou espaço) para uma representação no domínio da frequência. A DFT possui papel central em diversas áreas da matemática aplicada, incluindo análise numérica, teoria dos números computacional e processamento digital de sinais.

O cálculo direto da DFT possui complexidade $O(N^2)$. Entretanto, algoritmos conhecidos como Fast Fourier Transform (FFT) permitem calcular a mesma transformação em tempo $O(N \log N)$. O algoritmo mais conhecido foi introduzido por (Cooley; Tukey, 1965), e baseia-se na decomposição recursiva da DFT em transformadas menores, explorando propriedades algébricas das raízes da unidade.

A FFT tornou-se uma das ferramentas computacionais mais importantes da engenharia moderna. Entre suas principais aplicações estão, por exemplo: processamento digital de sinais, multiplicação rápida de inteiros e polinômios, e etc.

Em ciência da computação e matemática computacional, ela também desempenha papel fundamental em algoritmos rápidos para multiplicação de polinômios e em métodos de convolução.

Do ponto de vista conceitual, a FFT pode ser entendida como um algoritmo eficiente para computar a transformada associada à matriz de Fourier discreta. Essa estrutura algébrica tem analogias diretas com a Transformada de Fourier Quântica (QFT), utilizada em diversos algoritmos quânticos, incluindo o algoritmo de fatoração de Shor. Enquanto a FFT fornece uma forma eficiente de computar transformadas de Fourier em computadores clássicos, a QFT explora propriedades de interferência quântica para realizar transformações análogas sobre amplitudes de estados quânticos.

2.5 Transformada de Fourier Quântica (QFT)

A Transformada de Fourier Quântica (QFT) é a análoga quântica da Transformada Discreta de Fourier (DFT) e constitui o núcleo de diversos algoritmos de aceleração exponencial. Sua função principal é realizar uma transformação de base que mapeia um estado quântico $|x\rangle$ em uma superposição de estados de fase, conforme a operação:

$$QFT|x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i x y/N} |y\rangle. \tag{2.8}$$

As características fundamentais da QFT que a tornam indispensável são:

- **Eficiência Computacional:** Enquanto a FFT clássica opera em $O(N \log N)$, a QFT pode ser implementada com complexidade de circuitos de $O(n^2)$, onde $n = \log N$ é o número de qubits.
- **Extração de Periodicidade:** A QFT possui a propriedade de converter padrões de interferência em picos de probabilidade bem definidos. Se um estado de entrada possui uma periodicidade r, a aplicação da QFT concentra a amplitude de probabilidade em múltiplos de $1/r$.

Essa capacidade de isolar frequências específicas é a ferramenta matemática que viabiliza a sub-rotina de *order finding*. Ao transformar a superposição periódica gerada pela exponenciação modular, a QFT permite que o período r, informação essencial para a fatoração de grandes inteiros, seja medido com alta probabilidade no registrador de saída.

2.6 Algoritmo de Shor e Fatoração de Inteiros

O algoritmo de Shor resolve o problema da fatoração de um número inteiro composto N ao reduzi-lo ao problema de **encontrar a ordem** ($order\ finding$) em um grupo multiplicativo $\mathbb{Z}_N^*$.

2.6.1 Redução do Problema

A parte clássica da redução consiste em escolher um número aleatório $a < N$ tal que $mdc(a, N) = 1$. O objetivo passa a ser encontrar o menor inteiro positivo r (o período ou

ordem) que satisfaça a relação:

$$a^r \equiv 1 \pmod{N}. \tag{2.9}$$

Uma vez encontrado um r par, os fatores de N podem ser obtidos via $mdc(a^{r/2} \pm 1, N)$.

2.6.2 Fases do Algoritmo

O algoritmo é tipicamente dividido em duas fases principais:

- **Fase Quântica (Sub-rotina de Ordem):** Utiliza a Transformada de Fourier Quântica (QFT) e a Estimativa de Fase Quântica (QPE) para encontrar o valor de r. Esta etapa opera sobre um registrador de entrada parametrizado por q qubits, onde $q \approx 2\log_2 N$, garantindo precisão suficiente para a distinção dos picos de probabilidade.
- **Fase Clássica (Pós-processamento):** Conforme discutido anteriormente, esta fase utiliza os dados medidos do registrador quântico para extrair o valor exato de r através de métodos como frações continuadas ou algoritmos de reticulado.

2.7 Reticulados

Reticulados constituem a estrutura matemática central do pós-processamento proposto neste trabalho. Formalmente, dado um conjunto de vetores linearmente independentes $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d \in \mathbb{R}^n$, o **reticulado** gerado por eles é o conjunto de todas as combinações lineares inteiras desses vetores:

$$\Lambda = \left\{ \sum_{i=1}^{d} \lambda_i \mathbf{b}_i \ : \ \lambda_i \in \mathbb{Z} \right\}. \tag{2.10}$$

Os vetores $\mathbf{b}_1, \ldots, \mathbf{b}_d$ formam uma **base** do reticulado, e d é a sua **dimensão**. Quando organizados como as linhas de uma matriz $B \in \mathbb{R}^{d \times n}$, escreve-se $\Lambda = \Lambda(B)$. Diferentemente de um espaço vetorial, no qual os coeficientes são reais, em um reticulado os coeficientes são restritos aos inteiros, o que confere à estrutura um caráter discreto.

Embora a base não seja única, o valor absoluto do determinante permanece invariante e define o **determinante do reticulado**:

$$\det(\Lambda) = \sqrt{\det(BB^\top)}, \tag{2.11}$$

que, quando $d = n$, reduz-se a $|\det(B)|$. Geometricamente, $\det(\Lambda)$ corresponde ao volume do paralelepípedo fundamental gerado pelos vetores da base e é independente da base escolhida.

Uma quantidade fundamental associada a um reticulado é o seu **primeiro mínimo**, definido como a norma euclidiana do menor vetor não nulo do reticulado:

$$\lambda_1(\Lambda) = \min_{\mathbf{v} \in \Lambda \setminus \{\mathbf{0}\}} \|\mathbf{v}\|. \tag{2.12}$$

Encontrar tal vetor é o objeto do **Problema do Vetor Mais Curto** (*Shortest Vector Problem*, SVP), que é computacionalmente difícil em dimensões arbitrárias. Um limitante clássico sobre λ_1 é fornecido pelo **Teorema de Minkowski**, que garante a existência de um vetor não nulo cuja norma satisfaz

$$\lambda_1(\Lambda) \leq \sqrt{d}\, \det(\Lambda)^{1/d}. \tag{2.13}$$

Esse limitante é essencial para a análise desenvolvida no Capítulo 4, pois estabelece a norma típica esperada dos vetores de um reticulado em função de seu determinante e dimensão.

Embora o SVP seja difícil no caso geral, o algoritmo **LLL** (Lenstra–Lenstra–Lovász) (Lenstra; Lenstra; Lovász, 1982) resolve uma versão aproximada do problema em tempo polinomial. A partir de uma base qualquer, o LLL produz uma base **reduzida**, cujos vetores são curtos e aproximadamente ortogonais. O primeiro vetor dessa base reduzida satisfaz

$$\|\mathbf{b}_1\| \leq 2^{(d-1)/4}\, \lambda_1(\Lambda), \tag{2.14}$$

ou seja, aproxima o vetor mais curto por um fator que depende apenas da dimensão d. Essa garantia de aproximação, combinada com a eficiência polinomial do algoritmo, é o que viabiliza o uso de reticulados como ferramenta de pós-processamento clássico na recuperação da ordem, conforme detalhado no Capítulo 4.

3 Trabalhos relacionados

O desenvolvimento e aperfeiçoamento de algoritmos de fatoração quântica tem se concentrado em duas áreas principais: aumentar a probabilidade de sucesso de uma única execução da rotina quântica para reduzir o tempo total de processamento ou em minimizar os requisitos de recursos físicos (qubits e complexidade de circuito) do hardware quântico.

3.1 Melhoria da Probabilidade de Sucesso e Eficiência de Execução

Um gargalo conhecido na implementação padrão do Algoritmo de Fatoração de Shor é a necessidade de execuções repetidas, devido a uma taxa de falha que pode chegar a 50% na fase de pós-processamento clássico. A estratégia empregada tem caráter probabilístico, apresentando uma probabilidade de sucesso de, no mínimo $1 - 1/2^{k-1}$, onde k é o número de fatores primos distintos que dividem o número composto que se deseja fatorar (Shor, 1999). Dessa forma, a probabilidade de sucesso mínima alcançada pela proposta original é de aproximadamente $1/2$.

O trabalho (Leander, 2002) abordou este problema introduzindo uma etapa de pré-processamento clássico que melhora significativamente o limite inferior de sucesso. Ao selecionar entradas específicas y pertencente a $\mathbb{Z}_N^*$, utilizando o símbolo de Jacobi $(y/n) = -1$, o autor demonstra que a probabilidade de obter uma saída utilizável aumenta de $1/2$ para $3/4$. Esta abordagem reduz eficientemente a probabilidade de falha no pior caso para $1/4$, diminuindo consequentemente o número esperado de execuções quânticas necessárias para fatorar um determinado número inteiro e, por extensão, o tempo total de execução.

Um avanço complementar foi apresentado por (Bastos; Kowada, 2021), que propõem uma estratégia para determinar, sem executar o algoritmo quântico de busca de ordem, se o algoritmo de Shor revelaria ou não um fator não trivial de N para uma dada base

x, assumindo que a fatoração de N é conhecida. O resultado central do trabalho estabelece condições suficientes sobre as ordens locais de x módulo cada fator primo de N para garantir que existe um expoente c tal que $1 < \text{mdc}(x^c - 1, N) < N$. Notavelmente, a abordagem generaliza o algoritmo original de Shor ao considerar qualquer divisor d da ordem r, e não apenas o caso $d = 2$, o que permite explorar ordens ímpares sem descartá-las. Como consequência prática, os autores produzem a uma evidência experimental direta da taxa de sucesso do método: para compostos de até 4096 bits formados por dois primos aleatórios de tamanhos semelhantes, o algoritmo necessita em média de aproximadamente 1,5 invocações da rotina quântica de busca de ordem, com pior caso não superior a 11 tentativas. Esse resultado fornece uma base empírica sólida para avaliar variantes e refinamentos do algoritmo de Shor.

Já (Grosshans *et al.*, 2015) otimiza ainda mais a execução para a classe específica de semiprimos seguros (*safe primes*) $N = p1 \cdot p2$, onde $p1$ e $p2$ são *safe primes*. O fato de *safe primes* aumentarem a probabilidade de sucesso em algoritmos quânticos contrasta com a abordagem clássica onde os *safe primes* são números mais difíceis de fatorar usando o algoritmo de Pollard (Pollard, 1974). É por esse fato inclusive que essa classe de números primos ganharam o prefixo "safe". Usando *safe primes*, a abordagem de (Grosshans *et al.*, 2015) alcança uma probabilidade de sucesso de aproximadamente $1 - 4/N$ com apenas uma única chamada ao Quantum Order Finding Algorithm (QOFA). Além de reduzir o número de consultas, o algoritmo oferece vantagens práticas: permite o uso de uma entrada constante (ex: $a = 2$) e utiliza um circuito reutilizável. Ao contrário do algoritmo de Shor, que exige um novo circuito ou reinicialização para diferentes entradas aleatórias, este método permite que a mesma configuração física seja executada novamente sem modificação, reduzindo drasticamente o overhead de tempo em execuções experimentais.

Outro trabalho importante é (Johnston, 2017) . Nele a autora fornece uma estratégia nova de como aproveitar as ordens não pares retornadas pelo QOFA. A ideia é usar os fatores primos da ordem r para tentar fatorar N mesmo que a ordem não seja par, o que é uma premissa no trabalho original do Shor. Dessa forma, nós temos um ganho direto na probabilidade de sucesso na execução do algoritmo de Shor e com isso evitando ter de executá-lo mais de uma vez.

3.2 Redução dos Requisitos de Qubits e Complexidade de Circuito

O maior obstáculo para o hardware quântico de curto prazo é a complexidade da exponenciação modular, que requer registros amplos e circuitos extensos. (Ekerå; Håstad, 2017) apresentaram uma generalização que reformula a fatoração de inteiros RSA como um problema de "logaritmo discreto curto". Esta mudança permite uma redução direta no tamanho do computador quântico necessário. Numa implementação padrão de Shor para um inteiro de n bits, a exponenciação é realizada com um expoente de comprimento $2n$ bits. Em contraste, o algoritmo de (Ekerå; Håstad, 2017) reduz este comprimento de expoente para pouco mais de $n/2$ bits (especificamente $(1/2 + 1/s)n$ bits), onde s é um parâmetro ajustável). A redução no comprimento do expoente traduz-se diretamente num menor número de qubits nos registos de índice (utilizando registos de comprimento $l+m$ e l qubits). Isto torna o algoritmo significativamente menos exigente em termos de memória quântica e mais viável para execução em arquiteturas atuais. Este quadro permite um *trade-off* flexível: ao aumentar ligeiramente o número de execuções s, os investigadores podem reduzir ainda mais as exigências impostas ao hardware quântico (menos qubits).

O trabalho de (Regev, 2025) propõe um algoritmo de fatoração quântica que reduz a complexidade do circuito para $\tilde{O}(n^{3/2})$ portas para um inteiro de n bits. Esta é uma melhoria significativa em relação ao algoritmo de Shor, que requer $\tilde{O}(n^2)$ portas. O autor alcança essa redução ao tratar a fatoração como um problema multidimensional, executando independentemente um circuito quântico menor por $\sqrt{n} + 4$ vezes. Em vez de operar com grandes exponenciações modulares únicas, o algoritmo utiliza pequenos inteiros de $O(\log n)$ bits como bases (b_i), o que diminui a carga computacional direta no hardware quântico. O diferencial fundamental da parte quântica do algoritmo de Regev é a substituição de uma única e profunda exponenciação modular por múltiplas execuções independentes de um circuito mais “raso” e eficiente, que utiliza aproximadamente $\sqrt{n}$ bases pequenas em vez de uma única base grande. Enquanto o algoritmo de Shor busca encontrar um período escalar em uma única dimensão, o circuito de Regev opera de forma multidimensional para gerar amostras ruidosas de um reticulado, transferindo a complexidade de reconstruir a estrutura do fator para o pós-processamento clássico. Essa abordagem reduz o número de portas lógicas de $\tilde{O}(n^2)$ para $\tilde{O}(n^{3/2})$, pois as operações quânticas passam a lidar com expoentes significativamente menores e cálculos modulares distribuídos, tornando o processo mais rápido e menos exigente em termos de profundidade de coerência para o hardware quântico.

4 Nossa Proposta

A proposta investigada neste trabalho baseia-se em um *trade-off* entre recursos quânticos e pós-processamento clássico. Em particular, busca-se reduzir os requisitos de hardware quântico, especialmente o número de qubits necessários para a recuperação da ordem, ao custo de um tratamento clássico mais sofisticado baseado em redução de reticulados. Embora essa estratégia introduza um custo computacional clássico adicional, tal custo permanece polinomial e significativamente menos restritivo que o aumento equivalente de recursos quânticos em arquiteturas reais. Em cenários tolerante a falhas, a ampliação do número de qubits e da profundidade dos circuitos representa um dos principais gargalos físicos e tecnológicos da computação quântica. Dessa forma, a substituição parcial de precisão quântica por pós-processamento clássico constitui um *trade-off* potencialmente vantajoso em implementações práticas do algoritmo de Shor. A ideia de usar reticulado é compensar a perda de precisão na medição decorrente da diminuição de qubits utilizados no registrador de entrada. Esta escolha justifica-se pela capacidade dessa abordagem em recuperar o período r mesmo a partir de amostras ruidosas ou parciais, algo que o método tradicional de frações continuadas falha em realizar quando a precisão dos bits de fase é reduzida.

4.1 Análise da Redução de Qubits

Um dos principais objetivos desta pesquisa consiste em investigar até que ponto técnicas de pós-processamento baseadas em reticulados podem compensar a redução da precisão quântica necessária para a recuperação da ordem no algoritmo de Shor. Diferentemente da abordagem tradicional, que depende fortemente da precisão da estimativa de fase para que a recuperação da ordem seja realizada por meio de frações continuadas, a abordagem proposta explora a utilização de múltiplas medições combinadas com técnicas de redução de bases de reticulados, permitindo extrair a ordem mesmo quando a quantidade de informação fornecida por cada medição individual é menor.

Na formulação tradicional do algoritmo de Shor, o registrador de fase possui tamanho l escolhido de forma que

$$Q = 2^l > N^2,$$

onde Q é o número de estados e N é o inteiro a ser fatorado. Como consequência, para um inteiro de n bits, com $n = \lceil \log_2 N \rceil$, obtém-se aproximadamente

$$l \approx 2n.$$

Essa condição garante que a aproximação racional obtida após a Transformada de Fourier Quântica possua precisão suficiente para que a ordem seja recuperada utilizando o método de frações continuadas.

Entretanto, essa exigência representa um custo significativo em termos de recursos quânticos. Em arquiteturas tolerantes a falhas, o número de qubits constitui um dos principais fatores limitantes para a execução de algoritmos quânticos em larga escala. Além disso, o aumento do tamanho do registrador implica circuitos mais extensos, maior profundidade lógica e maior sobrecarga associada aos mecanismos de correção de erros.

A proposta investigada neste trabalho busca relaxar essa restrição, reduzindo o número de qubits utilizados no registrador de fase e transferindo parte do esforço computacional para o estágio clássico de pós-processamento. Para avaliar quantitativamente esse *trade-off*, foram realizados experimentos utilizando registradores progressivamente menores e comparando-se a taxa de sucesso da recuperação da ordem.

Os resultados experimentais indicaram que a recuperação da ordem permanece viável mesmo quando o tamanho do registrador é reduzido para aproximadamente 52% do valor tradicionalmente empregado. Em termos formais, se denotarmos por l' o tamanho do registrador utilizado na proposta, podemos escrever

$$l' = \alpha l,$$

onde α representa a fração de qubits preservada em relação ao algoritmo tradicional.

Os experimentos realizados neste trabalho sugerem

$$\alpha \approx 0.52.$$

Substituindo a relação clássica $l \approx 2n$, obtém-se

$$l' \approx 0.52 \cdot 2n,$$

ou seja,

$$l' \approx 1.04n.$$

Isso significa que o registrador de fase passa a exigir aproximadamente $1.04n$ qubits, em contraste com os $2n$ qubits requeridos pela abordagem tradicional baseada em frações continuadas.

A redução relativa é definida por

$$\text{Redução} = \frac{l - l'}{l},$$

resultando em

$$\text{Redução} = \frac{2n - 1.04n}{2n} = 0.48.$$

Portanto, os resultados observados sugerem uma economia aproximada de 48% no número de qubits do registrador de fase.

A Tabela 1 apresenta exemplos concretos dessa redução para os tamanhos de entrada utilizados nas simulações.

Tabela 1: Estimativa da redução do tamanho do registrador de fase.

Bits de N	Shor Tradicional	Proposta	Redução
128	256	133	48,0%
256	512	266	48,0%
512	1024	532	48,0%

É importante ressaltar que essa redução não implica uma diminuição da complexidade assintótica do algoritmo de Shor. O procedimento continua dependendo da execução do QOFA e preserva a natureza polinomial da fatoração quântica. O ganho obtido está relacionado à diminuição dos requisitos de hardware necessários para alcançar uma taxa de sucesso comparável, compensando a perda de precisão individual das medições por

meio de um pós-processamento clássico mais sofisticado.

Sob essa perspectiva, os resultados obtidos devem ser interpretados como evidência de um *trade-off* entre recursos quânticos e processamento clássico. Enquanto a abordagem tradicional privilegia a precisão das medições por meio de registradores maiores, a proposta apresentada explora a combinação de múltiplas medições com técnicas de redução de reticulados para recuperar a mesma informação utilizando menos qubits. Em cenários onde a disponibilidade de qubits constitui o principal fator limitante, tal estratégia pode representar uma alternativa prática e potencialmente vantajosa para futuras implementações do algoritmo de Shor.

4.2 Recuperação da Ordem via Reticulados

Após a medição na Transformada de Fourier Quântica no algoritmo de Shor (Shor, 1999), obtém-se um valor inteiro c tal que

$$\left|\frac{c}{Q} - \frac{k}{r}\right| \leq \frac{1}{2Q} \tag{4.1}$$

onde $Q = 2^l$, r é a ordem procurada e $k \in \{0, \ldots, r-1\}$. O problema clássico consiste em recuperar r a partir da aproximação racional c/Q.

4.2.1 Caso base: uma medição (Frações Continuadas).

Tradicionalmente, aplica-se o método de frações continuadas para encontrar convergentes p/q que satisfaçam

$$\left|\frac{c}{Q} - \frac{p}{q}\right| < \frac{1}{2q^2}. \tag{4.2}$$

Sob essa condição, garante-se $q = r$, desde que $\gcd(k,r) = 1$ (Nielsen; Chuang, 2010). Entretanto, essa abordagem apresenta limitações estruturais: requer precisão suficiente na aproximação, trata apenas uma única relação linear e falha quando k e r não são coprimos.

Para garantir a precisão necessária, o algoritmo original de Shor exige que o registrador de fase possua tamanho l tal que $Q = 2^l > N^2$. Como consequência, para um inteiro de n bits, tem-se

$$l \approx 2n \text{ qubits.} \tag{4.3}$$

4.2.2 Generalização: m medições via Reticulado.

Consideremos agora o cenário em que o QOFA é executado m vezes, produzindo as medições $c_1, c_2, \ldots, c_m$. Cada valor c_i satisfaz a relação (4.1), de modo que existem inteiros $k_i \in \{0, \ldots, r-1\}$ tais que

$$|r\,c_i - k_i\,Q| \;\leq\; \frac{r}{2}, \qquad i = 1, \ldots, m. \tag{4.4}$$

O problema clássico ampliado consiste em *recuperar r a partir do conjunto $\{c_1, \ldots, c_m\}$ e de Q, sem conhecer os k_i.*

[Reticulado de recuperação de ordem] Sejam $c_1, \ldots, c_m \in \mathbb{Z}$ medições obtidas pelo QOFA com registrador de tamanho Q. Define-se o *reticulado de recuperação de ordem* como o reticulado $\Lambda(\mathbf{B}_m)$ gerado pelas linhas da matriz $(m+1) \times (m+1)$:

$$\mathbf{B}_m = \begin{pmatrix} 1 & c_1 & c_2 & \cdots & c_m \\ 0 & Q & 0 & \cdots & 0 \\ 0 & 0 & Q & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & Q \end{pmatrix} \in \mathbb{Z}^{(m+1)\times(m+1)}. \tag{4.5}$$

Para $m = 1$, a base $\mathbf{B}_1$ coincide com a construção bidimensional clássica apresentada em trabalhos anteriores sobre pós-processamento via reticulado para o algoritmo de Shor (Ekerå, 2021):

$$\mathbf{B}_1 = \begin{pmatrix} Q & 0 \\ c & 1 \end{pmatrix}.$$

A formulação $(m+1)$-dimensional generaliza esse caso para múltiplas medições.

O vetor alvo e sua norma. Considere o vetor de coeficientes inteiros $\boldsymbol{\lambda}^* = (r, -k_1, -k_2, \ldots, -k_m)$. O ponto correspondente no reticulado é:

$$\mathbf{w}^* = \boldsymbol{\lambda}^* \cdot \mathbf{B}_m = \big(r,\; r\,c_1 - k_1\,Q,\; r\,c_2 - k_2\,Q,\; \ldots,\; r\,c_m - k_m\,Q\big). \tag{4.6}$$

Definindo os erros de aproximação $\varepsilon_i = r\,c_i - k_i\,Q$, com $|\varepsilon_i| \leq r/2$ pela condição (4.4), a norma euclidiana do vetor alvo satisfaz:

$$\|\mathbf{w}^*\|^2 = r^2 + \sum_{i=1}^{m} \varepsilon_i^2 \leq r^2\left(1 + \frac{m}{4}\right), \tag{4.7}$$

e portanto

$$\boxed{\|\mathbf{w}^*\| \;\leq\; r\sqrt{1+\frac{m}{4}}.} \tag{4.8}$$

Lema 1 (Recuperação da Ordem via Reticulado Multidimensional)**.** *Sejam $c_1,\ldots,c_m$ medições obtidas pelo QOFA com registrador de tamanho Q, satisfazendo $|r\,c_i - k_i\,Q| \leq r/2$ para inteiros k_i com $\gcd(k_i,r)=1$. Seja $\mathbf{B}_m$ a base definida em (4.5). Então:*

1. *O vetor $\mathbf{w}^* = (r,\varepsilon_1,\ldots,\varepsilon_m)$ pertence a $\Lambda(\mathbf{B}_m)$.*
2. *A primeira coordenada de $\mathbf{w}^*$ é a ordem r.*
3. *O vetor $\mathbf{w}^*$ é o mais curto em $\Lambda(\mathbf{B}_m)$ desde que*

$$r \;<\; \frac{2^{m/4}\,Q^{\,m/(m+1)}}{\sqrt{1+m/4}}. \tag{4.9}$$

Demonstração. O item (1) segue diretamente da definição de $\mathbf{w}^*$ em (4.6): o vetor é obtido por combinação inteira das linhas de $\mathbf{B}_m$ com coeficientes $(r,\,-k_1,\ldots,\,-k_m)$, logo pertence ao reticulado.

O item (2) é imediato pela construção: a primeira coordenada é sempre $r\cdot 1+(-k_i)\cdot 0 = r$.

Para o item (3), calcula-se o determinante da base: $\det(\mathbf{B}_m) = Q^m$. Pelo limitante de Minkowski para o primeiro mínimo λ_1 de um reticulado de dimensão d, tem-se $\lambda_1 \leq \sqrt{d}\,\det(\Lambda)^{1/d}$. O Teorema de Minkowski garante a existência de um vetor não nulo cuja norma é limitada superiormente por uma quantidade proporcional a $Q^{m/(m+1)}$. O algoritmo LLL encontra um vetor de norma no máximo $2^{(m+1-1)/4}\cdot\lambda_1 = 2^{m/4}\,Q^{m/(m+1)}$ em tempo polinomial (Lenstra; Lenstra; Lovász, 1982). Para que $\mathbf{w}^*$ seja o vetor retornado pelo LLL, é suficiente que $\|\mathbf{w}^*\| \leq 2^{m/4}\,Q^{m/(m+1)}$. Combinando com (4.8), obtém-se a condição (4.9). □

Corolário 1 (Limite Inferior para o Registrador de Fase)**.** *Sob as hipóteses do Lema 1, seja $Q' = 2^{l'}$ o tamanho do registrador de fase utilizado na recuperação da ordem via reticulado. Então a condição suficiente*

$$r < \frac{2^{m/4}Q'^{\,m/(m+1)}}{\sqrt{1+m/4}}$$

implica

$$l' > \frac{m+1}{m}\left(\log_2(r) + \frac{1}{2}\log_2\left(1+\frac{m}{4}\right) - \frac{m}{4}\right).$$

Consequentemente, o tamanho mínimo do registrador cresce apenas logaritmicamente com a ordem r, enquanto o aumento do número de medições m fortalece a condição de recuperação fornecida pelo Lema 1.

Observação 4.2.1 (Observação Experimental)**.** *Os resultados obtidos em 5.7 e 5.8 indicam que, para aproximadamente 40 medições independentes, a recuperação da ordem permanece estável quando o registrador de fase é reduzido para cerca de 52% do tamanho tradicionalmente utilizado pelo algoritmo de Shor. Para os casos estudados, isso corresponde a uma economia aproximada de 48% no número de qubits do registrador de fase.*

4.2.3 Por que o vetor mais curto carrega a ordem r

Seja $\Lambda(\mathbf{B}_m)$ o reticulado de recuperação de ordem definido na Seção 4.2. O vetor alvo $\mathbf{w}^* = (r, \varepsilon_1, \ldots, \varepsilon_m)$ pertence a esse reticulado, conforme demonstrado no Lema 1. A questão central é: por que ele é o *mais curto*?

A resposta reside na comparação entre dois tipos de vetores em $\Lambda(\mathbf{B}_m)$:

- **Vetores genéricos** têm norma típica da ordem de $Q^{m/(m+1)}$, que é o primeiro mínimo esperado pelo limitante de Minkowski para um reticulado de determinante Q^m e dimensão $m+1$. Como $Q \approx N^2$ e N é criptograficamente grande, esses vetores são enormes.
- **O vetor alvo** tem norma $\|\mathbf{w}^*\| \leq r\sqrt{1+m/4}$. Como $r < N \ll Q$, tem-se $r \ll Q^{m/(m+1)}$.

O vetor alvo é, portanto, *anormalmente curto* em relação ao que seria esperado para um reticulado desse determinante. Esse fenômeno ocorre porque a ordem r cria uma relação aritmética oculta entre Q e os valores medidos c_i: cada c_i/Q é uma aproximação de k_i/r, e r é exatamente o menor inteiro que satisfaz essas aproximações *simultaneamente* para todos os i. Essa relação se manifesta geometricamente como um vetor anormalmente curto no reticulado.

Para completar o argumento, é necessário garantir que não exista nenhum outro vetor mais curto com primeira coordenada diferente de r. Qualquer vetor $\mathbf{w} = (\lambda_0, \ldots)$ com $0 < \lambda_0 < r$ teria que satisfazer $\lambda_0 c_i \approx k_i' Q$ para inteiros k_i', ou seja, λ_0 também seria uma

ordem de a módulo N. Mas r é a *menor* ordem por definição, de modo que não existe $\lambda_0 \in (0, r)$ que satisfaça essa condição. Consequentemente, qualquer vetor com primeira coordenada menor que r possui pelo menos um erro $|\varepsilon_i'| > r/2$, o que aumenta sua norma acima de $\|\mathbf{w}^*\|$.

4.2.4 Derivação da matriz B_m a partir do problema.

A construção da base $\mathbf{B}_m$ não é arbitrária: ela é derivada diretamente das relações que definem o problema de recuperação de ordem.

O ponto de partida é a equação (4.4): para cada medição c_i, existem inteiros desconhecidos $\lambda_0 = r$ e $\lambda_i = -k_i$ tais que

$$\lambda_0 \cdot c_i + \lambda_i \cdot Q = \varepsilon_i, \qquad |\varepsilon_i| \leq \frac{r}{2}. \tag{4.10}$$

O objetivo é construir uma matriz cujas combinações inteiras de linhas com coeficientes $(\lambda_0, \lambda_1, \ldots, \lambda_m)$ produzam exatamente o vetor $(\lambda_0,\ \varepsilon_1,\ \varepsilon_2,\ \ldots,\ \varepsilon_m)$. Analisando coordenada a coordenada:

- **Primeira coordenada** deve ser λ_0. Isso exige que apenas a primeira linha contribua com coeficiente 1 nessa posição, e todas as demais com 0. Logo, a primeira coluna da matriz é $(1, 0, 0, \ldots, 0)^T$.
- **Coordenada $i+1$** deve ser $\lambda_0 c_i + \lambda_i Q$. Isso exige que a primeira linha contribua com c_i e a linha $i+1$ contribua com Q, sendo as demais nulas. Logo, a coluna $i+1$ é $(c_i, 0, \ldots, Q, \ldots, 0)^T$, onde Q ocupa a posição $i+1$.

Reunindo todas as colunas, obtém-se a estrutura em bloco:

$$\mathbf{B}_m = \left(\begin{array}{c|ccc} 1 & c_1 & \cdots & c_m \\ \hline \mathbf{0} & \multicolumn{3}{c}{Q \cdot \mathbf{I}_m} \end{array}\right), \tag{4.11}$$

onde $\mathbf{I}_m$ é a matriz identidade $m \times m$. Pode-se verificar diretamente que:

$$(r,\ -k_1,\ -k_2,\ \ldots,\ -k_m)\cdot\mathbf{B}_m = (r,\ rc_1 - k_1 Q,\ rc_2 - k_2 Q,\ \ldots,\ rc_m - k_m Q) = (r,\ \varepsilon_1,\ \varepsilon_2,\ \ldots,\ \varepsilon_m). \tag{4.12}$$

A escolha de $Q \cdot \mathbf{I}_m$ no bloco inferior direito é igualmente motivada: colocar Q na diagonal garante que os graus de liberdade $k_1, k_2, \ldots, k_m$ sejam *independentes entre si* e

ortogonais à primeira linha. Essa independência é essencial para que o LLL consiga separar o vetor alvo dos demais vetores do reticulado. Qualquer Q fora da diagonal introduziria dependências artificiais entre as medições que não existem no problema original.

4.2.5 Vantagem geométrica sobre frações continuadas

A principal vantagem da abordagem via reticulado reside em sua *robustez geométrica*. Diferentemente das frações continuadas, que operam essencialmente em dimensão unidimensional, a formulação via reticulado $(m+1)$-dimensional permite:

- maior tolerância a ruído e imprecisão nas medições;
- uso simultâneo de múltiplas medições como restrições conjuntas;
- independência estrutural da condição $\gcd(k,r) = 1$ em cada medição individual, uma vez que a redundância de m medições mitiga o impacto de medições degeneradas;

Enquanto frações continuadas são suficientes no regime ideal com alta precisão ($l \approx 2n$), a formulação via reticulados fornece uma estrutura mais geral, robusta e escalável para o pós-processamento clássico em algoritmos quânticos de fatoração.

4.2.6 Comparação com Frações Continuadas

As frações continuadas e o LLL resolvem a mesma tarefa — recuperar r a partir de aproximações de k/r — mas exploram informações distintas.

As frações continuadas constroem a expansão:

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots}}}, \tag{4.13}$$

onde $a_i = \lfloor x_i \rfloor$ e os convergentes p_k/q_k são as melhores aproximações racionais de $x = c/Q$ com denominador até q_k. O algoritmo para quando encontra q_k tal que $a^{q_k} \equiv 1 \pmod{N}$. Trata-se de uma busca *unidimensional*: o mecanismo consiste em um zoom progressivo na reta real, reduzindo o intervalo de incerteza em torno de c/Q a cada passo, até que o denominador do convergente coincida com r.

O LLL, por sua vez, opera em dimensão $m + 1$. Em vez de refinar uma única aproximação, ele combina m aproximações grosseiras como restrições simultâneas num sistema linear inteiro. A analogia geométrica é a de uma triangulação: nenhuma medição individual precisa ser precisa, mas a interseção de todas aponta para r.

A consequência prática dessa diferença é direta. As frações continuadas exigem que c/Q esteja a distância menor que $1/(2r^2)$ de k/r, o que impõe $Q > r^2 \approx N^2$ e, portanto, $m \approx 2n$ qubits. O LLL com m_{med} medições exige apenas que cada c_i/Q esteja a distância menor que $r/(2Q)$ de k_i/r, uma tolerância proporcionalmente muito maior. Uma interpretação heurística do modelo sugere que o aumento do número de medições reduz a precisão individual necessária em cada amostra. Isso indica uma dependência efetiva entre Q e r substancialmente mais favorável do que a condição $Q > r^2$ exigida pelas frações continuadas, fundamentando analiticamente a redução empírica de $\approx 48\%$ no tamanho do registrador de fase observada nos experimentos da Seção 5.7 e 5.8.

5 Nossa simulação e testes

Com o intuito de validar tanto as propriedades fundamentais do algoritmo de Shor quanto a eficácia da abordagem proposta neste trabalho, desenvolvemos um simulador clássico utilizando a linguagem *Python* (Gomes, 2026). A construção dessa ferramenta justifica-se pela necessidade de um ambiente controlado para testes, embora a transposição do modelo quântico para o domínio clássico imponha desafios computacionais significativos.

O algoritmo de Shor, em sua formulação original, exige o processamento de estados quânticos em um espaço de Hilbert, figura 3, cuja dimensão cresce exponencialmente com o número de qubits. Na prática, em uma simulação convencional, essa representação é frequentemente realizada por meio de vetores (arrays) de ponto flutuante, nos quais cada posição corresponde à amplitude de probabilidade de um estado quântico específico. Consequentemente, para números de relevância criptográfica, a memória necessária para armazenar tais amplitudes excede rapidamente a capacidade de qualquer supercomputador atual.

Diante dessa barreira de escalabilidade, nossa simulação incorporou adaptações estratégicas que viabilizam a fatoração de inteiros de maior magnitude sem o esgotamento dos recursos de hardware. Para assegurar a integridade desses resultados e confirmar que tais simplificações não comprometeram a fidelidade do algoritmo, foram aplicados testes de forma a garantir a corretude do comportamento do simulador frente aos benchmarks teóricos.

5.1 Passos da Simulação do Algoritmo de Shor

Independentemente de rodar a simulação em modo adaptado ou não, nós precisamos ter uma definição de quais passos serão executados em ambos os casos. O fluxo lógico da simulação segue as etapas descritas abaixo:

1. Verificar se o número N é um primo ou potência de primo;

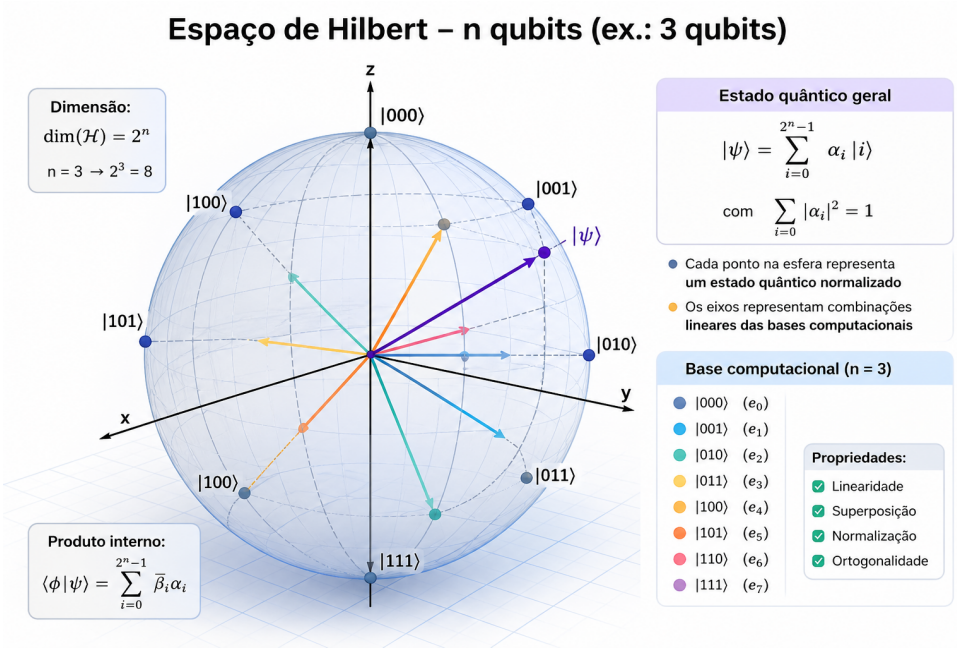

Figura 3: Espaço Hilbert

2. Em caso afirmativo, retornar o número primo e sua potência;
3. Caso contrário, seguir os passos abaixo;
4. Escolher um número aleatório x, onde $0 < x < N$;
5. Verificar se $mdc(x, N) = 1$;
6. Se for 1, ir para o passo 8;
7. Retornar $mdc(x, N)$ e $N/mdc(x, N)$;
8. Simular a rotina quântica para encontrar a ordem r (busca de ordem quântica);
9. Executar pós-processamento e validar r como a ordem de x;
10. Se a busca de ordem falhar, voltar ao passo 4;
11. Retornar $mdc((x^{r/2} - 1), N)$ e $mdc((x^{r/2} + 1), N)$.

5.2 Estimativa de Picos

Dentre as etapas do algoritmo de Shor apresentadas na Seção 5.1, a sub-rotina de busca de ordem quântica (QOFA - *Quantum Order Finding Algorithm*) destaca-se como a mais

importante. É por meio desta rotina que a complexidade do problema da fatoração é reduzida de exponencial para polinomial.

Resumidamente, o processo de geração dos picos segue os seguintes passos:

1. Preparação de um arranjo com N^2 posições;
2. Atribuição do valor $x^i \pmod N$ para cada posição i do arranjo;
3. Execução da IFFT sobre o arranjo resultante;
4. Identificação e retorno de um dos picos gerados pela transformada.

Nesse trabalho, a simulação do QOFA é executada sob duas perspectivas distintas: a metodologia padrão e a metodologia simplificada. A metodologia padrão baseia-se na aplicação da Transformada Rápida de Fourier Inversa (IFFT) sobre o vetor de estado completo, mapeado em uma estrutura de dados de alta dimensionalidade. Embora forneça a distribuição de probabilidade exata em todo o espaço de Hilbert, essa abordagem impõe uma demanda de memória proibitiva para sistemas de larga escala. Em contrapartida, a abordagem simplificada prioriza a eficiência ao identificar diretamente os valores de estado associados aos máximos locais de probabilidade (picos), mitigando o custo computacional ao evitar a construção integral do espectro.

5.2.1 Estimativa Padrão

Na metodologia padrão, o resultado da IFFT consiste em uma estrutura de dados contendo picos de amplitudes, que representam os estados de maior probabilidade em um sistema quântico. Para isolar esses picos, implementou-se uma técnica de busca binária iniciada a partir de uma posição aleatória (na realidade é pseudoaleatória, ver o trabalho de (Von Neumann *et al.*)). Contudo, como salientado anteriormente, a escalabilidade desta abordagem é limitada pelo consumo de memória, o que inviabiliza testes com números de grande magnitude.

A metodologia padrão pode ser interpretada como um pipeline composto por quatro etapas principais: (i) construção de um padrão periódico no registrador, (ii) transformação via IFFT, (iii) cálculo e acumulação das probabilidades e (iv) amostragem simulando a medição quântica. Diferentemente de uma descrição puramente conceitual, nesta seção cada uma dessas etapas é diretamente associada aos algoritmos implementados.

A primeira etapa consiste na construção explícita de um padrão periódico no registrador, conforme descrito no Algoritmo 1. Nesse procedimento, a variável *interval* define a periodicidade r a ser simulada, enquanto o laço iterativo (linhas 2–5) realiza a marcação dos índices múltiplos desse valor. Como resultado, o registrador passa a conter uma estrutura esparsa, na qual apenas posições específicas possuem valor unitário. Essa construção é fundamental, pois estabelece no domínio do tempo a periodicidade que será posteriormente evidenciada no domínio da frequência pela IFFT.

Algoritmo 1 Atualização de Estados por Intervalo

Require: $register, interval$

Ensure: $register$ atualizado conforme periodicidade

1: $index \leftarrow 1$
2: **while** $index <$ length$(register)$ **do**
3: $\quad register[index] \leftarrow 1$
4: $\quad index \leftarrow index + interval$
5: **end while**

Algoritmo 2 Cálculo de Probabilidades

Require: $order, number\_of\_states, register$

Ensure: $new\_register$ normalizado

1: $new\_register \leftarrow$ zeros(length$(register)$)
2: $offset \leftarrow number\_of\_states/order$
3: $iterations \leftarrow order$
4: **for** $index = 0$ **to** $iterations - 1$ **do**
5: $\quad position \leftarrow$ int$(index \times offset)$
6: $\quad new\_register[position] \leftarrow |register[position]|^2$
7: $\quad new\_register[position + 1] \leftarrow |register[position + 1]|^2$
8: **end for**
9: **return** $new\_register / \sum(new\_register)$ {Normalização estatística}

A partir desse registrador inicial, aplica-se a transformada inversa de Fourier, cujo efeito não é explicitado em pseudocódigo, mas se reflete diretamente nos dados de entrada do Algoritmo 2. Nesse ponto, os valores do registrador deixam de ser binários e passam a representar amplitudes complexas. O Algoritmo 2 realiza então a conversão dessas amplitudes em probabilidades, conforme indicado na linha 6, onde se computa $|register[position]|^2$.

Algoritmo 3 Geração de Probabilidade Cumulativa

Require: $order, number\_of\_states, register$

Ensure: Listas $nsum, psum$

1: $psum \leftarrow [0] \times (2 \times order + 1)$
2: $nsum \leftarrow [0] \times (2 \times order + 1)$
3: $k \leftarrow number\_of\_states/order$
4: $total \leftarrow 0$
5: **for** $i = 0$ **to** $order - 1$ **do**
6: $\quad pos \leftarrow \text{int}(i \times k)$
7: $\quad psum[2 \times i] \leftarrow pos$
8: $\quad total \leftarrow total + register[pos]$
9: $\quad nsum[2 \times i] \leftarrow total$
10: $\quad psum[2 \times i + 1] \leftarrow pos + 1$
11: $\quad total \leftarrow total + register[pos + 1]$
12: $\quad nsum[2 \times i + 1] \leftarrow total$
13: **end for**
14: $psum[2 \times order - 1] \leftarrow -\text{int}(\text{random}() \times number\_of\_states)$ {Resíduo de probabilidade}
15: $nsum[2 \times order - 1] \leftarrow 1$
16: **return** $nsum, psum$

Observa-se que o algoritmo não percorre todo o registrador, mas apenas posições específicas determinadas por *offset* (linha 2). Essa escolha reflete uma otimização importante: como a IFFT concentra probabilidade em torno de múltiplos de $\frac{N}{r}$, o algoritmo explora diretamente essa estrutura ao invés de avaliar todos os estados. A inclusão da posição adjacente (*position+1*, linha 7) captura efeitos de espalhamento espectral, garantindo maior robustez na estimação.

Algoritmo 4 Simulação de Medição

Require: $nsum, psum, number$

Ensure: $result$ (amostragem da medição)

1: $result \leftarrow [0] \times number$
2: **for** $i = 0$ **to** $number - 1$ **do**
3: $random\_value \leftarrow$ random()
4: $result[i] \leftarrow$ buscabin$(nsum, psum, random_value)$
5: **if** $result[i] = 0$ **then**
6: $result[i] \leftarrow 1$
7: **end if**
8: **end for**
9: **return** $result$

Na sequência, o Algoritmo 3 transforma a distribuição discreta obtida em uma função de probabilidade acumulada. As variáveis *psum* e *nsum* armazenam, respectivamente, os índices relevantes e os valores acumulados. Durante o laço principal (linhas 5–12), cada iteração adiciona incrementalmente a probabilidade associada a um pico, produzindo uma estrutura monotonicamente crescente. Essa transformação é essencial para viabilizar a etapa de amostragem eficiente, uma vez que permite mapear valores aleatórios uniformes em estados do registrador.

A etapa de medição é simulada pelo Algoritmo 4, no qual cada amostra é gerada a partir de um valor aleatório uniforme (linha 4). Esse valor é então utilizado como entrada para a função *buscabin*, responsável por localizar o intervalo correspondente na distribuição acumulada. O ajuste realizado nas linhas 6–8 evita a ocorrência de estados nulos, garantindo consistência na amostragem.

O mecanismo central dessa etapa é detalhado no Algoritmo 5, que implementa uma busca binária recursiva sobre o vetor acumulado. A cada chamada, o espaço de busca é reduzido pela metade (linhas 12–18), resultando em complexidade $O(\log n)$. Essa escolha é particularmente relevante, pois a alternativa linear se tornaria rapidamente inviável conforme o número de estados cresce exponencialmente com o número de qubits simulados.

Algoritmo 5 Busca Binária Customizada

Require: $sum, psum, m$

Ensure: Valor mapeado em $psum$

```
1: n ← length(sum)
2: if n = 0 then
3:    return 0
4: else if n = 1 then
5:    return psum[0]
6: else if n = 2 then
7:    if m ≤ sum[0] then
8:       return psum[0]
9:    else
10:      return psum[1]
11:   end if
12: else
13:   meio ← n//2
14:   if m = sum[meio] then
15:      return psum[meio]
16:   else if m < sum[meio] then
17:      return buscabin(sum[0 : meio], psum[0 : meio], m)
18:   else
19:      return buscabin(sum[meio : n], psum[meio : n], m)
20:   end if
21: end if
```

Por fim, a integração dessas etapas evidencia que a qualidade da estimativa do período está diretamente ligada à forma como o padrão periódico inicial (Algoritmo 1) é propagado pelas etapas subsequentes. Qualquer imprecisão na construção inicial ou na discretização das probabilidades impacta diretamente a distribuição acumulada e, consequentemente, os resultados da medição simulada.

5.2.2 Estimativa Simplificada

Para contornar as limitações de memória física, este trabalho propõe uma técnica fundamentada na estimativa analítica dos picos de medição. Em vez de instanciar o vetor de estado completo para a aplicação da Transformada Rápida de Fourier Inversa (IFFT),

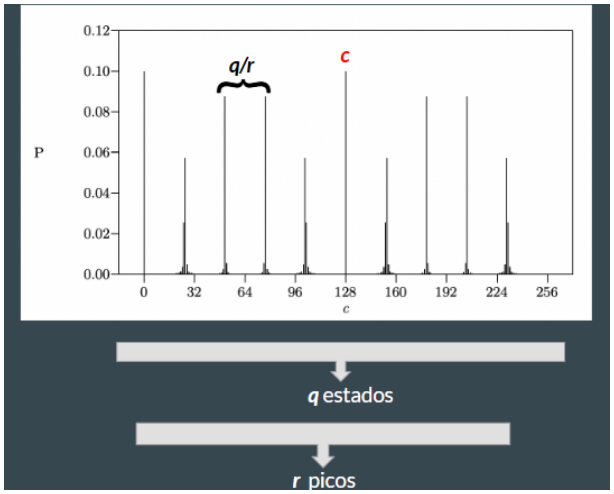

Figura 4: Padrão de picos gerados pelo QOFA

o simulador explora as propriedades matemáticas intrínsecas ao problema da busca de ordem:

- **Abstração do Espaço de Estados:** Ao abdicar do cálculo de estados intermediários, o simulador concentra-se no fato de que o resultado após rodar a IFFT sobre os estados contendo a periodicidade da função $f(x) = a^x \pmod{N}$ será a existência de r picos, onde r é a ordem, distrubuídos de forma equidistantes q/r entre os q estados, ver algoritmo 6. Exemplo: Na figura 4, o valor c é um pico, está equidistante dos demais e é um provável valor a ser recuperado na medição.

- **Eficiência de Memória:** Tal abordagem reduz drasticamente a carga computacional, permitindo a validação de propostas de otimização, como o uso de ordens ímpares ou divisores primos de r, sem a necessidade de processamento de matrizes de dimensões proibitivas.

Algoritmo 6 Gera picos de amplitude

Require: Q (tamanho do registrador), r (ordem), M (número de medições)

Ensure: Lista `measurements` com $2M$ valores

Inicializar `measurements` $\leftarrow [0]^{2M}$
`offset` $\leftarrow Q/r$
for $i = 0$ até $M - 1$ **do**
    `spike` $\leftarrow$ inteiro aleatório em $[0, r]$
    `pos` $\leftarrow$ `spike` $\cdot$ `offset`
    `measurements`$[2i] \leftarrow \lfloor$`pos`$\rfloor$
    `measurements`$[2i + 1] \leftarrow \lfloor$`pos` $+ 1\rfloor$
end for

return `measurements`

Esses algoritmos não constituem apenas uma implementação computacional, mas uma representação explícita da dinâmica esperada do QOFA. A periodicidade introduzida no domínio temporal é convertida em concentração espectral pela IFFT, permitindo reproduzir os picos de probabilidade associados à ordem r.

5.3 Pós-processamento

Após a etapa de medição na rotina quântica, o algoritmo transita para uma fase de pós-processamento estritamente clássica. Esta etapa é fundamental para converter as amostras brutas do vetor de estado em resultados estruturados, como a extração do período r. Na simulação que construímos, esta conversão pode ser realizada através de duas metodologias distintas:

1. **Frações Continuadas:** Utilizada para encontrar a melhor aproximação racional d/r a partir de c/q, onde c é o valor medido na rotina quântica e q é o número de estados quânticos gerados na superposção. Este método é eficiente para casos de amostragem única onde a precisão da Transformada de Fourier Quântica (QFT) permite a convergência direta para o denominador desejado.

2. **Redução de Reticulados:** Empregada em cenários de maior complexidade ou em variantes multidimensionais do algoritmo. Através do algoritmo de redução de base LLL (*Lenstra–Lenstra–Lovász*), busca-se o vetor mais curto em uma rede construída

a partir de múltiplas amostras, permitindo a recuperação do resultado mesmo em presença de ruído estatístico ou quando os "picos"de amplitude estão distribuídos de forma não trivial.

Esta dualidade de abordagens permite que o simulador adapte o rigor do tratamento de dados conforme a granulação (fina ou grossa) escolhida na etapa anterior, equilibrando a fidelidade do resultado com o custo computacional clássico.

5.4 Preparando a massa de testes

Para poder rodar as simulações e verificar os resultados alcançados com a proposta, nós precisamos gerar uma massa de teste composta por números primos que serão usados como fatores para gerar o número composto que será utilizada como informação de entrada para o algoritmo de fatoração de Shor.

É importante lembrar que vamos usar somente números *safe primes*. A escolha por utilizar exclusivamente *safe primes* não foi arbitrária. Em simulações clássicas do algoritmo de Shor, especialmente para inteiros de grande magnitude, a principal limitação computacional decorre da necessidade de representar explicitamente o espaço de Hilbert associado ao QOFA. Nessa situação, utilizar números cuja estrutura algébrica favoreça a recuperação da ordem r torna-se uma estratégia importante para viabilizar experimentos em larga escala.

Quando $N = p \cdot q$ é composto por *safe primes*, as ordens possíveis no grupo multiplicativo $\mathbb{Z}_N^*$ tornam-se significativamente mais estruturadas e previsíveis. Isso reduz drasticamente o espaço de busca para recuperação da ordem durante o pós-processamento clássico, permitindo validar os resultados do QOFA sem a necessidade de reconstruir integralmente a distribuição de probabilidades do sistema quântico.

Além disso, conforme discutido anteriormente, *safe primes* aumentam substancialmente a probabilidade de sucesso do algoritmo de Shor e de suas variantes otimizadas, especialmente quando combinados com restrições adicionais como o uso do símbolo de Jacobi. Dessa forma, essa escolha permitiu executar simulações com inteiros de tamanho criptograficamente relevante, incluindo instâncias de até 512 bits, mantendo o custo computacional clássico dentro de limites viáveis.

Para que possamos rodar a simulação com entradas de tamanho relevante para área de criptografia, nós usamos o openssl que fornece um CLI (command line interface) que

possibilita gerar números primos, inclusive com a restrição de serem safe primes. O comando utilizado é `openssl prime -generate -safe -bits <num_bits>`. Então, por exemplo, se queremos um conjunto de números compostos N para simulação de 256 bits, executamos `openssl prime -generate -safe -bits 128` duas vezes para gerar os fatores que vão formar o número composto de 256 bits.

Para coletar dados sobre a proposta, iremos executar a simulação com números de 128, 256 e 512 bits. Para isso, configuramos 5 *safe primes* de 64 bits, 5 *safe primes* de 128 bits e 5 *safe primes* de 256 bits, o que resulta em 10 números compostos N para cada tamanho de bits (combinação de 5 elementos tomados 2 a 2).

5.5 Parâmetros de execução da simulação

Para cada execução da simulação, nós podemos mudar alguns parâmetros de execução que fazem a simulação rodar de maneiras diferentes a depender dessas combinações.

Para cada execução da simulação, é possível ajustar os seguintes parâmetros:

Número de medições: define quantas vezes rodamos a rotina QOFA que retorna o candidato de ordem.

Tamanho do registrador: define quantos bits vamos usar no registrador. A ideia é variar esse número e avaliar qual é o limite inferior de número de bits que possibilita a execução da simulação com sucesso, ou seja, que a fatoração consiga encontrar os fatores de N.

Número de execuções: define quantas vezes vamos rodar a simulação para uma mesma entrada. Usamos esse parâmetro para extrair informações de médias e desvio padrão entre execuções diferentes da simulação com mesma entrada e mesma configuração.

Habilitar o uso de símbolo de Jacobi: com o intuito de otimizar a performance do algoritmo QOFA em ambiente de simulação, é possível habilitar o uso do símbolo de Jacobi para a seleção de bases. Ao ativar essa funcionalidade, o valor x, selecionado para a base da exponenciação, deixa de ser estritamente aleatório e passa a ser condicionado à restrição $\left(\frac{x}{n}\right) = -1$. A imposição dessa propriedade algébrica eleva a assertividade do QOFA a patamares próximos de 100%, conforme documentado em (Silva; Santos; Kowada, 2025; Leander, 2002). Ver prova no apêndice A.

5.6 Plano de execução

A ideia é executar 6 vezes a simulação para cada combinação entrada + parãmetros. A primeira execução é descartada para evitar impactos de inicialização nas métricas geradas. A tabela 2 mostra as combinações para todas as execuções da simulação realizadas para gerar as métricas. O intervalo entre 0.5 e 0.6 é o único que verificamos a cada 0.01%. É nesse intervalo que começamos a verificar a abordagem de reticulados começar a conseguir recuperar a ordem. Para as execuções de frações continuadas, os valores de 0.51 até 0.59 não foram utilizados porque eles não se mostraram necessários, já que mesmo com 0.6 do tamanho original do registrador de fase, essa abordagem não conseguiu obter sucesso na execução do algoritmo.

Tabela 2: Combinações de execução

Bits	**Número de medições**	**Tamanho do registrador(%)**
128 bits	5	0.1 até 0.5; 0.51 até 0.59; 0.6 até 1
128 bits	10	0.1 até 0.5; 0.51 até 0.59; 0.6 até 1
128 bits	40	0.1 até 0.5; 0.51 até 0.59; 0.6 até 1
256 bits	5	0.1 até 0.5; 0.51 até 0.59; 0.6 até 1
256 bits	10	0.1 até 0.5; 0.51 até 0.59; 0.6 até 1
256 bits	40	0.1 até 0.5; 0.51 até 0.59; 0.6 até 1
512 bits	5	0.1 até 0.5; 0.51 até 0.59; 0.6 até 1
512 bits	10	0.1 até 0.5; 0.51 até 0.59; 0.6 até 1
512 bits	40	0.1 até 0.5; 0.51 até 0.59; 0.6 até 1

Conforme descrito anteriormente, as simulações foram conduzidas para diferentes valores de N (inteiros a serem fatorados), tamanho de bits de N e diferentes tamanhos de registrador , comparando a quantidade de execuções necessárias para conseguir fatorar o número N, quando possível. Além disso, testamos a simulação com pós-processamento original proposto por Shor com o pós-processamento que estamos avaliando, usando reticulado.

5.7 Execuções com Frações Continuadas

Como experimento de referência, avaliou-se o desempenho do método tradicional de recuperação da ordem baseado em frações continuadas utilizando diferentes tamanhos para o registrador de fase. O objetivo foi verificar o impacto da redução do número de qubits sobre a capacidade de reconstruir corretamente a ordem (r) a partir das medições produzidas pelo QOFA.

De acordo com a formulação original do algoritmo de Shor (Shor, 1999), o registrador de fase deve possuir tamanho suficiente para garantir ($Q = 2^m > N^2$), o que corresponde aproximadamente a ($m \approx 2n$) qubits para um inteiro (N) de (n) bits. Essa condição assegura que a aproximação racional (c/Q) possua precisão adequada para que a ordem possa ser recuperada por meio do algoritmo de frações continuadas.

As Tabelas 3, 4 e 5 apresentam os resultados obtidos para diferentes quantidades de medições e diferentes proporções do número original de qubits. Observa-se que o método de frações continuadas falhou sistematicamente sempre que o registrador de fase foi reduzido, independentemente da quantidade de medições realizadas.

Um aspecto particularmente relevante é que o aumento do número de medições não produziu qualquer melhoria nos resultados. Mesmo com 40 medições independentes, a recuperação da ordem permaneceu inviável para todas as configurações que utilizavam menos de 100% dos qubits originalmente previstos pelo algoritmo de Shor. Esse comportamento evidencia uma limitação estrutural das frações continuadas: o método opera sobre uma única aproximação racional por vez e não dispõe de mecanismos para combinar múltiplas observações parciais em uma estimativa conjunta da ordem.

Os resultados confirmam, portanto, que a redução do registrador de fase provoca perda de precisão suficiente para inviabilizar a recuperação correta da ordem por frações continuadas. Em outras palavras, a simples repetição das medições não é capaz de compensar a diminuição da informação disponível em cada observação individual.

A Figura 5 resume esse comportamento. Observa-se uma transição abrupta entre falha e sucesso: o método somente apresenta sucesso quando o tamanho integral do registrador recomendado por Shor (Shor, 1999) é preservado. Esse resultado reforça a dependência das frações continuadas em relação à alta precisão da estimativa de fase e justifica a busca por técnicas alternativas de pós-processamento capazes de explorar múltiplas medições simultaneamente.

Tabela 3: Resultados para 5 medições

Sucesso	**% Qubits**	**Qubits original**
Não	0.1 até 0.9	128
Sim	1.0	128
Não	0.1 até 0.9	256

Continua na próxima página

Sucesso	% Qubits	Qubits original
Sim	1.0	256
Não	0.1 até 0.9	512
Sim	1.0	512

Tabela 4: Resultados para 10 medições

Sucesso	% Qubits	Qubits original
Não	0.1 até 0.9	128
Sim	1.0	128
Não	0.1 até 0.9	256
Sim	1.0	256
Não	0.1 até 0.9	512
Sim	1.0	512

Tabela 5: Resultados para 40 medições

Sucesso	% Qubits	Qubits original
Não	0.1 até 0.9	128
Sim	1.0	128
Não	0.1 até 0.9	256
Sim	1.0	256
Não	0.1 até 0.9	512
Sim	1.0	512

5.8 Execuções com Reticulado

Após estabelecer o comportamento de referência utilizando frações continuadas, os mesmos experimentos foram repetidos empregando o método de recuperação da ordem baseado em redução de reticulados. O objetivo foi avaliar se a utilização de múltiplas medições combinadas com pós-processamento geométrico seria capaz de compensar a perda de precisão decorrente da redução do número de qubits.

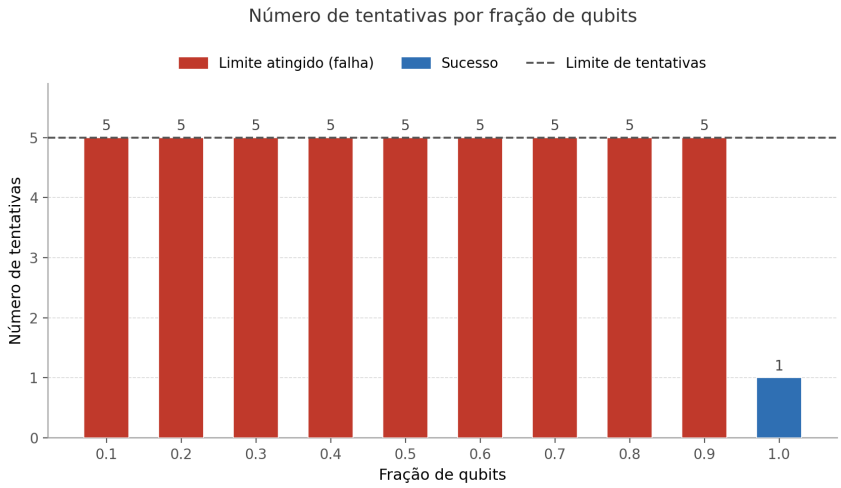

Figura 5: Resultados das execuções com frações continuadas

Os resultados obtidos para 5 medições mostram comportamento semelhante ao observado nas frações continuadas. Nessas configurações, a quantidade de informação disponível ainda não é suficiente para permitir a reconstrução confiável da ordem quando o registrador de fase é reduzido. Como consequência, o método continua apresentando falhas para todas as configurações com menos de 100% dos qubits originalmente previstos.

Entretanto, um comportamento significativamente diferente surge quando o número de medições é aumentado para 10. Conforme apresentado na Tabela 7, o método baseado em reticulados passa a recuperar corretamente a ordem mesmo em cenários onde apenas 55% do número original de qubits é utilizado. O melhor resultado foi com 40 medições que passa a recuperar corretamente com apenas 52% do tamanho original.

Esse resultado constitui a principal evidência experimental da hipótese investigada nesta tese. Diferentemente das frações continuadas, a abordagem baseada em reticulados não depende exclusivamente da precisão de uma única medição. Em vez disso, ela explora simultaneamente múltiplas observações, transformando o problema de recuperação da ordem em um problema geométrico de busca por vetores curtos em um reticulado. A redundância introduzida pelas diversas medições permite compensar parcialmente a perda de precisão causada pela redução do registrador de fase.

Comparando-se diretamente os resultados das Seções 5.7 e 5.8, observa-se que o aumento do número de medições não produz benefícios para as frações continuadas, mas passa a desempenhar papel fundamental na abordagem baseada em reticulados. Esse comportamento está de acordo com a análise teórica desenvolvida no Capítulo 4, onde se mostrou que múltiplas medições podem fornecer restrições adicionais capazes de auxiliar a recuperação da ordem mesmo quando a precisão individual das amostras é reduzida.

Os resultados obtidos indicam que a utilização de aproximadamente 52% do número de qubits originalmente requerido pelo algoritmo de Shor pode ser suficiente para recuperar corretamente a ordem quando combinada com um pós-processamento clássico baseado em reticulados e um número adequado de medições. Em termos práticos, isso corresponde a uma redução próxima de 48% nos requisitos de qubits do registrador de fase, preservando a capacidade de recuperação da ordem nas instâncias avaliadas.

Tabela 6: Resultados para 5 medições

Sucesso	**% Qubits**	**Qubits original**
Não	0.1 até 0.9	128
Sim	1.0	128
Não	0.1 até 0.9	256
Sim	1.0	256
Não	0.1 até 0.9	512
Sim	1.0	512

Tabela 7: Resultados para 10 medições

Sucesso	**% Qubits**	**Qubits original**
Não	0.1 até 0.54	128
Sim	0.55 até 1.0	128
Não	0.1 até 0.54	256
Sim	0.55 até 1.0	256
Não	0.1 até 0.54	512
Sim	0.55 até 1.0	512

Tabela 8: Resultados para 40 medições

Sucesso	**% Qubits**	**Qubits original**
Não	0.1 até 0.51	128
Sim	0.52 até 1.0	128

Continua na próxima página

Sucesso	% Qubits	Qubits original
Não	0.1 até 0.51	256
Sim	0.52 até 1.0	256
Não	0.1 até 0.51	512
Sim	0.52 até 1.0	512

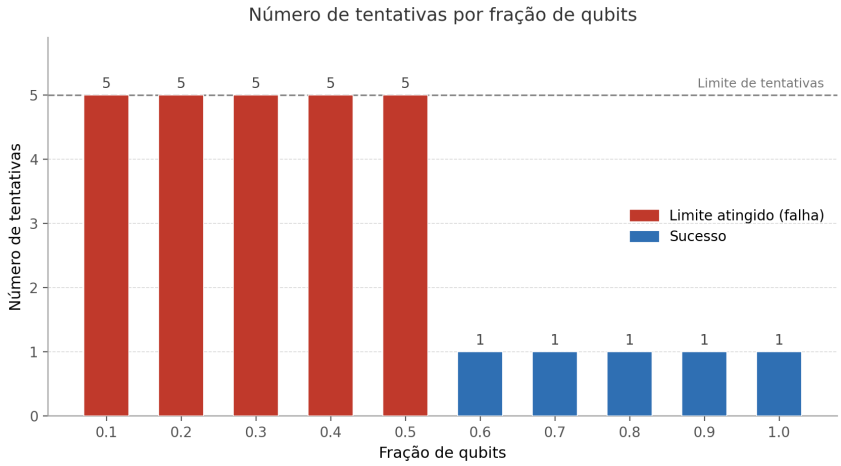

Figura 6: Resultados das execuções com reticulados

6 Conclusão

Neste trabalho, investigou-se a aplicação de técnicas de simulação do algoritmo de Shor com ênfase no aprimoramento do estágio clássico de pós-processamento, explorando o uso de estruturas de reticulado como alternativa às abordagens tradicionais baseadas em frações contínuas. O estudo foi motivado pelas limitações práticas associadas à implementação do algoritmo em hardware quântico real, especialmente no que se refere à quantidade de qubits necessária para a representação do registrador quântico e à sensibilidade do algoritmo a erros.

Os resultados obtidos ao longo das simulações indicam que a incorporação de métodos baseados em redução de bases de reticulados, como o algoritmo LLL, constitui uma abordagem promissora para a recuperação da ordem r a partir das medições quânticas. Em particular, observou-se que essa estratégia permite uma interpretação mais robusta das aproximações racionais derivadas do processo de medição, mitigando limitações associadas à precisão finita e à presença de ruído.

Um dos principais achados deste trabalho foi a constatação de que o uso de reticulados possibilita uma redução considerável no tamanho do registrador quântico necessário nas simulações. Essa redução decorre do fato de que a técnica baseada em reticulados consegue extrair informações relevantes mesmo quando a precisão da estimativa da fase e, consequentemente, o número de qubits utilizados é menor do que o exigido pelos métodos clássicos tradicionais. Em termos práticos, isso implica que o algoritmo pode operar com registradores mais compactos sem comprometer significativamente a taxa de sucesso na determinação da ordem.

Embora a abordagem proposta introduza um custo clássico adicional decorrente da redução de bases de reticulado, esse custo permanece polinomial e, nas dimensões consideradas neste trabalho, é significativamente inferior ao custo físico associado ao aumento do número de qubits e da profundidade dos circuitos quânticos. Dessa forma, os resultados sugerem que o *trade-off* entre recursos quânticos e pós-processamento clássico

pode ser vantajoso em cenários onde os qubits constituem o principal recurso limitante. Essa observação possui implicações diretas para a viabilidade experimental do algoritmo de Shor. A limitação no número de qubits disponíveis é um dos principais gargalos no desenvolvimento de computadores quânticos escaláveis. Portanto, qualquer abordagem que permita reduzir a demanda por recursos quânticos, mantendo desempenho aceitável, representa um avanço relevante. Os resultados aqui apresentados sugerem que técnicas híbridas, combinando processamento quântico e métodos clássicos mais sofisticados, podem desempenhar papel fundamental na transição entre simulações teóricas e implementações práticas.

Além disso, verificou-se que o uso de reticulados proporciona maior flexibilidade na escolha dos parâmetros do algoritmo, como o tamanho do registrador de fase, permitindo explorar configurações que seriam inviáveis sob a abordagem tradicional. Essa flexibilidade pode ser explorada para otimizar o equilíbrio entre custo computacional e probabilidade de sucesso, especialmente em cenários onde o número de execuções do circuito quântico é limitado.

Do ponto de vista metodológico, o trabalho também contribui ao demonstrar, por meio de experimentação computacional, a aplicabilidade de técnicas clássicas avançadas de teoria dos números e álgebra linear no contexto da computação quântica. Essa integração evidencia o caráter interdisciplinar do campo e reforça a importância de abordagens híbridas no avanço do estado da arte.

Por fim, destaca-se que, embora os resultados sejam promissores, eles foram obtidos em ambiente de simulação, o que implica certas idealizações, como ausência de ruído físico realista e limitações específicas de hardware. Ainda assim, as tendências observadas fornecem evidências consistentes de que o uso de reticulados pode representar uma melhoria concreta na implementação prática do algoritmo de Shor.

6.1 Trabalhos Futuros

Apesar dos avanços apresentados, diversas questões permanecem em aberto, abrindo oportunidades relevantes para investigações futuras.

Uma direção natural consiste na validação experimental dos resultados em hardware quântico real. A transição de simulações para dispositivos físicos envolve desafios adicionais, como decoerência, erros de porta e limitações de conectividade entre qubits. Avaliar o desempenho das técnicas baseadas em reticulados nesse contexto é essencial para de-

terminar sua aplicabilidade prática. Em particular, seria interessante investigar como o ruído afeta a qualidade das aproximações utilizadas como entrada para os algoritmos de redução de reticulado.

Outra linha promissora envolve o aperfeiçoamento dos algoritmos de redução de reticulado utilizados no pós-processamento. Embora o algoritmo LLL seja amplamente utilizado devido à sua eficiência polinomial, ele não fornece soluções ótimas em todos os casos. Alternativas como BKZ (*Block Korkine-Zolotarev*) podem oferecer melhor qualidade de redução, ao custo de maior complexidade computacional. A análise do *trade-off* entre custo clássico adicional e ganho em precisão constitui um tema relevante para estudos futuros.

Além disso, pode-se explorar a integração mais profunda entre o circuito quântico e o pós-processamento clássico, desenvolvendo abordagens adaptativas nas quais os resultados parciais das medições influenciam dinamicamente a execução do algoritmo. Essa estratégia pode permitir uma utilização ainda mais eficiente dos recursos quânticos, reduzindo a necessidade de execuções repetidas.

No âmbito teórico, há espaço para uma análise mais rigorosa da probabilidade de sucesso do algoritmo quando combinado com técnicas de reticulado, especialmente considerando diferentes distribuições de erro nas medições. Tal análise poderia fornecer garantias formais sobre o desempenho da abordagem proposta, contribuindo para sua consolidação.

Adicionalmente, seria relevante investigar a relação entre o tamanho mínimo do registrador quântico e a eficiência do pós-processamento baseado em reticulados, buscando estabelecer limites inferiores mais precisos. Esse tipo de resultado poderia orientar o projeto de arquiteturas quânticas mais eficientes, alinhadas às necessidades específicas do algoritmo de Shor.

Por fim, uma direção de grande impacto envolve a integração com técnicas de correção de erros quânticos. Uma vez que a redução do número de qubits pode aliviar parcialmente a sobrecarga associada à correção de erros, compreender como essas duas abordagens interagem pode abrir caminho para implementações mais viáveis em larga escala.

6.2 Considerações Finais

Em síntese, este trabalho demonstrou que o uso de reticulados no pós-processamento do algoritmo de fatoração do Shor não apenas constitui uma alternativa viável às técnicas

tradicionais, mas também oferece vantagens concretas, especialmente no que diz respeito à redução do tamanho do registrador quântico necessário. Essa contribuição reforça a importância de abordagens híbridas e sugere que avanços significativos na computação quântica podem emergir da combinação inteligente entre algoritmos quânticos e técnicas clássicas avançadas.

O aprofundamento das investigações propostas nos trabalhos futuros poderá consolidar essas evidências e contribuir de maneira decisiva para a viabilização prática de algoritmos quânticos para problemas de relevância criptográfica.

REFERÊNCIAS

BARENDS, Rami *et al.* Superconducting quantum circuits at the surface code threshold for fault tolerance. **Nature**, Nature Publishing Group UK London, v. 508, n. 7497, p. 500–503, 2014.

BASTOS, Daniel Chicayban; KOWADA, Luis Antonio Brasil. How to detect whether Shor's algorithm succeeds against large integers without a quantum computer. **Procedia Computer Science**, Elsevier, v. 195, p. 145–151, 2021.

BENNETT, Charles H. Logical reversibility of computation. **IBM journal of Research and Development**, IBM, v. 17, n. 6, p. 525–532, 1973.

CAMPBELL, Earl; KHURANA, Ankur; MONTANARO, Ashley. Applying quantum algorithms to constraint satisfaction problems. **Quantum**, Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften, v. 3, p. 167, 2019.

COOLEY, James W; TUKEY, John W. An algorithm for the machine calculation of complex Fourier series. **Mathematics of computation**, JSTOR, v. 19, n. 90, p. 297–301, 1965.

EKERÅ, Martin. On completely factoring any integer efficiently in a single run of an order-finding algorithm. **Quantum Information Processing**, Springer, v. 20, n. 6, p. 205, 2021.

EKERÅ, Martin. **On factoring integers, and computing discrete logarithms and orders, quantumly**. 2024. Tese (Doutorado) – KTH Royal Institute of Technology.

EKERÅ, Martin; HÅSTAD, Johan. Quantum algorithms for computing short discrete logarithms and factoring RSA integers. *In:* SPRINGER. INTERNATIONAL Workshop on Post-Quantum Cryptography. [*S. l.: s. n.*], 2017. p. 347–363.

FOWLER, Austin G *et al.* Surface codes: Towards practical large-scale quantum computation. **Physical Review A—Atomic, Molecular, and Optical Physics**, APS, v. 86, n. 3, p. 032324, 2012.

GOMES, Fábio. **qft-simul: Quantum Fourier Transform Simulator**. [*S. l.*]: GitHub, 2026. https://github.com/fabio-gomes/qft-simul.

GROSSHANS, Frédéric *et al.* Factoring safe semiprimes with a single quantum query. **arXiv preprint arXiv:1511.04385**, 2015.

JOHNSTON, Anna. Shor's algorithm and factoring: don't throw away the odd orders. **Cryptology ePrint Archive**, 2017.

KIM, Yoongu *et al.* Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors. **ACM SIGARCH Computer Architecture News**, ACM New York, NY, USA, v. 42, n. 3, p. 361–372, 2014.

LEANDER, Gregor. Improving the Success Probability for Shor's Factoring Algorithm. **arXiv preprint quant-ph/0208183**, 2002.

LENSTRA, Arjen K; LENSTRA, Hendrik Willem; LOVÁSZ, László. Factoring polynomials with rational coefficients. Springer Verlag, 1982.

NAKAMOTO, Satoshi. Bitcoin: A peer-to-peer electronic cash system, 2008.

NIELSEN, Michael A; CHUANG, Isaac L. **Quantum computation and quantum information**. [*S. l.*]: Cambridge university press, 2010.

POLLARD, John M. Theorems on factorization and primality testing. *In:* CAMBRIDGE UNIVERSITY PRESS, 3. MATHEMATICAL proceedings of the Cambridge philosophical society. [*S. l.: s. n.*], 1974. v. 76, p. 521–528.

REGEV, Oded. An efficient quantum factoring algorithm. **Journal of the ACM**, ACM New York, NY, v. 72, n. 1, p. 1–13, 2025.

SCHROEDER, Bianca; PINHEIRO, Eduardo; WEBER, Wolf-Dietrich. DRAM errors in the wild: a large-scale field study. **Communications of the ACM**, ACM New York, NY, USA, v. 54, n. 2, p. 100–107, 2011.

SHOR, Peter W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. **SIAM review**, SIAM, v. 41, n. 2, p. 303–332, 1999.

SILVA, Vitor Pio; SANTOS, Fábio Gomes dos; KOWADA, Luis Antonio Brasil. Boosting Shor's Factoring Success with the Jacobi Symbol in Single-Run Order Finding against large integer numbers. *In:* BOOK of Abstracts of the LVII Brazilian Symposium on Operations Research. Gramado, RS: Galoá, 2025. Acesso em: 27 abr. 2026. Disponível em: https://proceedings.science/sbpo/sbpo-2025/papers/boosting-shors-factoring-success-with-the-jacobi-symbol-in-single-run-order-find?lang=pt-br.

TANKASALA, Archana; ILATIKHAMENEH, Hesameddin. Quantum-kit: simulating shor's factorization of 24-bit number on desktop. **arXiv preprint arXiv:1908.07187**, 2019.

VANDERSYPEN, Lieven MK *et al.* Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. **Nature**, Nature Publishing Group UK London, v. 414, n. 6866, p. 883–887, 2001.

VON NEUMANN, John *et al.* Various techniques used in connection with random digits. **John von Neumann, Collected Works**, MacMillan New York, USA, v. 5, n. 768-770, p. 1, 1963.

WANG, David S; HILL, Charles D; HOLLENBERG, Lloyd CL. Simulations of Shor's algorithm using matrix product states. **Quantum Information Processing**, Springer, v. 16, p. 1–13, 2017.

APÊNDICE A – PROVA MATEMÁTICAS

A.1 Melhoria da probabilidade de obter a ordem com Símbolo de Jacobi

Teorema 1. *Seja $N = p \cdot q$, onde p e q são números primos. Suponha ainda que $p = 2^c \cdot u + 1$ e $q = 2^c \cdot v + 1$, onde u e v são inteiros ímpares e c é uma constante. Para todo $x \in \mathbb{Z}_N^*$ tal que $\gcd(x, N) = 1$ e $x > 1$, vale que, se o símbolo de Jacobi de x em relação a N for igual a -1, então*

$$1 < \gcd(x^{r/2} - 1, N) < N.$$

Demonstração. Inicialmente, recorda-se que o símbolo de Jacobi de um inteiro x em relação a $N = p \cdot q$, com p e q primos ímpares, pode ser expresso como o produto dos símbolos de Legendre correspondentes, isto é,

$$\text{Jacobi}(x,N) = \text{Legendre}(x,p) \cdot \text{Legendre}(x,q).$$

Além disso, pela definição do símbolo de Legendre, tem-se que Legendre$(a,b) \in \{-1, 0, 1\}$. Portanto, se Jacobi$(x,N) = -1$, conclui-se que necessariamente ocorre um dos seguintes casos: Legendre$(x,p) = 1$ e Legendre$(x,q) = -1$, ou Legendre$(x,p) = -1$ e Legendre$(x,q) = 1$.

Sem perda de generalidade, considera-se o caso em que Legendre$(x,p) = 1$ e Legendre$(x,q) = -1$. Nessa situação, do fato de Legendre$(x,p) = 1$, segue que a ordem de x módulo p, denotada por $r_p = \text{ord}(x,p)$, divide $\frac{p-1}{2} = 2^{c-1}u$. Consequentemente, r_p não é divisível por 2^c, o que implica que $\gcd(2^c, r_p) < 2^c$.

Por outro lado, a condição Legendre$(x,q) = -1$ implica que a ordem de x módulo q, denotada por $r_q = \text{ord}(x,q)$, não divide $\frac{q-1}{2} = 2^{c-1}v$. Dessa forma, conclui-se que 2^c divide r_q, isto é, $\gcd(2^c, r_q) = 2^c$.

Considerando agora a ordem de x módulo N, denotada por $r = \text{ord}(x,N)$, e utilizando o fato de que $r = \text{lcm}(r_p, r_q)$ (ver Lema 2.3 em (Bastos; Kowada, 2021)), obtém-se que $\gcd(r, 2^c) = 2^c$.

Dessa relação, conclui-se que $r/2$ é múltiplo de r_p, mas não é múltiplo de r_q. Em termos de congruências, isso implica que

$$x^{r/2} \equiv 1 \pmod{p} \quad \text{e} \quad x^{r/2} \not\equiv 1 \pmod{q}.$$

Portanto, $x^{r/2} - 1$ é divisível por p, mas não é divisível por q. Como $N = p \cdot q$, segue que

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

o que implica diretamente que $1 < \gcd(x^{r/2} - 1, N) < N$.

Isso conclui a demonstração. □

APÊNDICE B – EXECUÇÕES DA SIMULAÇÃO

B.1 Execuções com frações continuadas

Tabela 9: Resultados para 5 medições

Sucesso	Qubits	Qubits original	Média tentativas	Desvio padrão	Máximo
Não	0.1	128	5	0	5
Não	0.1	256	5	0	5
Não	0.1	512	5	0	5
Não	0.2	128	5	0	5
Não	0.2	256	5	0	5
Não	0.2	512	5	0	5
Não	0.3	128	5	0	5
Não	0.3	256	5	0	5
Não	0.3	512	5	0	5
Não	0.4	128	5	0	5
Não	0.4	256	5	0	5
Não	0.4	512	5	0	5
Não	0.5	128	5	0	5
Não	0.5	256	5	0	5
Não	0.5	512	5	0	5
Não	0.60	128	5	0	1
Não	0.60	256	5	0	1
Não	0.60	512	5	0	1
Não	0.70	128	5	0	1
Não	0.70	256	5	0	1

Continua na próxima página

Sucesso	Qubits	Qubits original	Média tentativas	Desvio padrão	Máximo
Não	0.70	512	5	0	1
Não	0.80	128	5	0	1
Não	0.80	256	5	0	1
Não	0.80	512	5	0	1
Não	0.90	128	5	0	1
Não	0.90	256	5	0	1
Não	0.90	512	5	0	1
Sim	1.00	128	1.0	0	1
Sim	1.00	256	1.0	0	1
Sim	1.00	512	1.0	0	1

Tabela 10: Resultados para 10 medições

Sucesso	Qubits	Qubits original	Média tentativas	Desvio padrão	Máximo
Não	0.1	128	5	0	5
Não	0.1	256	5	0	5
Não	0.1	512	5	0	5
Não	0.2	128	5	0	5
Não	0.2	256	5	0	5
Não	0.2	512	5	0	5
Não	0.3	128	5	0	5
Não	0.3	256	5	0	5
Não	0.3	512	5	0	5
Não	0.4	128	5	0	5
Não	0.4	256	5	0	5
Não	0.4	512	5	0	5
Não	0.5	128	5	0	5
Não	0.5	256	5	0	5
Não	0.5	512	5	0	5
Não	0.60	128	5	0	1
Não	0.60	256	5	0	1
Não	0.60	512	5	0	1

Continua na próxima página

Sucesso	Qubits	Qubits original	Média tentativas	Desvio padrão	Máximo
Não	0.70	128	5	0	1
Não	0.70	256	5	0	1
Não	0.70	512	5	0	1
Não	0.80	128	5	0	1
Não	0.80	256	5	0	1
Não	0.80	512	5	0	1
Não	0.90	128	5	0	1
Não	0.90	256	5	0	1
Não	0.90	512	5	0	1
Sim	1.00	128	1.0	0	1
Sim	1.00	256	1.0	0	1
Sim	1.00	512	1.0	0	1

Tabela 11: Resultados para 40 medições

Sucesso	Qubits	Qubits original	Média tentativas	Desvio padrão	Máximo
Não	0.1	128	5	0	5
Não	0.1	256	5	0	5
Não	0.1	512	5	0	5
Não	0.2	128	5	0	5
Não	0.2	256	5	0	5
Não	0.2	512	5	0	5
Não	0.3	128	5	0	5
Não	0.3	256	5	0	5
Não	0.3	512	5	0	5
Não	0.4	128	5	0	5
Não	0.4	256	5	0	5
Não	0.4	512	5	0	5
Não	0.5	128	5	0	5
Não	0.5	256	5	0	5
Não	0.5	512	5	0	5
Não	0.60	128	5	0	1

Continua na próxima página

Sucesso	Qubits	Qubits original	Média tentativas	Desvio padrão	Máximo
Não	0.60	256	5	0	1
Não	0.60	512	5	0	1
Não	0.70	128	5	0	1
Não	0.70	256	5	0	1
Não	0.70	512	5	0	1
Não	0.80	128	5	0	1
Não	0.80	256	5	0	1
Não	0.80	512	5	0	1
Não	0.90	128	5	0	1
Não	0.90	256	5	0	1
Não	0.90	512	5	0	1
Sim	1.00	128	1.0	0	1
Sim	1.00	256	1.0	0	1
Sim	1.00	512	1.0	0	1

B.2 Execuções com Reticulado

Tabela 12: Resultados para 5 medições

Sucesso	Qubits	Qubits original	Média tentativas	Desvio padrão	Máximo
Não	0.1	128	5	0	5
Não	0.1	256	5	0	5
Não	0.1	512	5	0	5
Não	0.2	128	5	0	5
Não	0.2	256	5	0	5
Não	0.2	512	5	0	5
Não	0.3	128	5	0	5
Não	0.3	256	5	0	5
Não	0.3	512	5	0	5
Não	0.4	128	5	0	5
Não	0.4	256	5	0	5

Continua na próxima página

Sucesso	Qubits	Qubits original	Média tentativas	Desvio padrão	Máximo
Não	0.4	512	5	0	5
Não	0.5	128	5	0	5
Não	0.5	256	5	0	5
Não	0.5	512	5	0	5
Sim	0.6	128	1.50	0.67	2
Sim	0.6	256	1.44	0.95	2
Sim	0.6	512	1.28	0.45	2
Sim	0.7	128	1.0	0.0	1
Sim	0.7	256	1.0	0.0	1
Sim	0.7	512	1.0	0.0	1
Sim	0.8	128	1.0	0.0	1
Sim	0.8	256	1.0	0.0	1
Sim	0.8	512	1.0	0.0	1
Sim	0.9	128	1.0	0.0	1
Sim	0.9	256	1.0	0.0	1
Sim	0.9	512	1.0	0.0	1

Tabela 13: Resultados para 10 medições

Sucesso	Qubits	Qubits original	Média tentativas	Desvio padrão	Máximo
Não	0.1	128	5	0	5
Não	0.1	256	5	0	5
Não	0.1	512	5	0	5
Não	0.2	128	5	0	5
Não	0.2	256	5	0	5
Não	0.2	512	5	0	5
Não	0.3	128	5	0	5
Não	0.3	256	5	0	5
Não	0.3	512	5	0	5
Não	0.4	128	5	0	5
Não	0.4	256	5	0	5
Não	0.4	512	5	0	5

Continua na próxima página

Sucesso	Qubits	Qubits original	Média tentativas	Desvio padrão	Máximo
Não	0.5	128	5	0	5
Não	0.5	256	5	0	5
Não	0.5	512	5	0	5
Sim	0.60	128	1.0	0.0	1
Sim	0.60	256	1.0	0.0	1
Sim	0.60	512	1.0	0.0	1
Sim	0.70	128	1.0	0.0	1
Sim	0.70	256	1.0	0.0	1
Sim	0.70	512	1.0	0.0	1
Sim	0.80	128	1.0	0.0	1
Sim	0.80	256	1.0	0.0	1
Sim	0.80	512	1.0	0.0	1
Sim	0.90	128	1.0	0.0	1
Sim	0.90	256	1.0	0.0	1
Sim	0.90	512	1.0	0.0	1

Tabela 14: Resultados para 40 medições

Sucesso	Qubits	Qubits original	Média tentativas	Desvio padrão	Máximo
Não	0.1	128	5	0	5
Não	0.1	256	5	0	5
Não	0.1	512	5	0	5
Não	0.2	128	5	0	5
Não	0.2	256	5	0	5
Não	0.2	512	5	0	5
Não	0.3	128	5	0	5
Não	0.3	256	5	0	5
Não	0.3	512	5	0	5
Não	0.4	128	5	0	5
Não	0.4	256	5	0	5
Não	0.4	512	5	0	5
Não	0.5	128	5	0	5

Continua na próxima página

Sucesso	Qubits	Qubits original	Média tentativas	Desvio padrão	Máximo
Não	0.5	256	5	0	5
Não	0.5	512	5	0	5
Sim	0.60	128	1.0	0.0	1
Sim	0.60	256	1.0	0.0	1
Sim	0.60	512	1.0	0.0	1
Sim	0.70	128	1.0	0.0	1
Sim	0.70	256	1.0	0.0	1
Sim	0.70	512	1.0	0.0	1
Sim	0.80	128	1.0	0.0	1
Sim	0.80	256	1.0	0.0	1
Sim	0.80	512	1.0	0.0	1
Sim	0.90	128	1.0	0.0	1
Sim	0.90	256	1.0	0.0	1
Sim	0.90	512	1.0	0.0	1