

Síntese de Operadores Unitários Baseada em Gramáticas

José Alex de Carvalho¹, Adenilton José da Silva¹

¹Centro de Informática - Universidade Federal de Pernambuco (UFPE)
Recife - Pernambuco - Brasil

{jac, ajsilva}@cin.ufpe.br

Abstract. *Quantum algorithms propose a computational advantage in the resolution time for a range of problems. In this scenario, quantum software compilation consists of translating these high-level routines into low-level gates, defining a crucial step given the series of limitations faced by current quantum devices. The synthesis of quantum operators is based on recursive matrix decompositions, which makes its rewriting possible through the use of formal grammars. This work proposes the use of grammar rules in the unitary synthesis process, aiming to optimize the construction of the circuit's syntax tree, thereby saving CNOT gates.*

Resumo. *Algoritmos quânticos propõem uma vantagem computacional no tempo de resolução de uma gama de problemas. Nesse cenário, a compilação de software quântico consiste na tradução dessas rotinas de alto nível em portas de baixo nível, e define uma etapa crucial diante da série de limitações enfrentadas pelos dispositivos quânticos atuais. A síntese de operadores quânticos baseia-se em decomposições matriciais recursivas, o que torna possível a sua reescrita através do uso de gramáticas formais. Esse trabalho propõe o uso de regras gramaticais no processo de síntese de unitários, visando otimizar a construção da árvore sintática relativa ao circuito, poupando portas CNOT.*

1. Introdução

O processamento de informações por meio de dispositivos quânticos oferece vantagens computacionais teóricas significativas na resolução de uma ampla gama de problemas, permitindo uma redução drástica no tempo necessário para a execução de tarefas altamente custosas para os computadores clássicos atuais [Kim et al. 2023]. Dessa forma, a computação quântica desponta como uma promessa para o futuro do processamento de dados e uma alternativa que busca não suplantá-la, mas coexistir de forma cooperativa com a computação baseada em transistores de silício [Callison and Chancellor 2022].

Contudo, os dispositivos quânticos atuais ainda estão em um estágio incipiente de desenvolvimento. Com efeito, os processadores quânticos exibem hoje diversas limitações consideráveis, como o tempo de coerência (fator determinante para a quantidade de tempo em que as informações utilizadas no processo de computação podem ser retidas), a presença de ruído atrelado às operações realizadas por esses dispositivos e até mesmo a quantidade limitada de portas lógicas elementares implementáveis de modo funcional em cada uma das arquiteturas quânticas atuais. Esse conjunto de restrições define os computadores quânticos de hoje, os chamados dispositivos ruidosos de escala intermediária, ou NISQ (do inglês, Noisy Intermediate-Scale Quantum) [Preskill 2018].

Desse modo, algoritmos quânticos de alto nível precisam ser decompostos em rotinas básicas para a execução em computadores quânticos reais. Esses algoritmos, que podem ser descritos em termos de operadores matriciais unitários, precisam ser traduzidos para um conjunto reduzido de portas lógicas básicas implementáveis, usualmente operadores que atuam em um único bit quântico (ou qubit) e portas CNOT. Esse procedimento de síntese de circuitos quânticos e de tradução da linguagem das matrizes unitárias para a de representações intermediárias de baixo nível compõe a área da compilação de software quântico, onde encontrar decomposições matriciais que realizem as operações necessárias, exigindo um tempo de execução limitado, é a tarefa primordial.

Nesse contexto, as decomposições descritas pela álgebra linear tornam-se fundamentalmente relevantes. Decomposições como a decomposição cosseno-seno, a decomposição espectral e a decomposição em valores singulares, são o cerne de diversos trabalhos importantes e necessárias como sub-rotinas em diversos algoritmos quânticos [Shende et al. 2006, Harrow et al. 2009, Plesch and Brukner 2011]. É justamente o caráter recursivo desse tipo de decomposição que aproxima novamente a síntese de operadores unitários da teoria de compiladores e das linguagens formais. Decomposições recursivas remetem invariavelmente às gramáticas formais e à aplicação sucessiva de regras de derivação. Nesse trabalho, mostramos como a definição de regras gramaticais e a construção de árvores sintáticas são ferramentas úteis no processo da síntese de unitários, e como elas permitem simplificar a construção do circuito e fornecem mecanismos eficazes para a otimização do processo de compilação de software quântico. Adicionalmente, trazemos um exemplo comparativo do emprego da abordagem proposta, onde a comparamos com abordagens de estado da arte em uma instância da decomposição quântica de Shannon.

2. Síntese de Unitários

Uma das principais abordagens para a decomposição de operadores unitários na computação quântica é a Decomposição Quântica de Shannon (QSD, do inglês *Quantum Shannon Decomposition*) [Shende et al. 2006]. Essa abordagem consiste em uma decomposição recursiva do operador unitário de n qubits em um conjunto de rotações uniformemente controladas e operadores de $n - 1$ qubits, e possui um custo base total de $\frac{3}{4}4^n - \frac{3}{2}2^n$ portas CNOT, e um custo de $\frac{23}{48}4^n - \frac{3}{2}2^n + \frac{4}{3}$ portas CNOT através de otimizações para a implementação de um operador unitário de dois qubits [Shende et al. 2004, Shende et al. 2006]. Essa decomposição está esquematizada no circuito da Figura 1.

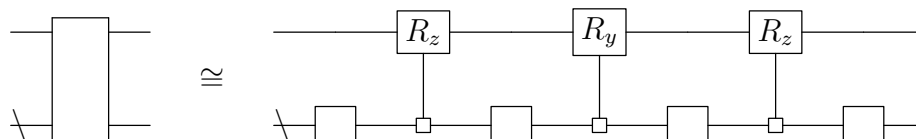


Figura 1. Circuito quântico referente à QSD recursiva.

Contudo, determinadas propriedades estruturais dessas matrizes podem permitir uma construção de circuito simplificada. Operadores representados por matrizes diagonais, por exemplo, podem ser implementados por meio da decomposição recursiva do

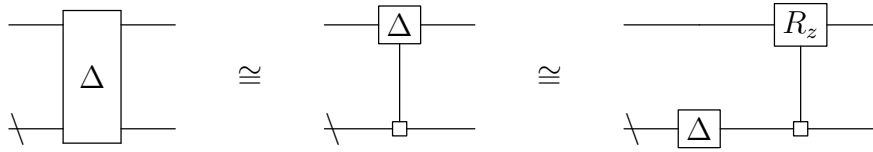


Figura 2. Circuito quântico referente à decomposição de operadores diagonais.

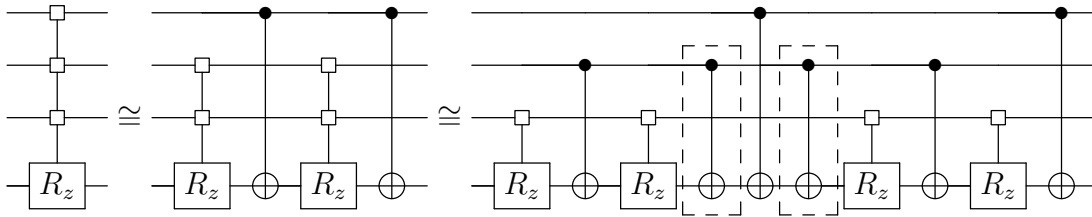


Figura 3. Circuito quântico referente à decomposição de rotações uniformemente controladas no eixo z , exemplificada para um conjunto de três qubits de controle e um de alvo. As portas CNOT destacadas podem ser removidas sem modificar o comportamento do circuito.

operador em uma nova porta diagonal e em rotações uniformemente controladas no eixo z da esfera de Bloch (UCRZs, do inglês *Uniformly Controlled Z-Rotation*), como expresso na Figura 2, com um custo de $2^{n+1} - 3$ portas elementares [Bullock and Markov 2004]. Operadores diagonais são comuns em diversos contextos na computação quântica, como na inicialização de estados [Bergholm et al. 2005] e na síntese de isometrias esparsas [Malvetti et al. 2021]. Outras classes de operadores com implementações simplificadas são as próprias rotações uniformemente controladas [Shende et al. 2006] com sua simplificação em circuitos separáveis [de Carvalho et al. 2025] ou os operadores unitários de dois níveis [Nielsen and Chuang 2010]. A Figura 3 exemplifica a decomposição de UCRZs com um custo de 2^k portas CNOTs, onde k é o número de bits de seleção, e um raciocínio semelhante é válido para rotações nos outros eixos. Além disso, a densidade da matriz (proporção de elementos nulos) permite decomposições de operadores esparsos com uma contagem menor de CNOTs, desde que certas restrições sejam cumpridas quanto ao número e à posição de elementos não nulos [Malvetti et al. 2021].

3. Síntese Baseada em Gramáticas

Dada a natureza recursiva dessas decomposições de operadores, é possível definir gramáticas livres de contexto (GLCs) cujas regras de produção são determinadas pelos passos dessa decomposição recursiva. Essas gramáticas definem uma linguagem cuja sintaxe determina a ordem de aplicação das portas quânticas. Tal linguagem representa circuitos quânticos e pode ser utilizada como uma ferramenta para a síntese desses circuitos. Por exemplo, a gramática da Figura 4 define a QSD, a gramática na Figura 5a representa a decomposição de portas diagonais, e a gramática em 5b representa a mesma situação para UCRZs.

Embora essas gramáticas sejam inerentemente ambíguas e apresentem recursão

```

<unitary>      ::= <csd>
                | u(num, num, num)
<csd>         ::= <qsd> ucry <qsd>
<qsd>         ::= <unitary> ucrz <unitary>
    
```

Figura 4. GLC representando a QSD para a síntese de operadores unitários genéricos. A variável `<unitary>` representa um operador quântico genérico, `<csd>` corresponde à decomposição cosseno-seno, parte importante da síntese de unitários e `<qsd>` é relacionada à etapa da decomposição quântica de Shannon propriamente dita. O terminal `u` representa um unitário de um qubit (parametrizado exceto por uma fase), `ucry` e `ucrz` representam operadores de rotação uniformemente controlados.

à esquerda, esse problema é contornado através da adição de um parâmetro n que corresponde ao número de qubits em que o operador atua, conferindo um caráter indutivo à derivação. Por exemplo, na decomposição de um operador de n qubits através da QSD, sempre surgem 4 operadores de $n - 1$ qubits. Portanto, a derivação sempre chega a um caso base que terá algum operador unitário de um qubit como símbolo terminal. Dessa forma, atribuímos atributos aos símbolos da gramática, associando regras semânticas às produções gramaticais. Assim, estruturamos o processo de síntese de unitários como uma tradução dirigida por sintaxe [Aho et al. 2006].

```

<diagonal>    ::= <diagonal> <first_ucrz>
                | p(num)
    
```

(a)

```

<first_ucrz>  ::= <ucrz> cx <ucrz> cx
                | rz(num) cx rz(num) cx
<ucrz>       ::= <ucrz> cx <ucrz>
                | rz(num) cx rz(num)
    
```

(b)

Figura 5. GLCs representando decomposições de operadores unitários com formato específico. (a) Operadores Diagonais - A variável `<diagonal>` representa um operador quântico diagonal. O terminal `p` representa um operador de fase, que implementa uma porta diagonal de um qubit exceto por uma fase. (b) UCRZs - A variável `<first_ucrz>` é a variável inicial representando a primeira UCRZ, `<ucrz>` representa UCRZs intermediárias, que permitem poupar portas CNOT. Os terminais `rz` e `cx` representam rotações em z e o operador CNOT, respectivamente.

Dessa forma, este trabalho propõe um *framework* de síntese otimizada de operadores unitários, utilizando regras de reescrita baseadas em gramáticas e árvores sintáticas (ASTs, do inglês Abstract Syntax Tree) anotadas para guiar o processo de decomposição de operadores e a geração de código OpenQASM, de modo semelhante ao trabalho de [Puschel et al. 2005] no contexto de transformadas e processamento digital de sinais. Através das produções dessas gramáticas, é possível construir um gerador de ASTs anotadas descendente recursivo guiado por predicados semânticos. Esse gerador decide qual

produção aplicar em cada nó por meio de um conjunto de funções de avaliação que verificam propriedades estruturais do operador, como diagonalidade ou esparsidade, de forma semelhante aos analisadores sintáticos com predicados semânticos, presentes no processo de compilação de diversas linguagens de propósito geral [Parr 2009]. Esse procedimento determina qual regra de produção é mais vantajosa para cada sub-operador, possibilitando uma redução na profundidade do circuito e no tempo total de compilação sem a necessidade de realizar uma busca exaustiva. Por fim, a geração do código OpenQASM é realizada através da visita da AST, traduzindo a estrutura da árvore e as anotações no conjunto de instruções correspondentes, finalizando o processo de tradução e síntese.

4. Avaliação e Discussão

Para exemplificar e avaliar a abordagem proposta, foi considerada uma instância específica da decomposição quântica de Shannon. Suponha que, durante a decomposição de um operador de n qubits, o operador do lado esquerdo da primeira etapa da decomposição (dada pela regra $\langle \text{csd} \rangle ::= \langle \text{qsd} \rangle \text{ucry} \langle \text{qsd} \rangle$ em Fig. 4) seja um operador de permutação geral, ou seja, um operador unitário com apenas um valor não nulo por coluna. Durante a construção da AST, o gerador sintático realizará a detecção da estrutura dessa matriz e procederá a geração da AST através das regras de produção da decomposição de operadores diagonais da Figura 5a. Essa abordagem é válida graças a possibilidade de fatorar o operador de permutação generalizada Π como $\Pi = \Delta \tilde{\Pi}$, onde Δ é um operador diagonal e $\tilde{\Pi}$ é um operador de permutação binário. $\tilde{\Pi}$ pode ser englobado pelo operador unitário do lado direito, desde que realizada uma reordenação dos operadores de rotação controlados, permitindo implementar Π como um operador diagonal. A árvore sintática referente a esse exemplo para um caso de três qubits está exemplificada na Figura 8.

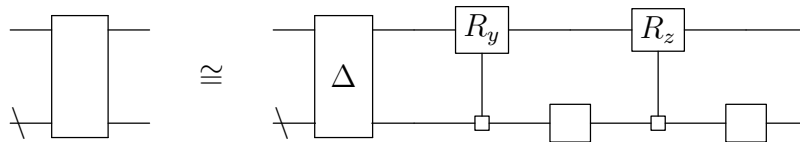


Figura 6. Circuito quântico referente à classe de operadores unitários otimizados no experimento proposto.

Para efeitos de comparação, avaliou-se a síntese de operadores com esse tipo de decomposição através do qiskit, SDK de código aberto mantido pela IBM e uma das ferramentas mais utilizadas na síntese e simulação de circuitos quânticos. Para os experimentos, gerou-se um operador unitário de n qubits através da multiplicação das matrizes de: a) um operador uniformemente controlado com um controle e operadores de $n - 1$ qubits atuando no alvo (o operador representado por $\langle \text{qsd} \rangle$ na gramática em Fig. 4); b) um operador de rotação uniformemente controlado em y com $n - 1$ controles; e c) um operador diagonal de n qubits, sendo todas as matrizes geradas de forma arbitrária. A figura 6 representa o circuito referente à decomposição desse tipo de operador. Dada a não-unicidade da decomposição cosseno-seno, ao se decompor a matriz gerada, pode-se obter um operador de permutação generalizada ao invés de um operador diagonal. Tal fato pode ser contornado utilizando a decomposição $\Pi = \Delta \tilde{\Pi}$ proposta anteriormente. Para isso, a inicialização desses operadores unitários foi testada em dois cenários: a) utilizando

a inicialização de unitários padrão da biblioteca, baseada em [Krol and Al-Ars 2024] (que propõe uma melhoria no fator constante da QSD, com um custo de $\frac{22}{48}4^n - \frac{3}{2}2^n + \frac{5}{3}$ portões CNOT); e b) identificando a presença do operador simplificável no primeiro nível da recursão, sintetizando uma porta diagonal em seu lugar. A Tabela 1 compila os resultados encontrados, comparando o número de CNOTs dos circuitos. Dada a redução no fator constante proposta por [Krol and Al-Ars 2024] e o uso de otimizações para a implementação de operadores de dois qubits [Shende et al. 2004, Shende et al. 2006], que não foram levados em consideração nos experimentos com a abordagem proposta, a síntese do qiskit possui uma leve vantagem para circuitos com poucos qubits, como evidente no caso com $n = 3$, mas cujo efeito assintótico se torna menos relevante a partir de $n = 4$.

Tabela 1. Comparação do número de portas CNOT obtidas com a síntese convencional do qiskit e com a abordagem proposta, onde n é o número de qubits.

n	Qiskit	Otimizado	Red.(%)	n	Qiskit	Otimizado	Red.(%)
2	2	2	0,0%	7	7.303	3.820	47,7%
3	18	20	-11,1%	8	29.591	15.144	48,8%
4	91	66	27,5%	9	119.287	60.330	49,4%
5	415	250	39,8%	10	478.871	240.812	49,7%
6	1.775	970	45,4%				

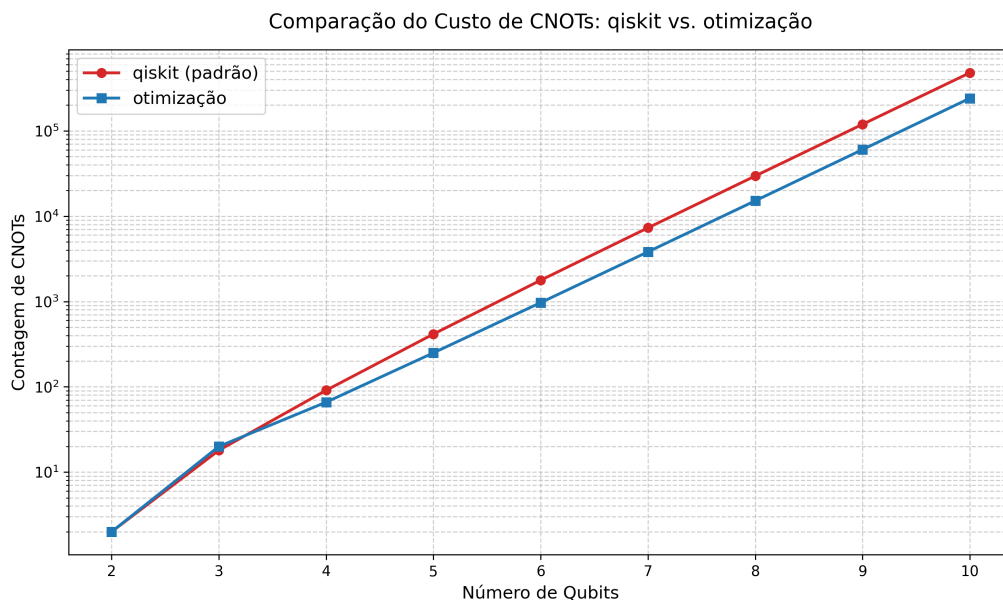
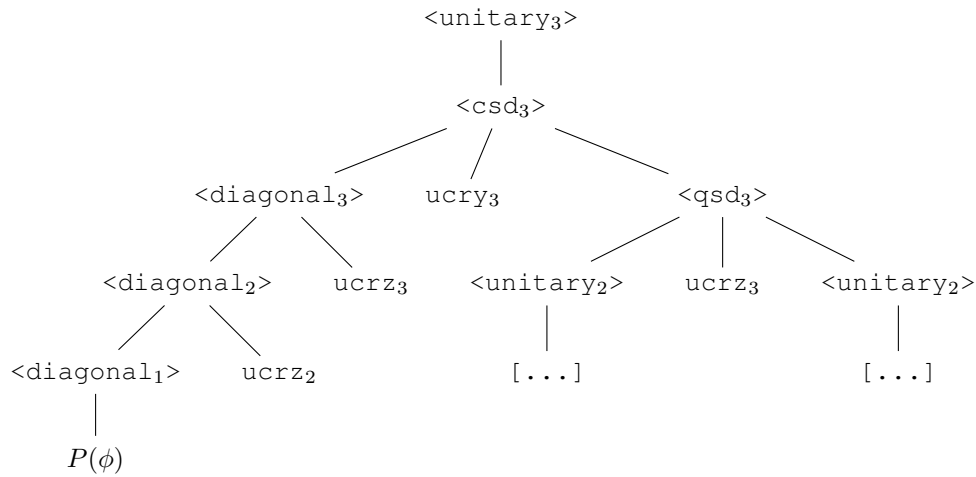
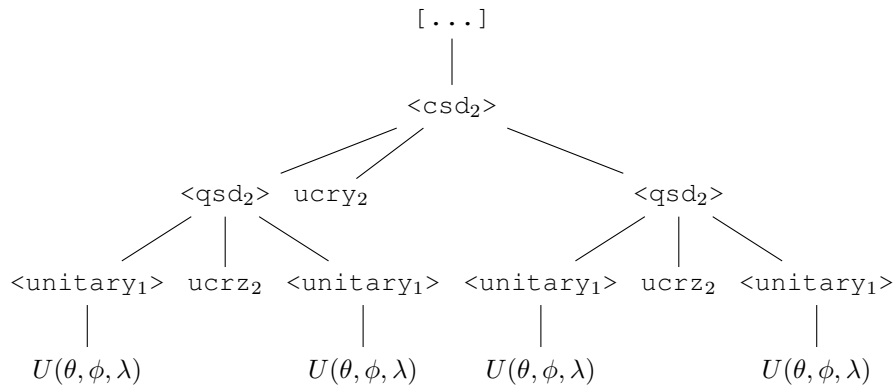


Figura 7. Comparação do número de portas CNOT evidenciando o comportamento assintótico das abordagens.

Pode-se notar que, através do emprego das regras de reescrita referentes à decomposição de operadores diagonais a partir da decomposição quântica de Shannon, conseguimos otimizar o processo de síntese a partir de $n = 4$ qubits, reduzindo a contagem de CNOTs para um valor que se aproxima assintoticamente de 50% do valor requerido para a implementação convencional, fato que pode ser aferido visualmente através



(a)



(b)

Figura 8. (a) Trecho da AST anotada resultante da aplicação das regras em Figura 4 e 5 (tratando rotações uniformemente controladas como variáveis finais) a partir de um operador unitário de 3 qubits em cuja decomposição apareça um operador de permutação generalizado. (b) Trecho da AST oculto em Figura 8b, correspondente a decomposição de um operador unitário de 2 qubits segundo as regras da QSD.

do gráfico da Figura 7. Evidentemente, a ferramenta de síntese proposta não se restringe apenas à busca por operadores diagonais, mas esse exemplo serve para mostrar o potencial da abordagem proposta em otimizar a síntese de operadores unitários em situações ainda pouco exploradas pelos compiladores de software quântico atuais.

5. Conclusão e Trabalhos Futuros

A abordagem de síntese otimizada de operadores quânticos baseada no uso de regras de produção gramaticais permite que simplificações sejam tomadas em cada nó da AST, otimizando a síntese e geração de código de representação intermediária do software quântico de forma que outras abordagens podem vir a negligenciar. Neste trabalho, mostramos como diversas estratégias de decomposição de circuitos quânticos podem ser expressas de formas simples e potencialmente úteis através do uso de gramáticas e construímos um exemplo que mostra a utilidade prática da abordagem.

Como expansão direta desse trabalho, temos a implementação do *framework* proposto utilizando uma linguagem de programação de alto nível e de alto desempenho como o C++, construindo um software de síntese otimizada de circuitos quânticos, seguindo uma arquitetura estruturada e as práticas de estado da arte da compilação de software. Adicionalmente, buscamos explorar as diferentes decomposições recursivas existentes na síntese de circuitos quânticos, formalizando-as através de gramáticas e identificando novas simplificações que possam reduzir o tempo de compilação e a profundidade dos circuitos em diferentes situações no contexto da computação quântica. Além disso, vamos considerar os impactos que a abordagem proposta pode ter na construção de software quântico sob a ótica das restrições de arquitetura e relacionadas às especificidades de implementações físicas de dispositivos quânticos. Em uma perspectiva de longo prazo, almejamos explorar as relações entre a síntese de circuitos quânticos baseada em gramáticas e a síntese baseada no cálculo ZX, assim como indagamos sobre as diferentes possibilidades de se conectar as gramáticas formais com as diferentes frentes da pesquisa atual em software de dispositivos quânticos.

Agradecimentos

O presente trabalho foi realizado com apoio da CAPES, CNPq, INCT de Computação Quântica Aplicada e o Instituto Quanta da UFPE.

Referências

- Aho, A. V., Lam, M. S., Sethi, R., and Ullman, J. D. (2006). *Compilers: Principles, Techniques, and Tools (2nd Edition)*. Addison Wesley.
- Bergholm, V., Vartiainen, J. J., Möttönen, M., and Salomaa, M. M. (2005). Quantum circuits with uniformly controlled one-qubit gates. *Physical Review A*, 71(5).
- Bullock, S. S. and Markov, I. L. (2004). Asymptotically optimal circuits for arbitrary n-qubit diagonal computations. *Quantum Inf. Comput.*, 4(1):27–47.
- Callison, A. and Chancellor, N. (2022). Hybrid quantum-classical algorithms in the noisy intermediate-scale quantum era and beyond. *Physical Review A*, 106(1).
- de Carvalho, J. A., Batista, C., de Veras, T., Araujo, I., and da Silva, A. J. (2025). Quantum multiplexer simplification for state preparation. *ACM Transactions on Quantum Computing*, 6(4):1–12.
- Harrow, A. W., Hassidim, A., and Lloyd, S. (2009). Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15).
- Kim, Y., Eddins, A., Anand, S., Wei, K. X., Van Den Berg, E., Rosenblatt, S., Nayfeh, H., Wu, Y., Zaletel, M., Temme, K., et al. (2023). Evidence for the utility of quantum computing before fault tolerance. *Nature*, 618(7965):500–505.
- Krol, A. M. and Al-Ars, Z. (2024). Beyond quantum shannon: Circuit construction for general n-qubit gates based on block zxz-decomposition.
- Malvetti, E., Iten, R., and Colbeck, R. (2021). Quantum circuits for sparse isometries. *Quantum*, 5:412.
- Nielsen, M. A. and Chuang, I. L. (2010). *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press.

- Parr, T. (2009). *Language implementation patterns: create your own domain-specific and general programming languages*. The Pragmatic Bookshelf.
- Plesch, M. and Brukner, Č. (2011). Quantum-state preparation with universal gate decompositions. *Physical Review A*, 83(3).
- Preskill, J. (2018). Quantum computing in the nisq era and beyond. *Quantum*, 2:79.
- Puschel, M., Moura, J. M., Johnson, J. R., Padua, D., Veloso, M. M., Singer, B. W., Xiong, J., Franchetti, F., Gacic, A., Voronenko, Y., et al. (2005). Spiral: Code generation for dsp transforms. *Proceedings of the IEEE*, 93(2):232–275.
- Shende, V., Bullock, S., and Markov, I. (2006). Synthesis of quantum-logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(6):1000–1010.
- Shende, V. V., Markov, I. L., and Bullock, S. S. (2004). Minimal universal two-qubit controlled-not-based circuits. *Physical Review A*, 69(6).