# Constructing Arbitrary Quantum Unitaries from Pauli Exponentials and Permutations

**Abstract**

We explore the construction of arbitrary unitary operators on $n$-qubit Hilbert spaces using exponentials of Pauli operators and permutation matrices. Building upon Lie algebraic completeness, we emphasize the role of Burnside's theorem in establishing irreducibility conditions that guarantee universality. We further introduce randomized schemes for approximating arbitrary unitaries through sequences of these building blocks. Connections to numerical linear algebra, including matrix factorizations and randomized algorithms, are highlighted, and potential applications in quantum computation and simulation are discussed.

## 1 Introduction

The ability to construct or approximate arbitrary unitary operations is foundational in quantum computing. Quantum circuits implement these unitaries through a finite set of universal gates.

We adopt a group-theoretic and Lie-algebraic viewpoint, analyzing unitaries acting on an $n$-qubit space $H = (\mathbb{C}^2)^{\otimes n}$ through:

- Permutation matrices, representing rearrangements of computational basis states.
- Exponentials of Pauli operators, generating rotations and entanglement via $\mathfrak{su}(N)$.

This formalism enables both exact and randomized decomposition schemes. We distinguish carefully between universality proofs, randomized circuit mixing, and compilation cost.

## 2 Preliminaries

Let $H = (\mathbb{C}^2)^{\otimes n}$ be the Hilbert space of $n$ qubits with dimension $N = 2^n$. The unitary group $U(N)$ acts on $H$. The set of Pauli strings $\mathcal{P}_n = \{I, X, Y, Z\}^{\otimes n}$ forms an orthogonal basis of $M_N(\mathbb{C})$.

**Lemma 1** (Pauli basis)**.** *The traceless Pauli strings form a real vector space of dimension $N^2 - 1$, isomorphic to the space of traceless Hermitian $N \times N$ matrices.*

**Proposition 1** (Lie algebra generation)**.** *The Lie algebra generated by $i$ times the traceless Pauli strings is $\mathfrak{su}(N)$. Exponentials of Pauli strings generate a subgroup dense in $SU(N)$.*

## 3 Burnside's Theorem and Universality

**Theorem 1** (Burnside)**.** *Any irreducible matrix algebra over $\mathbb{C}$ acting on $\mathbb{C}^N$ equals the full algebra $M_N(\mathbb{C})$.*

Since Pauli strings span $M_N(\mathbb{C})$, they act irreducibly. By Burnside's theorem, the generated algebra is $M_N(\mathbb{C})$, and exponentials of Pauli strings densely generate $SU(N)$.

**Remark 1.** *Permutations are not required for universality but help reduce circuit depth and adapt to hardware connectivity.*

## 4 Detailed Proofs and Examples

### 4.1 Pauli basis lemma

Each non-identity Pauli string is Hermitian, unitary, and traceless. Since $\dim(\mathfrak{su}(N)) = N^2 - 1 = 4^n - 1$, the traceless Pauli strings form a basis.

### 4.2 Lie algebra closure

The commutator $[\sigma_\alpha, \sigma_\beta]$ yields another Pauli string up to phase, ensuring closure. Thus their span under $i$ forms $\mathfrak{su}(N)$.

### 4.3 Example $n = 2$

For $n = 2$, $N = 4$. The 15 traceless Pauli strings span $\mathfrak{su}(4)$. Exponentials of these, together with SWAP (a permutation), generate $SU(4)$.

# 5 Randomized Approximation Schemes

## 5.1 Motivation

Exact decomposition of arbitrary $U$ may be inefficient. Randomized circuits provide probabilistic approximation.

## 5.2 Random circuits

Alternating random permutations and Pauli exponentials yield approximate unitary $t$-designs [**?**, **?**]. These approximate Haar measure and enable near-uniform sampling over $SU(N)$.

## 5.3 Algorithmic outline

1. Initialize $V_0 = I$.
2. For $j = 1$ to $k$:
   - Sample $P_j$ from $S_N$.
   - Sample $H_j = \sum_\alpha \theta_\alpha \sigma_\alpha$.
   - Update $V_j = P_j e^{iH_j} V_{j-1}$.
3. Stop when $V_k$ approximates target $U$ within tolerance.

Bounds on mixing depth depend polynomially on $n$ and design order $t$.

# 6 Connections to Numerical Linear Algebra

## 6.1 Matrix factorizations

The decomposition $U \approx \prod P_j e^{iH_j}$ resembles QR or Schur factorizations. Permutations act as pivoting steps; Pauli exponentials act like structured rotations.

## 6.2 Randomized algorithms

Analogous to randomized numerical linear algebra, random sampling of Pauli exponentials can reduce depth and improve approximation quality.

### 6.3 Error analysis

Tools from linear algebra quantify errors and stability of approximations. Burnside's theorem ensures no invariant subspaces impede convergence.

## 7 Implementation Costs

### 7.1 Decomposition of Pauli exponentials

An exponential of a weight-$w$ Pauli string decomposes into $O(w)$ two-qubit gates and one single-qubit rotation.

### 7.2 Trotter–Suzuki

For $H = \sum c_\alpha \sigma_\alpha$, approximate $e^{iHt}$ by

$$e^{iHt} \approx \Big( \prod_\alpha e^{ic_\alpha \sigma_\alpha t/m} \Big)^m,$$

with error $O(t^2/m)$. Higher-order Suzuki formulas improve accuracy.

### 7.3 Resource scaling

Local Hamiltonians with $O(n)$ terms of bounded weight are simulatable in polynomial resources, while dense Hamiltonians require exponential cost.

## 8 Applications and Outlook

- **Quantum compiling:** Efficient compilation of unitaries into hardware-native gates.
- **Quantum simulation:** Generation of random unitary ensembles for simulating complex systems.
- **Noise characterization:** Randomized sequences for benchmarking.
- **Quantum control:** Alternating permutations and Pauli exponentials yield flexible pulse sequences.

Future directions include tighter bounds on convergence, optimized decomposition strategies, and integration with hybrid algorithms.

# 9 References

## References


[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.

[2] B. C. Hall, *Lie Groups, Lie Algebras, and Representations*, Springer, 2015.

[3] C. Dankert, R. Cleve, J. Emerson, E. Livine, "Exact and approximate unitary 2-designs," Phys. Rev. A 80, 012304 (2009).

[4] F. Brandão, A. Harrow, M. Horodecki, "Local random quantum circuits are approximate polynomial-designs," Commun. Math. Phys. 346, 397–434 (2016).

[5] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 2012.

[6] M. W. Mahoney, "Randomized algorithms for matrices and data," Foundations and Trends in Machine Learning, 3(2):123–224, 2011.