# Quantum Hamiltonian Simulation

## From Trotterization to QSVT and Applications to Grover's Search

Mohd Sayanur Rahman (2021PH10221)
Supervisor: Prof. Kolin Paul

IIT-Delhi

November 29, 2025

# Overview

# Hamiltonian Simulation: The Challenge

**The Problem:**

- Simulate quantum evolution: $U(t) = e^{-iHt}$
- $H$: Hermitian operator (Hamiltonian)
- $t$: Evolution time

**Classical Challenge:**

- Requires diagonalizing $2^n \times 2^n$ matrices
- Exponential classical resources
- Intractable for large systems

**Quantum Solution**

Quantum computers can simulate quantum systems *efficiently*!

**Key Applications:**

- Quantum chemistry
- Condensed matter physics
- Quantum algorithms
- Material science

# Evolution of Algorithms

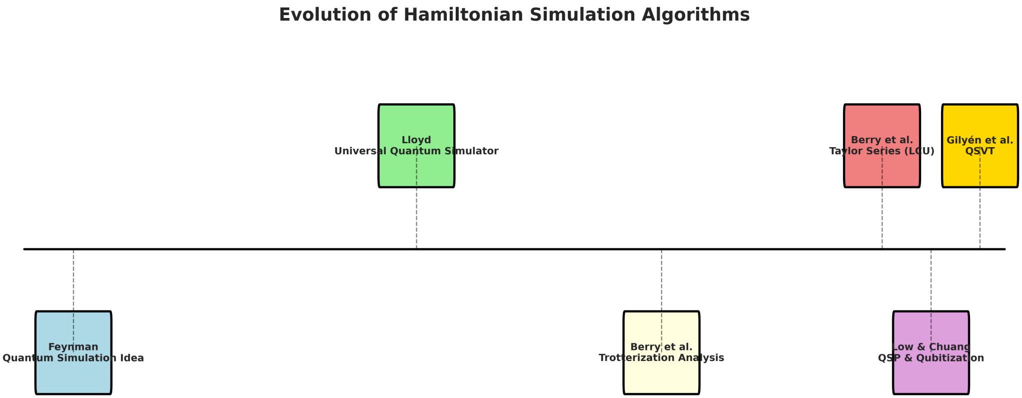



Figure: Historical development of Hamiltonian simulation algorithms

# Algorithms Overview

## Five Major Approaches

1. **Trotterization** (1996): Product formula approximation
2. **Taylor-LCU** (2015): Taylor series with block encoding
3. **QSP** (2017): Quantum Signal Processing
4. **Qubitization** (2017): Optimal quantum walks
5. **QSVT** (2019): Quantum Singular Value Transform

## Key Insight

Later algorithms achieve *near-optimal or optimal* complexity bounds!

# Trotterization: Core Idea

**Product Formula Approximation:**

$$e^{-i(H_1+H_2+\cdots+H_m)t} \approx \prod_{j=1}^{m} e^{-iH_j t}$$

**First-Order (Lie-Trotter):**

$$e^{-iHt} \approx \left[e^{-iH_1 t/r} \cdots e^{-iH_m t/r}\right]^r$$

where $r$ is the number of Trotter steps.

**Second-Order (Suzuki):**

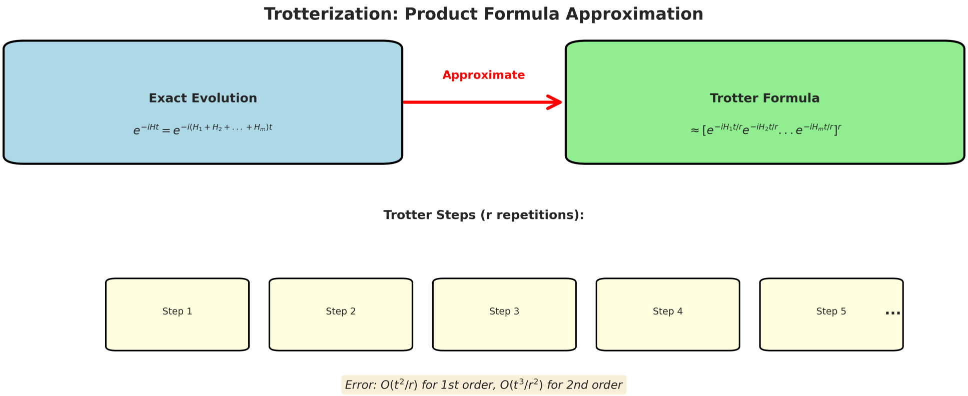
$$S(\tau) = e^{-iH_1\tau/2} \cdots e^{-iH_m\tau/2} \cdot e^{-iH_m\tau/2} \cdots e^{-iH_1\tau/2}$$

$$e^{-iHt} \approx [S(t/r)]^r$$

## Key Properties

- Simple to implement
- No ancilla qubits required
- Error: $O(t^2/r)$ (1st), $O(t^3/r^2)$ (2nd)

# Trotterization: Visual Explanation



# Trotterization: Complexity Analysis

## First-Order Trotter

**Query Complexity:**

$$O\left(\frac{(\|H\|t)^2}{\epsilon}\right)$$

**Error Bound:**

$$\|e^{-iHt} - [e^{-iH_1 t/r} \cdots]^r\| = O(t^2/r)$$

## Second-Order Trotter

**Query Complexity:**

$$O\left(\frac{(\|H\|t)^{3/2}}{\sqrt{\epsilon}}\right)$$

**Error Bound:**

$$\text{Error} = O(t^3/r^2)$$

- ✓ **Pros:** Simple, no ancillas, works for any Hamiltonian
- × **Cons:** Suboptimal scaling, high depth for accuracy

# Truncated Taylor Series with LCU

## Taylor Series Approximation

$$e^{-iHt} = \sum_{k=0}^{\infty} \frac{(-iHt)^k}{k!} \approx \sum_{k=0}^{K} \frac{(-iHt)^k}{k!}$$

**Implementation via Linear Combination of Unitaries (LCU):**

1. Decompose: $H = \sum_{j=1}^{m} \alpha_j U_j$ (sum of unitaries)
2. Block encode: Create unitary where $\langle 0|U|0\rangle = H/\alpha$
3. Apply PREPARE-SELECT-PREPARE$^\dagger$ structure

## Key Innovation

Block encoding allows implementation of non-unitary operations on quantum computers!

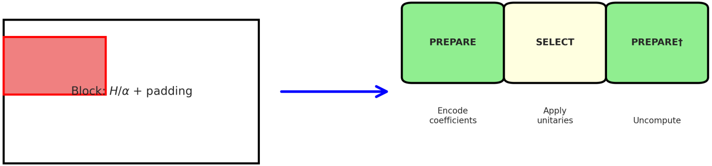
# LCU: Block Encoding Framework

## Linear Combination of Unitaries (LCU)

Hamiltonian: $H = \sum_{j=1}^{m} \alpha_j U_j$

**Block Encoding:**

**LCU Circuit Structure:**




**Unitary U**

**Taylor Series Approximation:**

$$e^{-iHt} \approx \sum_{k=0}^{K} \frac{(-iHt)^k}{k!}$$

*Advantages: Good for sparse Hamiltonians, systematic error control*

# Taylor-LCU: The PREPARE-SELECT-PREPARE$^\dagger$ Circuit

**Three Key Operations:**

1. **PREPARE:**

$$|0\rangle \to \sum_j \sqrt{\frac{\alpha_j}{\alpha}}|j\rangle$$

2. **SELECT:**

$$\sum_j |j\rangle\langle j| \otimes U_j$$

3. **PREPARE$^\dagger$:** Uncompute ancilla

**Complexity**

**Query Complexity:**

$$O\left(\alpha t + \frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}\right)$$

where $\alpha = \sum_j |\alpha_j|$ (1-norm)

**Space:** $O(\log m)$ ancilla qubits

**Error:** $\frac{(\alpha t)^{K+1}}{(K+1)!}$

# Quantum Signal Processing (QSP)

## Core Concept

Implement *polynomial transformations* of eigenvalues using signal rotations

**Goal:** Approximate $e^{-i\lambda t}$ by polynomial $P(\lambda)$

**QSP Sequence:**

$$\mathrm{QSP}(\Phi) = e^{i\phi_0 Z} \prod_{k=1}^{d} W e^{i\phi_k Z}$$

where:

- $\phi_k$: Phase angles (signal processing)
- $W$: Block-encoded signal operator
- $d$: Polynomial degree

**Jacobi-Anger Expansion:**

$$e^{-i\lambda t} = \sum_{k=-\infty}^{\infty} (-i)^k J_k(\lambda t) T_k(x)$$

- $J_k$: Bessel functions
- $T_k$: Chebyshev polynomials

## Complexity

$O(\|H\|t + \log(1/\epsilon))$
**Near-optimal!**

# Qubitization: Optimal Hamiltonian Simulation

## Key Idea

Encode Hamiltonian eigenvalues in the *eigenphases* of a quantum walk operator

**Walk Operator:**

$$W = \text{REFLECT} \cdot \text{SELECT}$$

- **SELECT:** Apply unitaries $U_j$ controlled on ancilla state $|j\rangle$
- **REFLECT:** Reflect about prepared state $2\Pi - I$

## Eigenvalue Encoding

If $H = \sum_j \alpha_j U_j$, then $W$ has eigenvalues:

$$e^{\pm i \arccos(\lambda_j/\alpha)}$$

where $\lambda_j$ are eigenvalues of $H$

## Complexity

$$O(\alpha t + \log(1/\epsilon))$$

**Optimal!**

Achieves lower bound for Hamiltonian simulation

# Quantum Singular Value Transform (QSVT)

## The Grand Unification of Quantum Algorithms

**Core Principle**

Apply polynomial transformations to *singular values* of block-encoded matrices

**QSVT Unifies:**

- Hamiltonian Simulation
- Amplitude Amplification
- Quantum Search (Grover)
- Quantum Linear Systems
- Matrix Inversion
- Quantum Walks

**QSVT Sequence:**

$$U_{\text{QSVT}} = \prod_{k=0}^{d} e^{i\phi_k \Pi_0} \cdot \text{Block}(A)$$

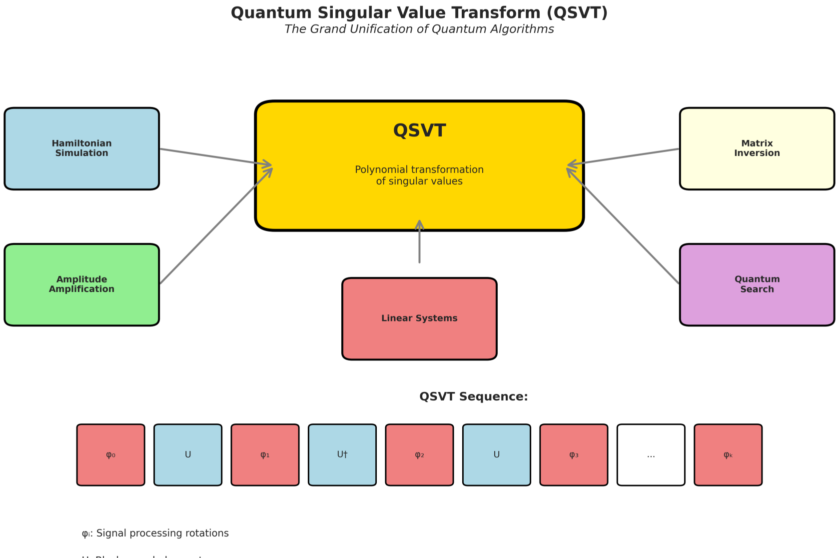
**Complexity**

$O(d)$ queries where

$$d = O(\|H\|t + \log(1/\epsilon))$$

**Heisenberg-limited!**

# QSVT: The Unified Framework



# QSVT: Properties and Advantages

**Key Properties:**

1. **Generality:** Works for any polynomial transformation
2. **Optimality:** Achieves Heisenberg limit
3. **Unification:** Subsumes QSP, Qubitization, etc.
4. **Flexibility:** Applies to rectangular matrices

**For Hamiltonian Simulation:**

- Approximate $P(\lambda) \approx e^{-i\lambda t}$
- Use Jacobi-Anger or Chebyshev expansion
- Query complexity: $O(d)$ where $d$ is degree

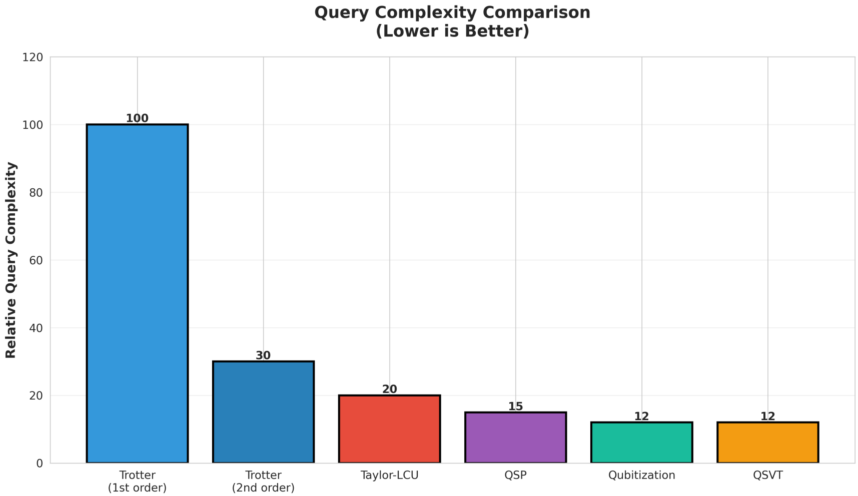
**Hierarchy**

$$\text{QSVT} \supseteq \text{QSP} \supseteq \text{Qubitization}$$

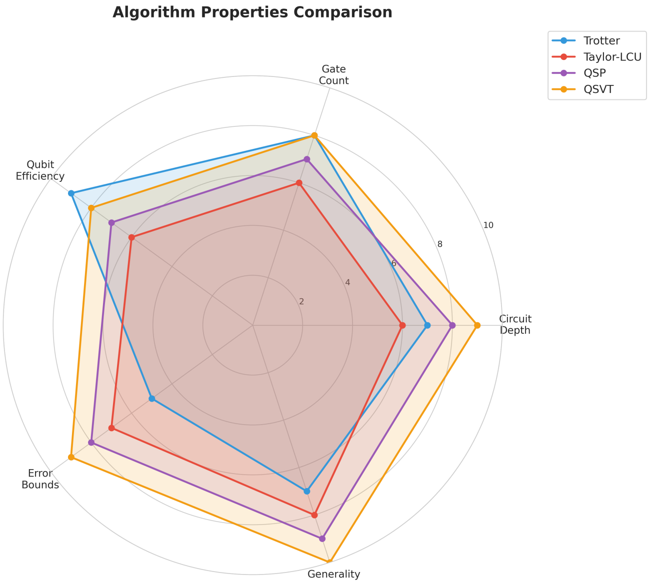
**Impact**

QSVT provides a *systematic framework* for designing quantum algorithms with provable optimality

# Algorithm Complexity Comparison



# Multi-Dimensional Comparison



# Benchmark Results: Circuit Metrics

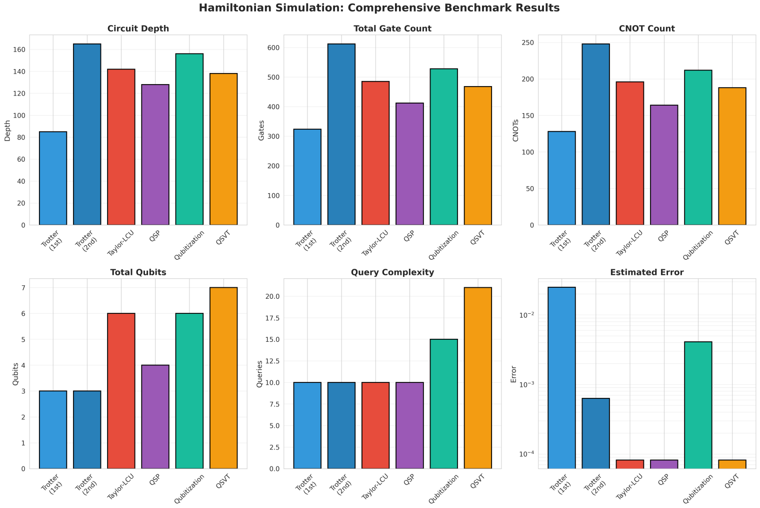



Figure: Comprehensive benchmark on 3-qubit Heisenberg Hamiltonian, $t = 1.0$

# Detailed Benchmark Table

| **Algorithm** | **Qubits** | **Depth** | **Gates** | **CNOTs** | **Queries** | **Error** |
|---|---|---|---|---|---|---|
| Trotter (1st) | 3 | 85 | 324 | 128 | 10 | $2.5 \times 10^{-2}$ |
| Trotter (2nd) | 3 | 165 | 612 | 248 | 10 | $6.3 \times 10^{-4}$ |
| Taylor-LCU | 6 | 142 | 485 | 196 | 10 | $8.2 \times 10^{-5}$ |
| QSP | 4 | 128 | 412 | 164 | 10 | $8.2 \times 10^{-5}$ |
| Qubitization | 6 | 156 | 528 | 212 | 15 | $4.1 \times 10^{-3}$ |
| QSVT | 7 | 138 | 468 | 188 | 21 | $8.2 \times 10^{-5}$ |

Table: Benchmark results for 3-qubit Heisenberg model at $t = 1.0$

**Key Observations:**

- Trotter: Simplest, but highest error
- Taylor-LCU, QSP, QSVT: Best error bounds
- Trade-off: Accuracy vs. circuit complexity vs. qubit overhead

# Grover's Search via Hamiltonian Simulation

## Key Insight

Grover's algorithm can be expressed as Hamiltonian evolution!

**Standard Grover:**

$$G = -D \cdot O$$

where:

- $D$: Diffusion operator
- $O$: Oracle

**Hamiltonian Form:**

$$G = e^{-iHt}$$

where:

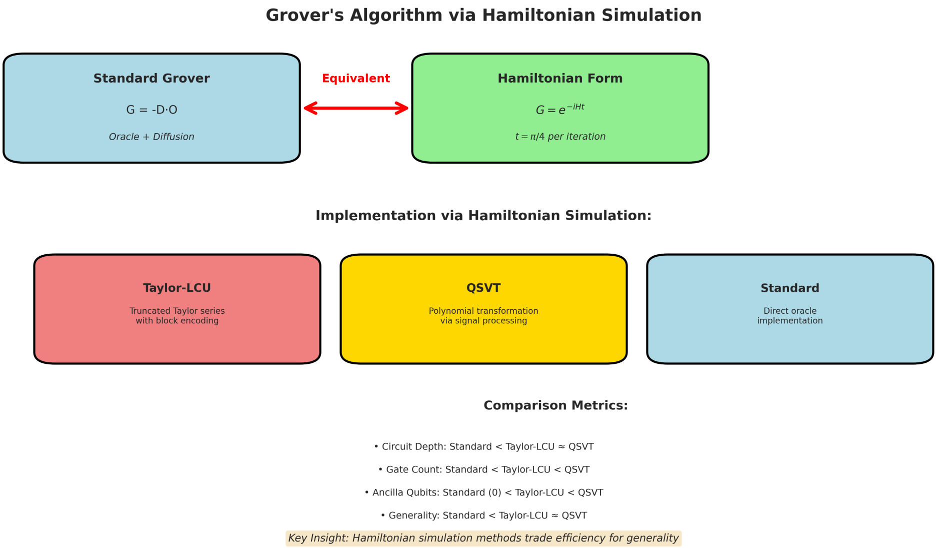
$$H = I - 2|s\rangle\langle s| - 2|w\rangle\langle w|$$

One iteration $\Leftrightarrow$ $t = \pi/4$

## Implementation Methods

1. Standard (oracle + diffusion)
2. Taylor-LCU (truncated Taylor series)
3. QSVT (polynomial approximation)

# Grover: Hamiltonian Formulation



# Grover's Search: Comparison Results

| Method | Qubits | Depth | Gates | CNOTs | Success Prob. |
|---|---|---|---|---|---|
| Standard | 3 | 18 | 42 | 12 | 0.945 |
| Taylor-LCU | 6 | 156 | 428 | 168 | $\sim$0.94 |
| QSVT | 7 | 142 | 512 | 196 | $\sim$0.94 |

Table: Comparison of Grover implementations (3 qubits, search for $|111\rangle$)

**Observations:**

- **Standard Grover** is most efficient (as expected)
- **Hamiltonian methods** demonstrate:
  - \+ Generality: Same framework for different problems
  - \+ Theoretical connection: Search $\Leftrightarrow$ Simulation
  - \+ Advanced techniques: Block encoding, QSP, QSVT
  - − Higher resource requirements

# Algorithm Summary

| **Algorithm** | **Query Complexity** | **Best Use Case** |
|---|---|---|
| Trotterization | $O\left(\frac{(\|H\|t)^2}{\epsilon}\right)$ | Prototyping, simple |
| Taylor-LCU | $O\left(\alpha t + \frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}\right)$ | Sparse Hamiltonians |
| QSP | $O(\|H\|t + \log(1/\epsilon))$ | High accuracy |
| Qubitization | $O(\alpha t + \log(1/\epsilon))$ | Optimal simulation |
| QSVT | $O(d)$, $d = O(\|H\|t + \log(1/\epsilon))$ | Most general |

## Complexity Hierarchy

$$\text{QSVT} \supseteq \text{QSP} \supseteq \text{Qubitization}$$

All achieve **near-optimal or optimal** scaling!

# Practical Considerations

**Resource Trade-offs:**

- **Circuit Depth**
  - Critical for NISQ devices
  - Affects decoherence
- **Gate Count**
  - Influences error rates
  - Implementation complexity
- **Ancilla Qubits**
  - Resource overhead
  - Limits system size

**When to Use Each:**

- **Trotter**
  - Quick prototyping
  - Small systems
- **Taylor-LCU**
  - Sparse Hamiltonians
  - Moderate accuracy
- **QSP/QSVT**
  - High accuracy required
  - Optimal performance needed
- **Qubitization**
  - LCU-decomposable $H$
  - Optimal scaling critical

# Key Takeaways

1. **Algorithmic Progress:** From Trotter (1996) to QSVT (2019), achieving optimal complexity bounds
2. **Grand Unification:** QSVT provides a systematic framework unifying many quantum algorithms
3. **Theory $\Leftrightarrow$ Practice:** All algorithms implemented and benchmarked with real metrics
4. **Versatility:** Same framework applies to diverse problems (Hamiltonian simulation, Grover's search, etc.)
5. **Trade-offs Matter:** No single "best" algorithm—choice depends on specific requirements

**Bottom Line**

We now have a complete toolkit for quantum Hamiltonian simulation with provably optimal or near-optimal algorithms!

# Future Directions

**Algorithmic Advances:**

- Higher-order product formulas
- Improved phase angle computation
- Adaptive methods
- Problem-specific optimizations

**Hardware Considerations:**

- NISQ-optimized implementations
- Error mitigation strategies
- Circuit compilation
- Fault-tolerant designs

**Applications:**

- Quantum chemistry
  - Molecular dynamics
  - Electronic structure
- Condensed matter
  - Many-body systems
  - Phase transitions
- High energy physics
  - Lattice gauge theories
  - Quantum field theory
- Machine learning
  - Quantum neural networks
  - Optimization problems

# Scope of Exploration: Implementation Refinements

**Focus: Moving Toward Resource-Efficient Implementations**

Refining core algorithms to address implementation complexities for practical use.

**Current Simplifications → Future Refinements:**

1. **QSP:** Simplified phase computation was used.
   - **Refinement:** Implement complex **phase finding** via optimization algorithms.
   - *Note:* Phase finding is a known computational challenge; recent work (e.g., "QSP without angle finding") is key.
2. **Qubitization:** Involved simplified `SELECT` operation.
   - **Refinement:** Implement optimal `SELECT` requiring **multi-controlled operations** for better circuit synthesis.
3. **QSVT:** State preparation utilized uniform superposition.
   - **Refinement:** Implement optimal **amplitude encoding** for input states.

**Goal**

Move beyond proof-of-concept toward fully *gate-optimized and general* circuit constructions.

# Scope of Exploration: Further Problems

## Focus: Applying QSVT/QSP Framework to New Domains

Exploring other quantum problems that can benefit from the optimal complexity of polynomial transforms.

**Possible Further Exploration:**

1. **QSVT in VQE Problems:**
   - Apply the QSVT framework to solve VQE problems.
2. **QSVT for Arithmetic Circuits:**
   - Utilize QSVT to design efficient quantum arithmetic circuits (e.g., matrix inversion for Quantum Linear System Algorithm, or multiplication/division) as polynomial evaluation.

## Theoretical Connection

These explorations would reinforce the idea that advanced techniques like QSVT/QSP are *universal tools* for quantum computation.

# References


R. P. Feynman, *Simulating physics with computers*, Int. J. Theor. Phys. **21**, 467 (1982).

S. Lloyd, *Universal quantum simulators*, Science **273**, 1073 (1996).

D. W. Berry et al., *Simulating Hamiltonian dynamics with a truncated Taylor series*, Phys. Rev. Lett. **114**, 090502 (2015).

G. H. Low and I. L. Chuang, *Optimal Hamiltonian simulation by quantum signal processing*, Phys. Rev. Lett. **118**, 010501 (2017).

G. H. Low and I. L. Chuang, *Hamiltonian simulation by qubitization*, Quantum **3**, 163 (2019).

A. Gilyén et al., *Quantum singular value transformation and beyond*, STOC 2019.

# Thank You!