Quantum Integration Engine: Analysis and Benchmark Report

Abstract

We present a generalized quantum algorithm designed to calculate the definite integral of arbitrary mathematical functions. By utilizing Iterative Quantum Amplitude Estimation (IQAE) and Chebyshev polynomial approximations, the implementation demonstrates a quadratic speedup over classical Monte Carlo methods. We experimentally verify that the quantum approach achieves a query complexity of $O(1/\epsilon)$. Furthermore, we identify critical applications for this engine in financial portfolio optimization and combinatorial search problems. For the complete rigorous proof of the algorithm's convergence and error bounds, please refer to the attached document `proof.pdf`.

1 Introduction

Definite integration is a fundamental mathematical operation with wide-ranging applications in physics, finance, and engineering. While classical numerical integration methods, such as Monte Carlo integration, are robust, they often suffer from slow convergence rates. To reduce the estimation error by a factor of 10, classical Monte Carlo methods typically require a 100-fold increase in sample size.

This project provides an end-to-end framework for quantum integration. It accepts a continuous, user-defined classical function, approximates it using polynomials, and encodes it into a quantum oracle. The algorithm runs on a quantum simulator (using Qiskit) to calculate the integral with user-specified precision. The primary objective is the experimental verification of the quadratic quantum speedup [1].

2 Algorithms and Complexity

2.1 Integral to Probability Mapping

The core mechanism leverages a mapping between classical integration and quantum probability estimation. Consider the definite integral I:

$$I = \int_a^b f(x)dx \tag{1}$$

This can be rewritten in terms of the expectation value (probability a) of a scaled function $p_{scaled}(x)$. Let m and M be the minimum and maximum values of $f(x)$ on the domain $[a, b]$. We define the scaled function as:

$$p_{scaled}(x) = \frac{f(x) - m}{M - m} \tag{2}$$

The true integral I is related to the probability a (the average value of the scaled function) by the linear transformation:

$$I = a \cdot (b - a)(M - m) + m(b - a) \tag{3}$$

2.2 State Preparation

We prepare a quantum state on $n + 1$ qubits. The first n qubits encode the integration domain via a uniform superposition, created by applying Hadamard gates to the zero state:

$$|\psi\rangle_n = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle \tag{4}$$

An oracle operator is then applied to rotate an ancilla qubit such that the probability of measuring it in the $|1\rangle$ state is equal to the target probability a.

2.3 Quantum Oracle and Chebyshev Approximation

To support arbitrary mathematical functions, we employ a **Chebyshev polynomial approximation**. This method minimizes Runge's phenomenon and provides a near-optimal polynomial representation with uniformly distributed error across the interval.

The coefficients of this polynomial are loaded into a `PolynomialPauliRotations` gate. This forms the state preparation circuit $\mathcal{A}$, which includes the initial Hadamard transforms and the polynomial oracle (Figure 1).

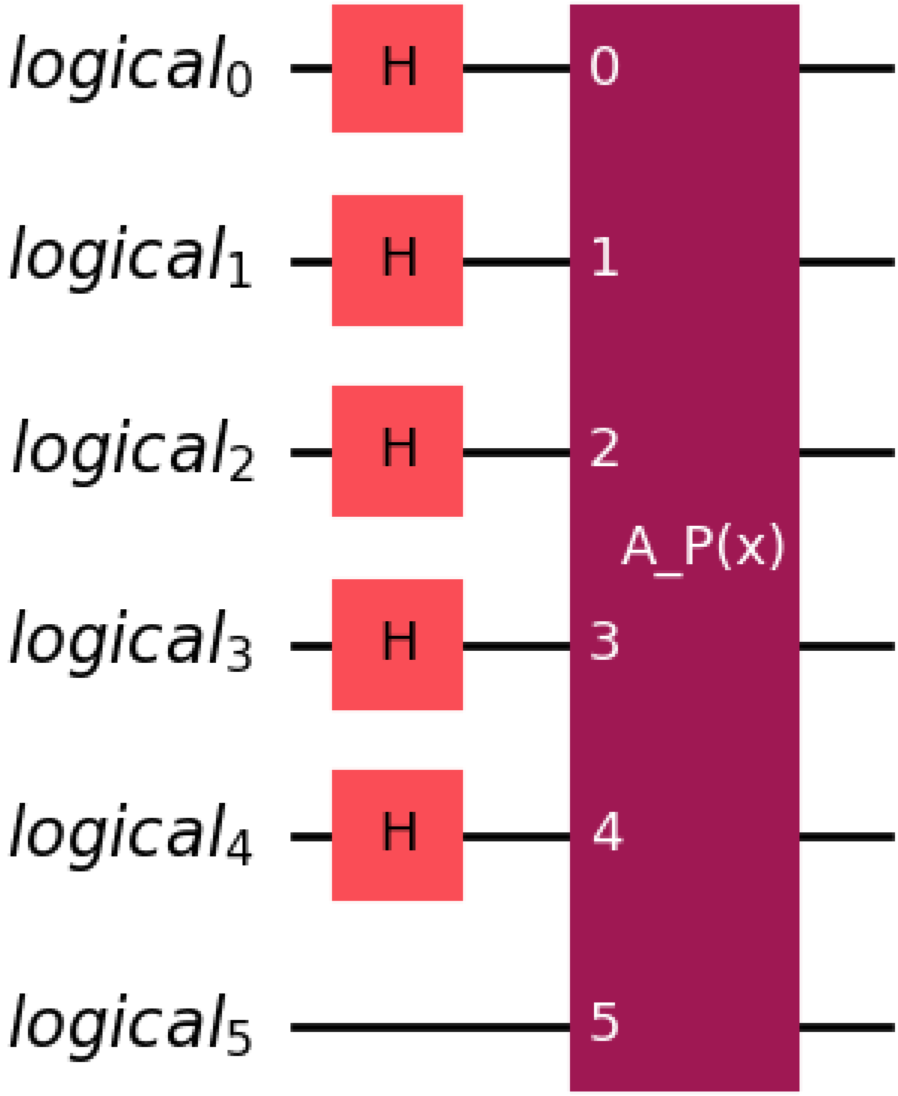

Figure 1: State preparation circuit diagram, showing the loading of polynomial coefficients into the quantum state.

2.4 Complexity Bounds

The performance difference between the classical and quantum approaches is defined by their query complexity relative to the desired precision ϵ.

- **Classical Monte Carlo:** To estimate the integral with precision ϵ, the number of samples $M_{classical}$ scales according to the variance:

$$M_{classical} \approx O\left(\frac{1}{\epsilon^2}\right) \tag{5}$$

- **Quantum Amplitude Estimation (IQAE):** The quantum algorithm utilizes the Grover operator $\mathcal{Q}$ to amplify the probability amplitude. The convergence rate is quadratically faster [2]:

$$M_{quantum} \approx O\left(\frac{1}{\epsilon}\right) \tag{6}$$

- **Circuit Depth:** For a polynomial approximation of degree d implemented on n qubits, the circuit depth of the oracle scales as:

$$D_{circuit} \approx O(d \cdot n^2) \tag{7}$$

This scaling ensures that for low-degree polynomial approximations, the quantum circuit remains efficient enough for near-term execution.

3 Experimental Workflow

The benchmark is implemented in Python 3.11, utilizing Qiskit 2.2.1 for circuit construction and simulation.

3.1 Approximation Verification

Before quantum execution, the system validates the polynomial approximation. As shown in Figure 2, the fitted polynomial closely tracks the target function (e.g., $f(x) = \log(x)$) within the scaled domain.

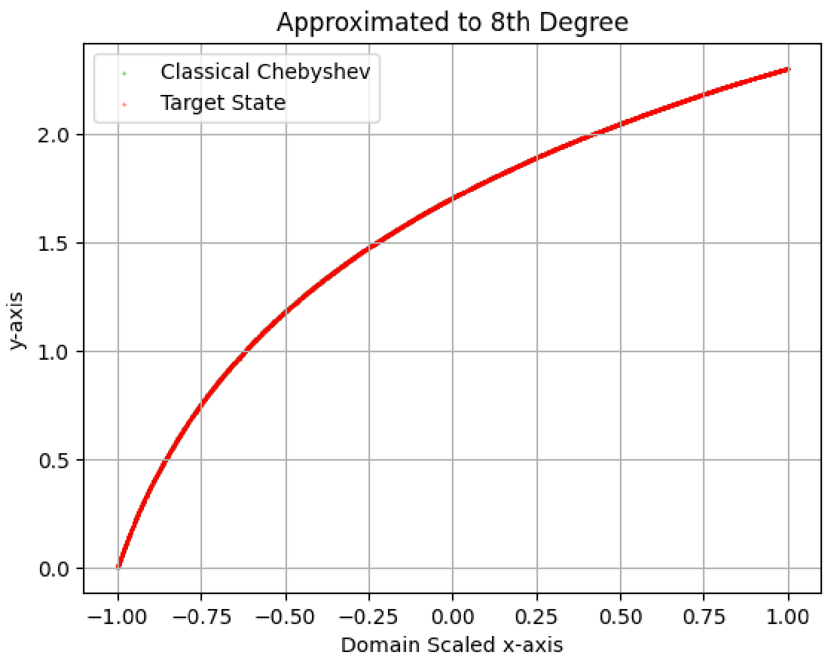

Figure 2: Chebyshev approximation of $\log(x)$ in the domain $[1, 10]$, scaled to $[-1, 1]$.

The corresponding error is minimal and uniformly distributed across the integration interval, as demonstrated in Figure 3.

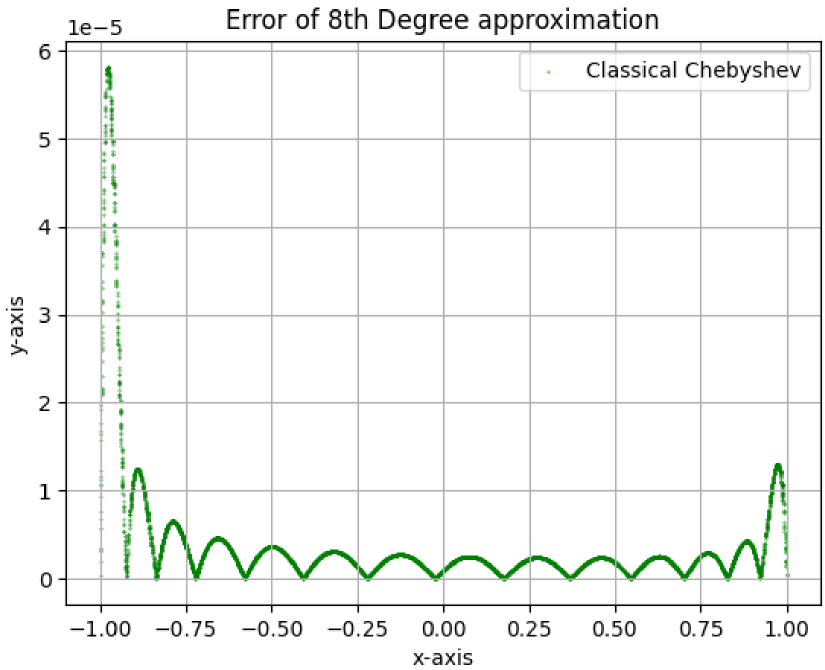

Figure 3: Error distribution of the Chebyshev approximation across the interval.

4 Results

We conducted experiments to compare the query counts required by Classical Monte Carlo versus Quantum Amplitude Estimation to reach a target error rate.

4.1 Quadratic Speedup Verification

The results confirm the theoretical predictions:

1. **Classical Scaling:** The classical method follows a steep curve proportional to $1/\epsilon^2$. As the precision requirement tightens, the computational cost explodes.
2. **Quantum Scaling:** The quantum method maintains a linear relationship proportional to $1/\epsilon$.

Figure 4 illustrates this divergence. For the function x^2 on domain $[0, 9]$, the quantum algorithm requires significantly fewer queries than the classical method to achieve the same level of accuracy.

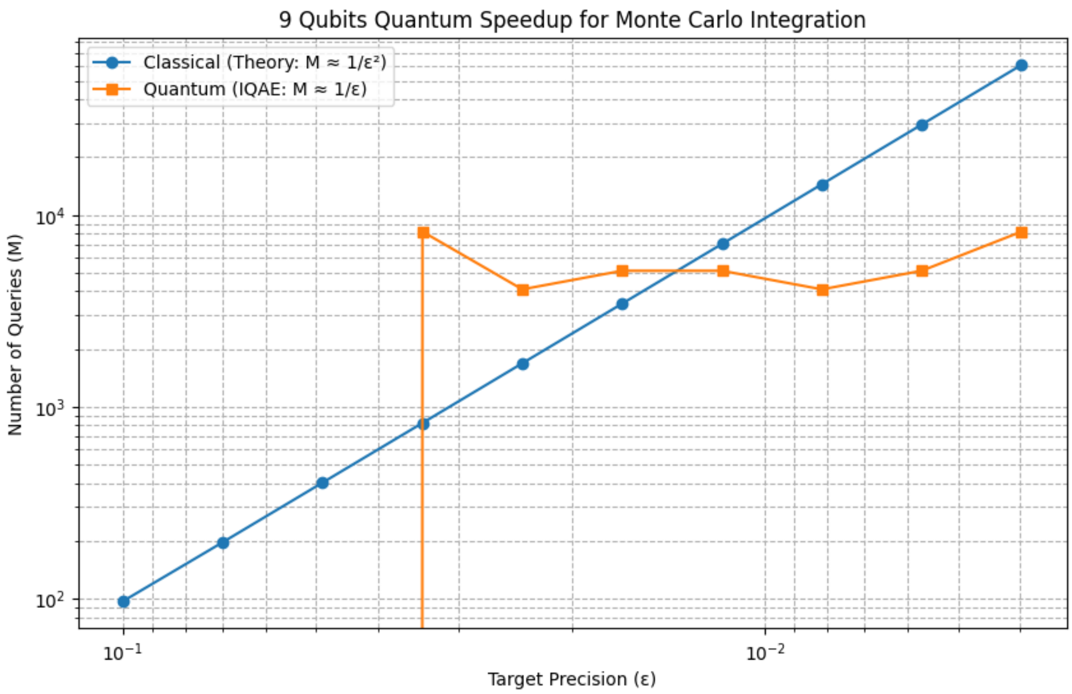

Figure 4: Speedup comparison for x^2. The Quantum method (orange) demonstrates $O(1/\epsilon)$ scaling, significantly outperforming the Classical Monte Carlo method (blue) which scales as $O(1/\epsilon^2)$.

5 Potential Applications

The validated quadratic speedup of the Quantum Integration Engine has significant implications for domains relying on stochastic modeling and combinatorial optimization.

5.1 Financial Portfolio Optimization

In computational finance, assessing the risk profile of a portfolio is computationally expensive. Metrics such as **Value at Risk (VaR)** and **Conditional Value at Risk (CVaR)** require estimating the tail ends of loss distributions [3].

- **Classical Bottleneck:** Classical Monte Carlo methods must sample thousands of random market trajectories to accurately approximate the integral of the loss function, specifically for rare "black swan" events ($O(1/\epsilon^2)$).
- **Quantum Advantage:** By encoding the asset price distributions into the quantum state and defining the loss function as the oracle, IQAE can estimate the expected loss (risk) with quadratically fewer samples ($O(1/\epsilon)$). This allows for real-time risk analysis of complex derivatives and portfolios.

5.2 NP-Complete Problems

The underlying mechanism of IQAE is closely related to Grover's search algorithm, which offers a quadratic speedup for unstructured search problems.

- **Solution Counting:** Many NP-complete problems (e.g., SAT, Traveling Salesperson) require exploring an exponentially growing solution space. IQAE can be adapted to count the number of valid solutions (or measure the volume of the solution space) significantly faster than classical brute-force enumeration.

- **Optimization Landscapes:** For continuous optimization problems within NP-hard landscapes, this integration engine can rapidly estimate the gradient or expected value of cost functions over a domain, facilitating faster convergence for heuristic solvers.

6 Conclusion

The Quantum Integration Engine successfully demonstrates the practical application of quantum amplitude estimation for mathematical integration. By effectively combining Chebyshev approximations with Qiskit's IQAE implementation, we verified the $O(1/\epsilon)$ scaling, offering a clear quadratic speedup over classical Monte Carlo techniques. This framework serves as a foundational tool for accelerating financial simulations and solving complex combinatorial problems.

References

[1] Grinko, D., Gacon, J., Zoufal, C., & Woerner, S. (2021). *Iterative quantum amplitude estimation.* npj Quantum Information, 7(1), 52.

[2] Brassard, G., Hoyer, P., Mosca, M., & Tapp, A. (2002). *Quantum amplitude amplification and estimation.* Contemporary Mathematics, 305, 53-74.

[3] Woerner, S., & Egger, D. J. (2019). *Quantum risk analysis.* npj Quantum Information, 5(1), 15.