

# Quantum Amplitude Estimation for Option Pricing

Jayneel Parikh

April 15, 2026

## Abstract

Quantum amplitude estimation promises a quadratic speedup over classical Monte Carlo for pricing financial derivatives, reducing the sample complexity from  $O(\epsilon^{-2})$  to  $O(\epsilon^{-1})$  for additive error  $\epsilon$ . This statement is a query-complexity result; it does not include variance reduction on the classical side or the cost of implementing the quantum oracle. We test this distinction for European and arithmetic Asian call options. The European case is used to validate the QAE pricing map with a statevector circuit. The Asian case is used to compare simple Monte Carlo with the Kemna–Vorst control variate and to examine the circuit cost of path dependence through small finite-grid quantum oracles. With daily monitoring, the control variate reduces the Asian standard error by a factor of 36.3, so that  $10^5$  variance-reduced paths have about the same variance as  $1.32 \times 10^8$  plain paths. A three-date toy Asian oracle already has depth 2396 after transpilation, before amplitude estimation is applied. These numbers do not locate a hardware crossover. They identify which costs would have to enter that estimate.

*Note:* All figures are original and the source code is available at <https://github.com/jayneel-p>.

## 1 Introduction

A substantial portion of the computational resources deployed in financial markets is spent on pricing and risk management [1]. Financial derivatives are contracts whose payoff depends on the future price or trajectory of an underlying asset. Their prices require expectations under stochastic models, and only for simplest contracts do these expectations have a closed form. The Black–Scholes–Merton model [2, 3] prices vanilla European options analytically, but once the payoff depends on the path of the asset, or on several correlated assets, or if the underlying assumptions for the BS model fail (constant volatility, log-normal distribution, constant risk-free interest, perfect liquidity), the standard tool becomes Monte Carlo simulation [4].

Monte Carlo handles arbitrary payoff structures and stochastic dynamics with little change to the method. Its cost is statistical: with  $M$  independent paths the standard error decays as  $O(M^{-1/2})$ , so halving the error demands four times the computation. This rate is a fundamental consequence of the central limit theorem, and it has motivated a long line of classical work aimed at reducing the variance of the estimator without changing the convergence exponent. Kemna and Vorst [5] observed that for arithmetic Asian options,

whose payoff depends on the time-average of the underlying, the geometric-average Asian has a closed form and is highly correlated with the arithmetic payoff. Using it as a control variate removes most of the sampling noise. At daily monitoring frequency the standard error drops by a factor of roughly 36, comparable to the 20–40× range reported in the literature for similar regimes [5, 4]. A quantum method for this contract should be compared with that reduced-variance scale, not only with simple Monte Carlo.

Quantum amplitude estimation [6] tackles the same expectation-estimation problem from a different direction. Given a unitary operator that prepares the probability distribution and encodes the payoff into an amplitude, amplitude estimation recovers the expectation with  $O(\epsilon^{-1})$  oracle queries for additive error  $\epsilon$ , a quadratic improvement over the classical sampling rate. Rebenrost, Gupta, and Bromley [1] formulated this pricing map under the Black–Scholes model. Stamatopoulos *et al.* [7] made the construction explicit at the circuit level, including comparators, payoff rotations, weighted sums for basket and path-dependent options, and small IBM-hardware demonstrations. Their reported full amplitude-estimation circuits for European examples already range from depth 3927 at  $m = 3$  sampling qubits to depth 285204 at  $m = 9$ , before considering daily path dependence. Wang and Kan [8] push the resource question further for Asian and barrier options under stochastic volatility: even their more efficient weak-Euler instances have  $T$ -counts of order  $10^{11}$  and require  $10^4$ – $10^5$  logical qubits. Manzano *et al.* [9] study an alternative encoding pipeline, while Rendon *et al.* [10] pursue a PDE-based route that changes the bottleneck rather than improving the same Monte Carlo estimator.

The QAE speedup should be read as a query-model statement. It counts calls to the pricing oracle, not the cost of building that oracle. Each oracle call is itself a quantum circuit whose depth grows with the number of qubits used to discretize the price distribution, the complexity of the payoff function, and the number of time steps in the path. The classical baseline changes with the estimator. For path-dependent and high-dimensional contracts, variance reduction can lower the effective cost of Monte Carlo by orders of magnitude. Many existing quantum pricing papers benchmark against simple Monte Carlo, which can overstate the practical gap.

This paper separates three questions. First, how much does the Kemna–Vorst control variate change the Asian Monte Carlo baseline? Second, does the implemented European circuit encode the intended payoff expectation? Third, how quickly does path dependence enter the oracle cost? The answer is unfavorable to simple quantum-advantage narratives: the query speedup remains, but the classical baseline and the oracle are both stronger than the headline comparison suggests.

## 2 Financial background

### 2.1 Risk-neutral geometric Brownian motion

Let  $S_t$  be the price of one risky asset at time  $t$ . In the Black–Scholes model the physical-market dynamics are written

$$dS_t = \mu S_t dt + \sigma S_t dW_t, \tag{1}$$

where  $\mu$  is the physical drift,  $\sigma > 0$  is the volatility, and  $W_t$  is a Brownian motion. Pricing is performed under a risk-neutral measure  $\mathbb{Q}$ , where the drift is the risk-free rate  $r$ . The

pricing dynamics are

$$dS_t = rS_t dt + \sigma S_t dW_t^{\mathbb{Q}}. \quad (2)$$

Here  $W_t^{\mathbb{Q}}$  is Brownian motion under  $\mathbb{Q}$ . This is the model used in the simulations below and in the standard quantum-pricing setup [1, 7].

For initial asset price  $S_0$  and maturity time  $T$ , Eq. (2) gives

$$S_T = S_0 \exp \left[ \left( r - \frac{1}{2} \sigma^2 \right) T + \sigma \sqrt{T} Z \right], \quad Z \sim N(0, 1). \quad (3)$$

where  $Z$  is a standard normal random variable. Thus  $S_T$  is lognormal.

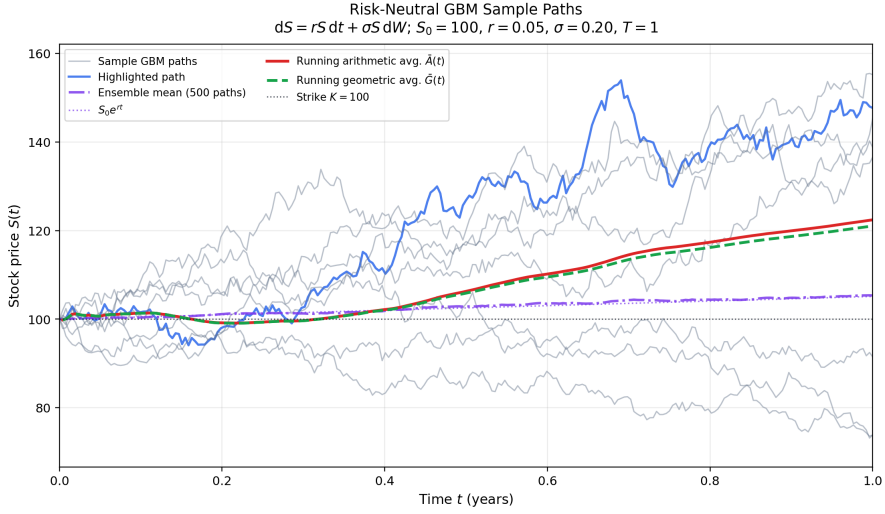


Figure 1: Sample geometric Brownian motion paths under the risk-neutral model. European payoffs depend only on  $S_T$ , while Asian payoffs depend on the path average.

If  $f$  is the payoff paid at time  $T$ , the time-zero price is

$$V_0 = e^{-rT} \mathbb{E}_{\mathbb{Q}}[f]. \quad (4)$$

Here  $V_0$  is the option value at time zero and  $e^{-rT}$  is the discount factor.

## 2.2 European calls

A European call with strike  $K$  has payoff

$$f_E(S_T) = \max(S_T - K, 0). \quad (5)$$

Here  $f_E$  denotes the European-call payoff and  $S_T$  is the terminal asset price. Under Eq. (2), its Black–Scholes price is

$$C_{BS} = S_0 \Phi(d_1) - K e^{-rT} \Phi(d_2), \quad (6)$$

where  $C_{\text{BS}}$  is the Black–Scholes call price,  $\Phi$  is the standard normal cumulative distribution function, and

$$d_1 = \frac{\log(S_0/K) + (r + \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}, \quad d_2 = d_1 - \sigma\sqrt{T}. \quad (7)$$

The quantities  $d_1$  and  $d_2$  are the normalized log-moneyness terms. This closed form is used as a benchmark.

### 2.3 Arithmetic Asian calls

Let  $t_1, \dots, t_N$  be  $N$  monitoring dates satisfying

$$0 < t_1 < t_2 < \dots < t_N = T. \quad (8)$$

The arithmetic Asian call depends on the average of the asset prices at these dates. The dates are taken equally spaced and  $S_0$  is not included. The arithmetic average is

$$A_N = \frac{1}{N} \sum_{i=1}^N S_{t_i}, \quad (9)$$

where  $A_N$  is the average over the monitored prices. The payoff is

$$f_A = \max(A_N - K, 0). \quad (10)$$

Here  $f_A$  denotes the arithmetic Asian payoff. The arithmetic average of lognormal random variables is not lognormal, so Eq. (10) has no Black–Scholes-type closed form in this model.

The corresponding geometric average is

$$G_N = \left( \prod_{i=1}^N S_{t_i} \right)^{1/N} = \exp\left( \frac{1}{N} \sum_{i=1}^N \log S_{t_i} \right). \quad (11)$$

Here  $G_N$  is the geometric average over the same monitoring dates. Since the log-prices are jointly normal,  $\log G_N$  is normal. The geometric Asian call therefore has a closed-form price and is highly correlated with the arithmetic Asian call. Kemna and Vorst use this structure as a control variate [5].

For equally spaced fixings excluding  $S_0$ , define the effective geometric volatility  $\sigma_G$  and drift parameter  $m_G$  by

$$\sigma_G = \sigma \sqrt{\frac{(N+1)(2N+1)}{6N^2}}, \quad m_G = \left( r - \frac{1}{2}\sigma^2 \right) \frac{N+1}{2N}. \quad (12)$$

Then the logarithm of the geometric average is distributed as

$$\log G_N \sim N(\log S_0 + m_G T, \sigma_G^2 T). \quad (13)$$

The resulting geometric Asian call price is

$$C_G = e^{-rT} \left[ S_0 e^{(m_G + \frac{1}{2}\sigma_G^2)T} \Phi(\tilde{d}_1) - K \Phi(\tilde{d}_2) \right], \quad (14)$$

where  $C_G$  is the geometric Asian call price and

$$\tilde{d}_1 = \frac{\log(S_0/K) + (m_G + \sigma_G^2)T}{\sigma_G \sqrt{T}}, \quad \tilde{d}_2 = \tilde{d}_1 - \sigma_G \sqrt{T}. \quad (15)$$

The variables  $\tilde{d}_1$  and  $\tilde{d}_2$  are the analogues of  $d_1$  and  $d_2$  for the distribution of  $G_N$ .

### 3 Classical Monte Carlo baseline

#### 3.1 Simple Monte Carlo

Let  $X$  be the discounted payoff

$$X = e^{-rT} f \quad (16)$$

where  $f$  is the payoff at maturity. If  $X_1, \dots, X_M$  are independent samples of  $X$ , the simple Monte Carlo estimator is

$$\hat{V}_M = \frac{1}{M} \sum_{j=1}^M X_j. \quad (17)$$

where  $\hat{V}_M$  is the estimated time-zero price. The estimator is unbiased under the risk-neutral model. Its standard error is

$$\text{SE}(\hat{V}_M) = \frac{\sqrt{\text{Var}(X)}}{\sqrt{M}}, \quad (18)$$

where  $\text{Var}(X)$  is the payoff variance. In computation this is estimated by replacing  $\text{Var}(X)$  with the sample variance. The  $M^{-1/2}$  dependence is the classical Monte Carlo rate [7, 11].

For the European call, samples are drawn directly from Eq. (3). For the arithmetic Asian call, paths are simulated on the monitoring grid. With time step  $\Delta t = T/N$ , the exact geometric Brownian transition is

$$S_{t_{i+1}} = S_{t_i} \exp \left[ \left( r - \frac{1}{2}\sigma^2 \right) \Delta t + \sigma \sqrt{\Delta t} Z_i \right], \quad Z_i \sim N(0, 1). \quad (19)$$

Here  $S_{t_i}$  and  $S_{t_{i+1}}$  are consecutive monitored prices and  $Z_i$  is a standard normal draw. The grid values are therefore sampled without time-discretization bias.

The numerical experiments use the parameter set

$$S_0 = 100, \quad K = 100, \quad r = 0.05, \quad \sigma = 0.2, \quad T = 1. \quad (20)$$

For the Asian option,  $N = 252$  daily monitoring dates are used. The European Black-Scholes benchmark is

$$C_{\text{BS}} = 10.450584. \quad (21)$$

The geometric Asian benchmark is

$$C_G = 5.565509. \quad (22)$$

### 3.2 Kemna–Vorst control variate

Simple Monte Carlo is not the strongest classical baseline for Asian options. Let

$$X = e^{-rT} \max(A_N - K, 0) \quad (23)$$

be the discounted arithmetic payoff and let

$$Y = e^{-rT} \max(G_N - K, 0) \quad (24)$$

be the discounted geometric payoff. Here  $X$  is the target payoff and  $Y$  is the control payoff. The expectation  $\theta = \mathbb{E}[Y]$  is known from Eq. (14). A control-variate estimator is

$$\widehat{V}_{\text{CV}} = \overline{X} + \widehat{\beta} (\theta - \overline{Y}), \quad (25)$$

where  $\overline{X}$  and  $\overline{Y}$  are sample averages over the same simulated paths and

$$\widehat{\beta} = \frac{\widehat{\text{Cov}}(X, Y)}{\widehat{\text{Var}}(Y)}. \quad (26)$$

The quantity  $\widehat{\beta}$  estimates the variance-minimizing coefficient  $\beta^* = \text{Cov}(X, Y)/\text{Var}(Y)$ . The estimator remains centered on the arithmetic Asian price but removes the part of the arithmetic payoff fluctuation explained by the geometric payoff.

For  $M = 100,000$  paths and  $N = 252$  fixings, the plain estimator has standard error 0.025301. The Kemna–Vorst estimator has standard error 0.000697. This is a standard-error reduction of 36.3 and a variance reduction of  $1.32 \times 10^3$ . In this parameter regime,  $10^5$  control-variate paths have about the same variance as  $1.32 \times 10^8$  simple Monte Carlo paths.

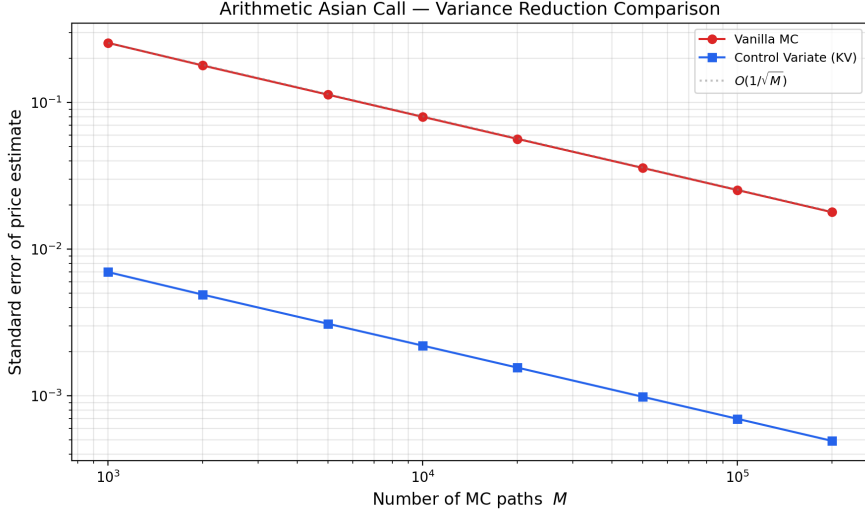


Figure 2: Standard error for the arithmetic Asian call using simple Monte Carlo and the Kemna–Vorst control variate. Both estimators use the same simulated paths in each trial.

The control variate leaves the  $M^{-1/2}$  rate in place, but it lowers the constant by a large factor. This is the baseline used below.

## 4 Quantum amplitude estimation framework

### 4.1 Encoding an expectation as an amplitude

The quantum approach does not sample payoffs. Instead it encodes the undiscounted payoff expectation as the probability of measuring a single qubit in the state  $|1\rangle$ . Discounting by  $e^{-rT}$  is applied classically afterward, exactly as in Eq. (4).

Discretize the terminal asset price onto  $2^n$  grid points  $x_0, \dots, x_{2^n-1}$  with probabilities  $p_i = \Pr[X = x_i]$ . A distribution-loading unitary  $P_X$  prepares the state

$$P_X|0\rangle_n = \sum_{i=0}^{2^n-1} \sqrt{p_i} |i\rangle_n, \quad (27)$$

where  $|i\rangle_n$  is the  $n$ -qubit computational basis state representing  $x_i$ . The amplitudes are square roots of probabilities, so that the Born rule returns  $p_i$  upon measurement of basis state  $|i\rangle$ . The price register, any ancillas, and the objective qubit are part of one joint quantum state. Payoff operations may entangle these registers, but the full map remains unitary until the final measurement.

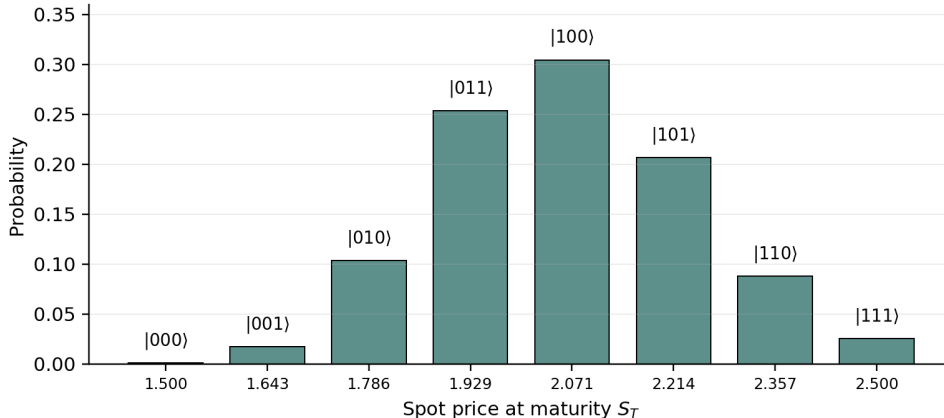


Figure 3: Reproduction of the three-qubit log-normal maturity distribution used in Fig. 3 of Stamatopoulos *et al.* [7]. The basis labels are the states loaded by  $P_X$ . Parameters:  $S_0 = 2$ ,  $r = 0.04$ ,  $\sigma = 0.10$ ,  $T = 300/365$ , and  $S_T \in [1.5, 2.5]$ .

Let  $g_i = g(x_i)$  be the payoff at grid point  $x_i$ , and let  $g_{\max}$  bound it from above so that the normalized payoff  $f_i = g_i/g_{\max}$  satisfies  $0 \leq f_i \leq 1$ . A payoff unitary  $U_f$  rotates one objective qubit conditioned on the price register:

$$U_f|i\rangle_n|0\rangle = |i\rangle_n \left( \sqrt{1-f_i}|0\rangle + \sqrt{f_i}|1\rangle \right). \quad (28)$$

The full state-preparation operator  $A = U_f(P_X \otimes I)$  acts on  $n+1$  qubits. Applying it to the all-zero input gives

$$A|0\rangle_n|0\rangle = \sqrt{1-a}|\psi_0\rangle + \sqrt{a}|\psi_1\rangle, \quad (29)$$

where  $|\psi_1\rangle$  collects all terms with objective qubit  $|1\rangle$  and  $|\psi_0\rangle$  collects those with  $|0\rangle$ . The Born rule now does the work. The probability of measuring the objective qubit in state  $|1\rangle$  is the sum of the squared amplitudes in the  $|1\rangle$  subspace, which by construction equals

$$a = \Pr[\text{objective} = 1] = \sum_{i=0}^{2^n-1} p_i f_i = \mathbb{E}[f(X)]. \quad (30)$$

Thus the measurement probability is exactly the normalized expected payoff. The operator  $A$  follows the prepare-then-compute pattern common to quantum algorithms. Here  $P_X$  prepares the input distribution,  $U_f$  computes the function of interest into an ancilla amplitude, and measurement extracts the expectation [1, 7].

For a European call,  $g_i = \max(x_i - K, 0)$  and the time-zero price is

$$C_0 = e^{-rT} g_{\max} a. \quad (31)$$

Stamatopoulos *et al.* implement  $U_f$  with controlled  $R_y$  rotations and a first-order approximation to the payoff map [7]. The implementation used here keeps the same division between distribution loading and payoff rotation.

## 4.2 Amplitude amplification and estimation

Estimating  $a$  by repeated measurement of  $A|0\rangle_n|0\rangle$  would reproduce the  $O(M^{-1/2})$  Monte Carlo rate. The quadratic improvement comes from amplitude estimation, which repeatedly applies the Grover iterate defined below [6]. In Grover search the goal is to find a marked item; here the “marked” states are those where the option finishes in the money, and the goal is to estimate their total weight.

Define the projector  $\Pi_1 = I_n \otimes |1\rangle\langle 1|$  onto the subspace where the objective qubit is  $|1\rangle$ , and the two reflections

$$S_1 = I - 2\Pi_1, \quad S_0 = I - 2|0\rangle_{n+1}\langle 0|_{n+1}. \quad (32)$$

The Grover iterate is

$$Q = -AS_0A^\dagger S_1. \quad (33)$$

The reflection  $S_1$  marks the good subspace;  $AS_0A^\dagger$  reflects about the prepared state. Two reflections compose into a rotation in the two-dimensional subspace spanned by  $|\psi_0\rangle$  and  $|\psi_1\rangle$ . Writing  $a = \sin^2 \theta$  with  $\theta \in [0, \pi/2]$ , repeated applications give

$$Q^k A|0\rangle_n|0\rangle = \cos((2k+1)\theta) |\psi_0\rangle + \sin((2k+1)\theta) |\psi_1\rangle, \quad (34)$$

Hence each application of  $Q$  advances the state by angle  $2\theta$  toward the good subspace. Amplitude estimation recovers  $\theta$ , and hence  $a = \sin^2 \theta$ , by applying phase estimation to  $Q$  [6]. With  $M = 2^m$  oracle calls the error scales as  $O(M^{-1})$ , the quadratic improvement over sampling.

The numerical experiments use iterative amplitude estimation (IAE), which avoids the inverse quantum Fourier transform and controlled- $Q^{2^j}$  powers of the circuit. IAE estimates the same angle  $\theta$  through a sequence of Grover iterates at different depths, combined with classical likelihood inference [12, 7]. At each depth  $k$ , the measured success probability is  $\sin^2((2k+1)\theta)$ , so the classical post-processing updates an interval for  $\theta$  rather than reading it from a phase-estimation register. Only the estimation subroutine differs from canonical QAE; the state preparation in Eqs. (27)–(30) is unchanged.

## 4.3 European-call circuit

The operator  $A$  for a European call has two stages: *i*) load the terminal price distribution and *ii*) rotate the objective qubit according to the normalized payoff. Fig. 4 shows the high-level structure. In practice,  $a$  is estimated by IAE rather than by naive Bernoulli sampling of the objective qubit; the classical rescaling in Eq. (31) then recovers the option price [12, 7].

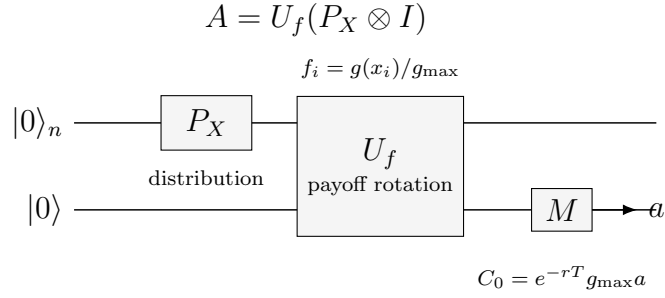


Figure 4: State-preparation circuit for a European call. The distribution loader  $P_X$  prepares the discretized law of  $S_T$ , and  $U_f$  rotates the objective qubit by an amount determined by the normalized payoff.

The circuits below follow the gate-level decomposition of Stamatopoulos *et al.* and are meant to make the oracle structure explicit. Our implementation uses Qiskit Finance’s log-normal distribution loader and `LinearAmplitudeFunction` to produce the same normalized piecewise-linear payoff rotation [13]. Thus the code and the figures implement the same pricing map, although our code delegates the low-level comparator and rotation to Qiskit rather than the gates shown here.

Stamatopoulos *et al.* implement  $U_f$  in two stages [7]. A reversible comparator determines whether the encoded price index  $i$  exceeds the strike. Let  $t[k]$  be the  $k$ th bit of the strike in binary. The comparator acts on the price register  $|i\rangle_n$ , carry ancillas  $|a_1\rangle, \dots, |a_n\rangle$ , and a comparator qubit  $|c\rangle$ , setting  $|c\rangle = |1\rangle$  when  $i \geq K$ . The construction uses only CNOT and Toffoli gates, similar to the half-adder.

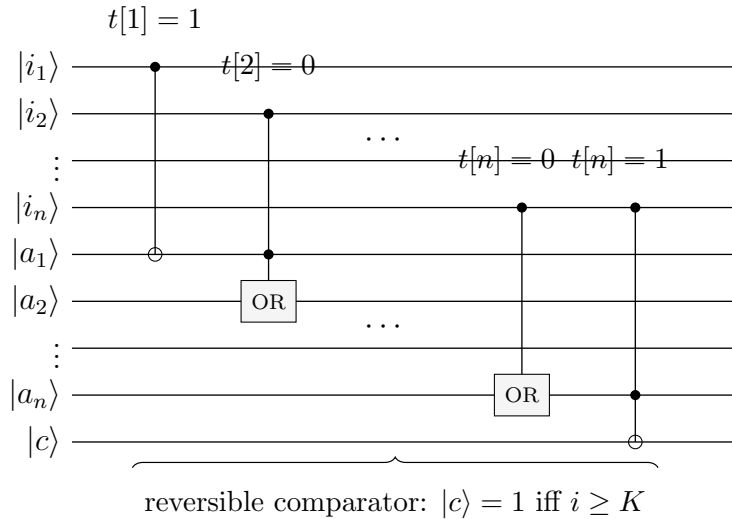


Figure 5: Comparator structure used for the European call, following the construction of Stamatopoulos *et al.* [7]. Only one of the  $t[k] = 0$  or  $t[k] = 1$  branches is used for each bit of the fixed strike.

The OR gate in Fig. 5 decomposes into Pauli- $X$  gates and a Toffoli, as shown in Fig. 6: negate the inputs, apply a Toffoli (which computes AND), negate the output and restore the inputs. The result is  $c \leftarrow a \vee b$  with  $a$  and  $b$  unchanged.

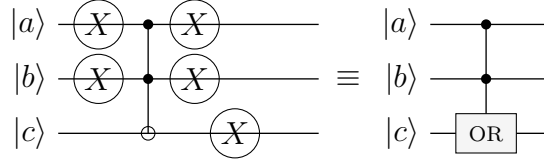


Figure 6: Reversible OR subcircuit used inside the comparator. The right-hand symbol abbreviates the Toffoli-based construction on the left.

Second, once  $|c\rangle$  stores the comparison result, a controlled  $R_y$  rotation loads the payoff into a fresh ancilla qubit. The rotation is applied only on the in-the-money branch ( $i \geq K$ ). Following Stamatopoulos *et al.*, the rotation angle on that branch is  $g_0 + g(i)$ , where

$$g(i) = \frac{2c(i - K)}{i_{\max} - K}, \quad g_0 = \frac{\pi}{4} - c, \quad (35)$$

$c$  is a scaling parameter that controls the approximation quality, and  $i_{\max} = 2^n - 1$  is the largest grid index. The key step is the expansion around  $\pi/4$ :

$$\sin^2\left(\frac{\pi}{4} + x\right) = \frac{1}{2} + x + O(x^3). \quad (36)$$

For small  $x$ , the objective-qubit probability is therefore affine in the payoff (a linear transformation of the option payoff plus a small constant). This linearity is what allows Eq. (30) to hold as a pricing identity after the known shift and scale are removed.

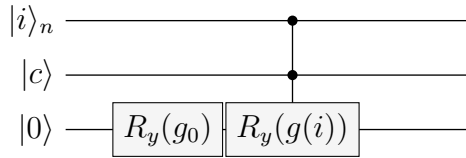


Figure 7: Payoff-loading rotation for the European call. The comparator qubit  $|c\rangle$  activates the  $i$ -dependent rotation only on the branch  $i \geq K$ .

The Asian case uses the components at a greater cost. Instead of loading a single terminal price, the circuit must prepare  $N$  correlated monitoring-date prices, compute their arithmetic average into a work register using reversible adders, apply the comparator and payoff rotation to the average, and then uncompute the work registers so that  $A^\dagger$  in Eq. (33) returns the ancillas to a clean state. Every Grover query therefore contains path loading, reversible averaging, comparison with the strike, payoff rotation, and the corresponding inverse operations. This overhead is the circuit cost of path dependence, and it is why path-dependent options are often used to motivate quantum advantage in derivative pricing [7, 8]. There is a hitch, variance reduction, optimized integrators, and better sampling can make the practical gap much smaller than a simple-Monte-Carlo comparison suggests.

## 5 Results and discussion

### 5.1 Classical benchmark

The European call is mainly a check on the simulation pipeline. Simple Monte Carlo converges to the Black–Scholes price with the expected  $M^{-1/2}$  rate from Eq. (18). With  $5 \times 10^5$  paths, averaged over 50 independent trials, the mean standard error is 0.02081 and the mean absolute error is 0.01456. This result is not computationally interesting by itself because the European call has a closed form.

The Asian option is the relevant classical benchmark. Fig. 2 shows that the Kemna–Vorst control variate preserves the Monte Carlo exponent but changes the prefactor sharply. At  $M = 100,000$  paths, the standard error falls from 0.025301 for simple Monte Carlo to 0.000697 with the control variate. This is a standard-error reduction of 36.3 and a variance reduction of  $1.32 \times 10^3$ , consistent with the high correlation between arithmetic and geometric Asian payoffs used by Kemna and Vorst [5, 4].

Table 1: Benchmark checks for the Asian-pricing code. Xu–Zhang–Wang values are from Table 4, row “w/o ALL”, their plain Black–Scholes Monte Carlo ablation [11].

Check	Reference value	Our estimate	Diagnostic
Geometric Asian	5.565509	$5.577325 \pm 0.014132$	$z = 0.84$
Xu $T = 30$ days	$1.3871 \pm 0.0201$	$1.4331 \pm 0.0020$	$z = 2.29$
Xu $T = 90$ days	$2.6589 \pm 0.0373$	$2.7030 \pm 0.0038$	$z = 1.18$
Xu $T = 180$ days	$4.0166 \pm 0.0546$	$4.0870 \pm 0.0056$	$z = 1.29$

These checks have a narrow purpose. They verify the simulation and control-variate implementation. They do not speak to the legitimacy of the market model as a predictor. For the Xu–Zhang–Wang rows, the diagnostic uses the Monte Carlo standard error reported in their table. The 30-day check is the loosest row, but it remains inside the three-standard-error band. The quantum method is therefore being measured against a validated variance-reduced baseline, not only against simple Monte Carlo.

### 5.2 European quantum validation

The European call has a closed form, which makes it useful as a calibration problem. The statevector calculation removes sampling noise and IAE uncertainty, so the test isolates the pricing map in Eqs. (27)–(30). If the state preparation and payoff rotation are correct, the objective-qubit probability should reproduce the Black–Scholes price after the known rescaling in Eq. (31).

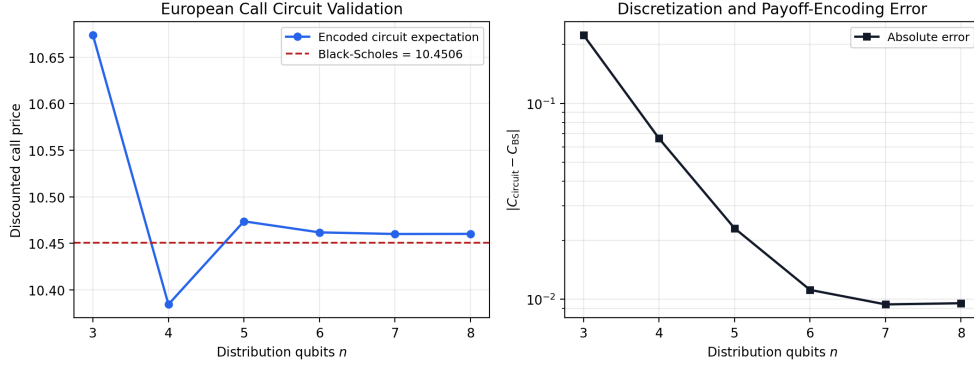


Figure 8: European-call circuit validation using exact statevector probabilities. The comparison tests the encoded amplitude, not the sampling behavior of IAE.

The encoded price is already close with a small register. At  $n = 3$  distribution qubits the circuit gives 10.673862, an absolute error of 0.223279 against the Black–Scholes price 10.450584. At  $n = 6$  the error falls to 0.011167. It then plateaus near  $10^{-2}$ , with errors 0.009416 and 0.009547 at  $n = 7$  and  $n = 8$ . It seems like refining the price grid helps up to a limit.

IAE estimates the amplitude already present in the circuit [12]. It does not remove truncation error in the log-normal grid, nor does it repair the first-order payoff rotation used in the Stamatopoulos construction [7]. The statevector result therefore validates the circuit as an amplitude-encoding implementation, but not as a hardware speedup.

### 5.3 Resource tradeoff

The query speedup from QAE is meaningful only after the cost of one query is fixed. Table 2 reports the compiled size of the European  $A$  operator. One Grover iterate contains  $A$ ,  $A^\dagger$ , and two reflections, so an IAE run repeats a circuit at least of this scale many times. The counts are generated by `scripts/generate_quantum_results.py --quick` with Qiskit 2.3.1, Qiskit Finance 0.4.1, and Qiskit Algorithms 0.4.0, transpiled to the `cx, u` basis at optimization level 1. Grinko *et al.* reduce the estimation overhead by avoiding QPE, but the state-preparation and payoff oracles remain the dominant problem-specific cost [12].

Table 2: Transpiled European-call  $A$ -operator resources. Counts are for state preparation and payoff loading only.

Distribution qubits	Grid points	Total qubits	Depth	CX gates
3	8	7	143	94
4	16	9	218	141
5	32	11	303	196
6	64	13	426	267
7	128	15	607	370
8	256	17	922	537

Table 3: Toy arithmetic-Asian oracle checks. Each monitoring date uses a two-qubit shock grid. These circuits validate the finite-grid payoff map; they are not production resource estimates.

Monitoring dates	Grid paths	Total qubits	Depth	CX gates
2	16	5	293	194
3	64	7	2396	1963

The growth is moderate for the European call because the payoff depends only on  $S_T$ , unlike the Asian circuit. Table 3 shows the same payoff map on small finite grids. The three-date circuit reproduces its exact grid price to numerical precision, but it already reaches depth 2396 before amplitude estimation. A linear extrapolation from three dates to 252 dates would give depth on the order of  $2 \times 10^5$  and about  $1.6 \times 10^5$  CX gates before any Grover repetitions. This estimate is only a scale check. Exact finite-grid loading would grow faster, while a serious production oracle would need more structure than the toy circuit uses to compensate for market complicated market dynamics.

For the daily Asian option used in the classical benchmark, a quantum oracle would need to load 252 correlated prices, compute the arithmetic average reversibly, compare it with the strike, rotate the payoff qubit, and uncompute the work registers. Stamatopoulos *et al.* identify this path-dependent structure as the place where reducing the number of samples could matter most [7]. Wang and Kan reach a similar conclusion under stochastic volatility, where the resource analysis is dominated by arithmetic and path simulation circuits rather than by the abstract QAE primitive [8].

These counts are still noiseless circuit counts. Stamatopoulos *et al.* ran only small European-call instances on IBM hardware and needed error mitigation to reduce two-qubit gate errors [7]. IAE removes the QFT register and deep controlled powers, but it still requires repeated coherent applications of the Grover algorithm [12]. For larger path-dependent contracts, the relevant hardware question is therefore not only the number of oracle calls, but whether the full oracle can be executed with enough fidelity. This is why fault-tolerant resource estimates, such as the  $T$ -count and  $T$ -depth analysis in Wang and Kan, are more informative than query counts alone [8].

## 6 Conclusion

We studied option pricing with QAE against a variance-reduced classical baseline. For the European call, the statevector calculation verifies the amplitude map used in the previous section. After loading the log-normal terminal distribution and encoding the payoff into the objective qubit, the recovered price approaches the Black–Scholes value as the grid is refined. The remaining error likely comes from grid truncation and payoff encoding.

The Asian option gives the more useful test. With daily monitoring, the Kemna–Vorst control variate lowers the variance of the arithmetic Asian estimator by  $1.32 \times 10^3$  at  $10^5$  paths. This changes the baseline by three orders of magnitude in variance. On the quantum side, each Asian oracle must prepare the path distribution, compute the average reversibly, compare with the strike, load the payoff, and uncompute the work registers before the Grover iterate can be applied. The "quantum advantage" depends on the payoff, the available variance reduction, and the cost of implementing the oracle. It also depends on hardware. A deep oracle repeated inside amplitude estimation is sensitive

to noise on present devices and expensive under error correction. For the Black–Scholes Asian case considered here, the evidence points in one direction. The query speedup is real, but it is not the bottleneck. The bottleneck is the full path-dependent oracle, measured against a control variate that already removes most of the classical variance.

## References

- [1] P. Rebstroff, B. Gupt, and T. R. Bromley. “Quantum Computational Finance: Monte Carlo Pricing of Financial Derivatives”. In: *Physical Review A* 98.2 (Aug. 20, 2018), p. 022321. ISSN: 2469-9926. DOI: 10.1103/PhysRevA.98.022321. URL: <https://link.aps.org/doi/10.1103/PhysRevA.98.022321> (visited on 04/15/2026).
- [2] F. Black and M. Scholes. “The Pricing of Options and Corporate Liabilities”. In: *Journal of Political Economy* 81.3 (1973), pp. 637–654. DOI: 10.1086/260062.
- [3] R. C. Merton. “Theory of Rational Option Pricing”. In: *The Bell Journal of Economics and Management Science* 4.1 (1973), pp. 141–183. DOI: 10.2307/3003143.
- [4] P. Glasserman. *Monte Carlo Methods in Financial Engineering*. New York: Springer, 2004. DOI: 10.1007/978-0-387-21617-1.
- [5] A. Kemna and A. Vorst. “A Pricing Method for Options Based on Average Asset Values”. In: *Journal of Banking & Finance* 14.1 (Mar. 1990), pp. 113–129. ISSN: 03784266. DOI: 10.1016/0378-4266(90)90039-5. URL: <https://linkinghub.elsevier.com/retrieve/pii/0378426690900395> (visited on 04/15/2026).
- [6] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp. “Quantum Amplitude Amplification and Estimation”. In: *Contemporary Mathematics* 305 (2002), pp. 53–74. DOI: 10.1090/conm/305/05215.
- [7] N. Stamatopoulos, D. J. Egger, Y. Sun, C. Zoufal, R. Iten, N. Shen, and S. Woerner. “Option Pricing Using Quantum Computers”. In: *Quantum* 4 (July 6, 2020), p. 291. ISSN: 2521-327X. DOI: 10.22331/q-2020-07-06-291. arXiv: 1905.02666 [quant-ph]. URL: <http://arxiv.org/abs/1905.02666> (visited on 04/15/2026).
- [8] G. Wang and A. Kan. “Option Pricing under Stochastic Volatility on a Quantum Computer”. In: *Quantum* 8 (Oct. 23, 2024), p. 1504. ISSN: 2521-327X. DOI: 10.22331/q-2024-10-23-1504. arXiv: 2312.15871 [quant-ph]. URL: <http://arxiv.org/abs/2312.15871> (visited on 04/15/2026).
- [9] A. Manzano, G. Ferro, Á. Leitao, C. Vázquez, and A. Gómez. “Alternative Pipeline for Option Pricing Using Quantum Computers”. In: *EPJ Quantum Technology* 12.1 (Dec. 2025), p. 28. ISSN: 2662-4400. DOI: 10.1140/epjqt/s40507-025-00328-3. URL: <https://epjquantumtechnology.springeropen.com/articles/10.1140/epjqt/s40507-025-00328-3> (visited on 04/15/2026).
- [10] G. Rendon, R. Kshirsagar, and Q. H. Tran. *Exponential Improvement on Asian Option Pricing Through Quantum Preconditioning Methods*. May 8, 2025. DOI: 10.48550/arXiv.2501.15614. arXiv: 2501.15614 [quant-ph]. URL: <http://arxiv.org/abs/2501.15614> (visited on 04/15/2026). Pre-published.

- [11] L. Xu, H. Zhang, and F. L. Wang. “Pricing of Arithmetic Average Asian Option by Combining Variance Reduction and Quasi-Monte Carlo Method”. In: *Mathematics* 11.3 (Jan. 23, 2023), p. 594. ISSN: 2227-7390. DOI: 10.3390/math11030594. URL: <https://www.mdpi.com/2227-7390/11/3/594> (visited on 04/15/2026).
- [12] D. Grinko, J. Gacon, C. Zoufal, and S. Woerner. “Iterative Quantum Amplitude Estimation”. In: *npj Quantum Information* 7.1 (2021), p. 52. DOI: 10.1038/s41534-021-00379-1. arXiv: 1912.05559 [quant-ph]. URL: <https://www.nature.com/articles/s41534-021-00379-1>.
- [13] Qiskit Finance Development Team. *Qiskit Finance*. Version 0.4.1, released 2024-02-29. Accessed 2026-04-15. 2024. URL: <https://github.com/qiskit-community/qiskit-finance>.