Intermediate Qutrit-based Improved Quantum Arithmetic Operations with Application on Financial Derivative Pricing

Amit Saha, Turbasu Chatterjee, Anupam Chattopadhyay, Senior Member, IEEE, and Amlan Chakrabarti, Senior Member, IEEE

Abstract—In some quantum algorithms, arithmetic operations are of utmost importance for resource estimation. In binary quantum systems, some efficient implementation of arithmetic operations like, addition/subtraction, multiplication/division, square root, exponential and arcsine etc. have been realized, where resources are reported as a number of Toffoli gates or T gates with ancilla. Recently it has been demonstrated that intermediate qutrits can be used in place of ancilla, allowing us to operate efficiently in the ancilla-free frontier zone. In this article, we have incorporated intermediate qutrit approach to realize efficient implementation of all the quantum arithmetic operations mentioned above with respect to gate count and circuit-depth without T gate and ancilla. Our resource estimates with intermediate qutrits could guide future research aimed at lowering costs considering arithmetic operations for computational problems. As an application of computational problems, related to finance, are poised to reap the benefit of quantum computers, in which quantum arithmetic circuits are going to play an important role. In particular, quantum arithmetic circuits of arcsine and square root are necessary for path loading using the re-parameterization method, as well as the payoff calculation for derivative pricing. Hence, the improvements are studied in the context of the core arithmetic circuits as well as the complete application of derivative pricing. Since our intermediate qutrit approach requires to access higher energy levels, making the design prone to errors, nevertheless, we show that the percentage decrease in the probability of error is significant owing to the fact that we achieve circuit robustness compared to qubit-only works.

Index Terms—Derivative pricing, Quantum arithmetic, Intermediate qutrit, Quantum resource estimation, Qudit systems.

1 INTRODUCTION

Quantum computers are expected to outperform classical computers at certain computational tasks with an asymptotic advantage [1]. While quantum algorithms for computationally difficult tasks have favourable asymptotic improvement, specific run-time estimates are frequently lacking due to the dearth of efficient implementation of quantum arithmetic operations that are employed as functions in quantum algorithms [2]. We have tried to address this issue by designing advanced arithmetic circuits with the adoption of higher energy level of quantum state. As an application, we consider financial derivative pricing problem to demonstrate our efficient approach of synthesizing arithmetic operations in quantum domain.

In this article, we shall be focusing on the usage of quantum algorithms valuation or the pricing of these financial derivatives. Recently, it has been shown that quantum approaches to derivative pricing might exhibit quantum advantage with bounded error [3]. The pricing methodology with a strong focus on the implementation of the algorithms in a gate-based quantum computer has also been addressed in [3], [5]. According to the most recent work on the decomposition of the solution of derivative pricing presented in [6], at the granular level, one has to take into account the basic building blocks of the algorithm: The arithmetic operations, which encompass addition, subtraction, multiplication, square root, exponentiation, and the arcsine. An inefficient implementation of these blocks can employ colossal noise in the system, hence, derivative pricing algorithm may suffer with inappropriate outcome.

The goal of this paper is to disintegrate the quantum derivative pricing algorithm presented in [6] into the arithmetic operations and put more efficient algorithms for the same at its foundation. In doing so, the methods prescribed in this paper do hope to carry out an asymptotic reduction of the complexity prescribed in the paper and improve the error bounds of the algorithm in its state-of-the-art form by decomposing Toffoli gates with the adoption of intermediate qutrit approach [7]. All the generalized qutrit gates [8], [9] have already been successfully implemented on superconducting [10], trapped ion [11] and photonic systems [12] indicating it is possible to consider higher level systems apart from qubit-only systems. The Toffoli gate decomposition technique necessitates access to a higher energy level, rendering the design vulnerable to errors. Nonetheless, we demonstrate that the percentage reduction in the probability of error is significant because we reduced both gate count and circuit depth compared to previous work [6].

Our specific contributions in a nutshell are as follows:

- We employ intermediate qutrits to address basic quantum arithmetic operations and estimate the resources
to show that our approach supersedes the current techniques on the basis of circuit depth and circuit robustness.

- Since arithmetic operations are an integral aspect of derivative pricing, hence we exhibit this as an example for application efficiently.
- Our numerical analysis establishes that adoption of intermediate qutrit for Toffoli decomposition is sub-liner than the existing works by obtaining a percentage increase in the probability of success.

The rest of this article is organized as follows: Section 2 exhibits relevant preliminaries about Derivative pricing with arithmetic operations. Section 3 demonstrates our intermediate qutrit approach to quantum arithmetic operations, and Section 4 analyzes the resources required for proposed quantum arithmetic operations. The performance of the Toffoli decomposition with intermediate qutrit under various types of noise is presented in Section 5. Section 6 captures our concluding remarks.

2 DERIVATIVE PRICING

In finance, the term ‘derivative’ refers to a contract that derives its pricing from an underlying financial entity, such as, asset, index and interest rate. A derivative contract is typically valid until its expiration date and is issued between an issuer and a holder. History tells us that these derivatives date way back to the time of Aristotle [13], and ever since then, these financial instruments saw a massive boom in trade during the last fifty years and have developed into complex financial instruments. The complexity of pricing a derivative in the modern market scenario is due to the nature of the underlying, on which they are being valued. Moreover, these financial instruments can be subject to a broad spectrum of usage, e.g., valuation, arbitrage, speculation and hedging. Calculating the fair value through derivative pricing can be difficult due to the stochastic nature of the parameters on which they are defined. Hence, many numerical approaches for derivative pricing are frequently used, with Monte Carlo being one of the most prominent due to its versatility and ability to handle stochastic parameters generically [14]. Monte Carlo methods, despite their appealing qualities in derivative pricing, typically demand a lot of computer resources to generate reliable option price estimates, especially for complex options. It is evident from the literature that a quantum computer may provide novel ways to solve computationally intensive problems by leveraging quantum mechanical laws [1]. More specifically, there have been multiple approaches to derivative pricing on the quantum frontier. We, however, focus our paper on the works of Chakrabarti et al. [6], wherein we come across rigorous resource estimates for executing derivative pricing on a quantum computer.

The price of an underlying asset(s) is generally modeled as a stochastic process under the pretext of “no arbitrage,” which assumes that an underlying asset(s) may not be priced differently in different markets, such that, there are no profits from trading between markets. The Black-Scholes method is used to model the evolution of the value of a financial asset as a geometric Brownian motion. Therefore, let $S^i_t \in \mathbb{R}^d$ be a vector at time $t$ for $d$ underlyings. Let $(S^0_t, S^1_t, \ldots, S^d_t) = \omega \in \Omega$ be a path of a discrete-time multivariate stochastic process describing the values of the underlying assets. The probability density function is denoted by $p(\omega)$. Let $f(\omega) = f(S^0_t, \ldots, S^d_t) \in \mathbb{R}$ be the discounted payoff of some derivative on those assets. In order to evaluate the price of the derivative, we would calculate

$$\mathbb{E}(f) = \int_{\omega \in \Omega} p(\omega) f(\omega) d\omega. \quad (1)$$

The pricing for a given derivative can be calculated analytically using a variety of methods, notably the Black-Scholes model for path-independent evaluations and Monte-Carlo methods for path-dependent evaluations, the likes of autocallables and Target redemption forwards (TARFs). Computationally, it has been noted that the path-independent derivative is less intensive, while path-dependent evaluations still remain to be more intensive. Using Monte-Carlo methods classically, given $M$ paths, the derivative pricing accuracy converges as $O(1/\sqrt{M})$. This bound was improved upon by Montenaro and was found to be $O(1/M)$ in the quantum realm using quantum algorithms based on amplitude estimation, thus paving the way for investigations in derivative pricing, risk analysis and overall, quantum finance.

In order to price the derivatives on a discrete variable quantum computer (DVQC), the paths need to be discretized and then mapped onto quantum states. Therefore the expectation now looks like

$$\mathbb{E}(f) \approx \frac{1}{N} \sum_{\omega = 1}^{M} f(\omega), \quad (2)$$

which converges to the true expected value with error $\epsilon = O(M^{-1/2})$ by the central limit theorem. This convergence has been shown to exhibit quadratic speedup to $\epsilon = O(M^{-1})$ when using quantum amplitude estimation [15] for Monte-Carlo processes [3], [16], [17]. Quantum amplitude estimation takes a unitary operator $\mathcal{A}$, acting on $n + 1$ qubits such that,

$$\mathcal{A}|0\rangle_{n+1} = \sqrt{1 - a}|\psi_0\rangle_n|0\rangle + \sqrt{a}|\psi_1\rangle_n|1\rangle, \quad (4)$$

where the parameter $a$ is unknown and the final qubit acts as a label and is used to distinguish the $|\psi_0\rangle$ states from the $|\psi_1\rangle$ states.

The goal of the quantum amplitude estimation algorithm is to estimate the unknown parameter $a$. This is done by repeatedly applying the Grover operator $Q = \mathcal{A}\mathcal{S}_0\mathcal{A}^\dagger\mathcal{S}_\psi$, where $S_0 = I - 2|0\rangle_n\langle 0|_{n+1}$ and $S_\psi = I - 2(|\psi_0\rangle_n|0\rangle\langle 0|_{n+1}$ are reflection operators. The parameter $a$ can be determined with an accuracy of $O(M^{-1})$ using quantum phase estimation and quantum Fourier transform [1]. However, due to the use of the expensive quantum Fourier transform, this method has undergone development [18], [19], [20], [21], [22], [23] and the Iterative Quantum Amplitude Estimation (IQAE) [20], which has been regarded as the most efficient
methods of the quantum amplitude estimation has seen its use in [6].

The basic approach to the derivative pricing algorithm consists of the following steps:

1) Apply some operator \( \mathcal{P} \) to the state \( |0\rangle \), such that
\[
\mathcal{P}|0\rangle = \sum_{\omega} \sqrt{p(\omega)}|\omega\rangle
\]

2) Calculate \( \delta(\omega) = \arcsin \sqrt{f(\omega)} \), such that
\[
\sum_{\omega} \sqrt{p(\omega)}|\omega\rangle|\delta(\omega)\rangle
\]

3) Introduce an ancilla qubit and use controlled rotations to rotate the value of \( f(\omega) \) register into its amplitude, given by
\[
\sum_{\omega} \sqrt{p(\omega)(1 - f(\omega))}|\omega\rangle|0\rangle + \sum_{\omega} \sqrt{p(\omega)(f(\omega))}|\omega\rangle|1\rangle
\]

4) Use quantum amplitude estimation and extract the probability of the ancilla being \( |1\rangle \)

The output yields the discretized expected payoff, \( \mathbb{E}(\tilde{f}) = \sum_{\omega} p(\omega)f(\omega) \), subsequently rescaled to yield \( \mathbb{E}(f) = (f_{\text{max}} - f_{\text{min}})\mathbb{E}(\tilde{f}) + f_{\text{min}} \).

In order for this algorithm to have a quantum advantage, one needs to take into account, the costs for path loading for the payoff calculation. The loading of general probability distributions have been proven to be exponentially hard. Therefore, one needs to find an efficient path loading operator \( \mathcal{P} \). The Grover-Rudolph method is an efficient method for loading distributions which are integrable. This however is not useful for applications in derivative pricing as the probability distributions herein are not integrable and the paths are calculated using quantum Monte Carlo methodologies. Here, the author of [6] uses a variational method, termed as the Re-parameterization method to tackle the problem of path loading.

The Re-parameterization method works by observing that probability distributions may be loaded by preparing, in parallel, several standard normal distributions, and applying affine transformations to shape them into the desired distribution with the required mean and standard deviation. The steps for the Re-parameterization method can be outlined as follows:

1) Apply \( dT \) Gaussian operators \( \mathcal{G} \), to \( ndT \) qubits, thereby constructing:
\[
\bigotimes_{j=1}^{d} \bigotimes_{t=1}^{T} \mathcal{G}|0\rangle = \sum_{\omega_R} \sqrt{p(\omega_R)}|\omega_R\rangle_{\text{ndT}},
\]

where \( \omega_R \) runs over all \( 2^{ndT} \) different realizations of the multivariate standard Gaussian, and \( p(\omega_R) \) are the corresponding probabilities.

2) Given \( \Sigma = LL^T \) is the Cholesky decomposition of the covariance matrix, perform the affine transformation \( \tilde{R}' = \tilde{R} + L^T \tilde{R} \) to adjust the mean and standard deviation of each Gaussian. These will be used to calculate the asset prices from the log-returns given by
\[
S'_{t} = \sum_{j=0}^{d} e^{\mu_t j + \sum_{i=0}^{d} L_{ij} \tilde{R}'} = e^{i\sum_{j=0}^{d} \mu_t j + \sum_{i=0}^{d-1} L_{ij} \tilde{R}'}
\]

3) If the payoffs can be calculated directly from the log-returns, then the quantity \( \delta(\omega_R) = \arcsin \sqrt{f(\omega_R)} \) can be calculated in the quantum register as follows:
\[
\sum_{\omega_R} \sqrt{p(\omega_R)|\omega_R\rangle|\delta(\omega_R)\rangle}
\]

where \( \omega_R \) are the paths and \( p(\omega_R) \) are the corresponding probabilities. If the payoff is defined in terms of prices and not just log-returns, then price path \( \omega \) is calculated for each asset using
\[
\tilde{S}' = \sum_{j=1}^{d} e^{L_{ij} \tilde{R}'}.
\]

This can be done in parallel for each asset.

4) An ancilla qubit is introduced and the value of \( \tilde{f}(\omega) \) is rotated into its amplitude:
\[
\sum_{\omega} \sqrt{p(\omega)(1 - \tilde{f}(\omega))}|\omega\rangle|0\rangle + \sum_{\omega} \sqrt{p(\omega)(\tilde{f}(\omega))}|\omega\rangle|1\rangle
\]

5) Quantum amplitude estimation is used to extract the probability of the ancilla being \( |1\rangle \)

The above routine can be visualized in figure [1].

Fig. 1. Circuit computing \( T \) registers containing cumulative log-returns \( \tilde{R}' \) for each timestep \( t' \in [0, T] \). The \( T \) Gaussian operators \( \mathcal{G} \) are applied to the registers to generate Gaussian probability distributions at each timestep. The ADD operators are then used in series to transform \( |\tilde{x}\rangle|\tilde{y}\rangle(0) \to |\tilde{x}\rangle|\tilde{y}\rangle|\tilde{x} + \tilde{y}\rangle \) [6].

As the Gaussian path loading operator \( \mathcal{G} \) takes the place of the operator \( \mathcal{P} \), it is imperative to talk about the construction of \( \mathcal{G} \). The authors of [6] use a Variational Quantum Eigensolver approach that uses a parameterized circuit, which in turn produce a parameterized quantum state \( |\psi(\theta)\rangle \), which approximately represents the target state \( |\psi_0\rangle \). The parameters \( \theta \) are updated using a classical optimizer that optimizes expectation value of a suitable cost function. This can be formulated by
\[
E_0 = \langle \psi_0 | H | \psi_0 \rangle,
\]

where energy expectation \( E \), reaches its ground state \( E_0 \), when calculated on a target state. It is noted that the energy Hamiltonian of the quantum harmonic oscillator is given by:
\[
H = \frac{p^2}{2m} + \frac{m(x - x_0)^2}{2},
\]

furthermore, by clever observation, it is noted that this Hamiltonian has a ground state energy
\[
\phi_0(x) = \left( \frac{m}{\pi} \right)^{1/4} e^{-m(x-x_0)^2},
\]
which is a Gaussian function. Here \(X\) is the position operator in real space, \(P = -i \frac{\partial}{\partial x}\) is the momentum operator, \(m\) is a parameter representing the variance of the Gaussian distribution, whereas \(x_0\) is the center of the distribution. By setting \(m = 1/(2\sigma^2)\), one can find the state \(\psi_a(x)\), such that \(\psi_a^2(x) = \mathcal{N}(x_0, \sigma)\).

In order to calculate these on a quantum computer one must observe that \(X^2\) is diagonal in the \(Z\)-basis, therefore it can be measured directly from the qubit register as \(N_{\text{counts}}\) wave function collapses on measurement. The \(P^2\) is diagonal on the momentum basis and therefore a centering Quantum Fourier Transform (QFT) is invoked after the state preparation block. Since the calculations are done on the discrete position space, the transformation \(x_i = -w + i\Delta x\) is used, where \(w = 0, \cdots, 2^n - 1\) and \(\Delta x = 2w/2^n\). Without loss of generality, if the center is chosen to be at zero, the energy \(E = E_{X^2} + E_{P^2}\) is calculated as follows:

\[
E_{X^2} = \frac{1}{N_{\text{shots}}} \sum_{j=0}^{N} \frac{m}{2} N_{\text{counts}}(j)(j \times \Delta x - x_0)^2
\]

\[
E_{P^2} = \frac{1}{N_{\text{shots}}} \sum_{j=0}^{N} \frac{1}{2m} N_{\text{counts}}(j)(j \times \Delta p)^2
\]

where, \(N_{\text{shots}}\) is the total number of circuit repetitions for the spacial and the momentum basis. Thus derivative pricing problem can be solved using re-parameterization method followed by payoff calculation efficiently with the help of arcsine and square root quantum operation. But there is no free lunch, hence, these methods have to deal with several errors which are discussed forthwith.

**Error Analysis**

The authors of [6] have characterized the errors induced in their approach as four main components: Truncation Error, Discretization Error, Amplitude Estimation Error and arithmetic error. First three are discussed here and the arithmetic error is discussed in the next subsection [3] for the sake of understanding.

(a) **Truncation Error.** Given the fault-tolerant setting of the algorithm presented in [6], to feasibly compute the infinitely many path integrals, the domain of integration is restricted to a range \([B_i, B_j]\), thereby leaving out a probability mass \(\alpha\). Given an upper bound \(f_\alpha = f_{\text{disc}} - f_{\text{max}}\) on the discounted payoff as calculated, and an upper bound \(P_{\text{max}}\) on the density function at each step, a truncation error of \(\epsilon_{\text{trunc}} = P_{\text{max}} f_\alpha\) is incurred. For \(dT\) different n-qubit registers that are \(w\) standard deviations around the mean for each timestep, the truncation error is given by:

\[
\epsilon_{\text{trunc}} \leq 2dT e^{-w^2/2}
\]

(b) **Discretization Error.** This error arises when approximating the integral over finite grid of points and can be reduced by increasing the number of qubits \(n\). Assuming \(\beta\) is an upper bound on the second derivative of the integrand and given \(n\) qubits, thereby discretizing the integral domain over \(2^{ndT}\) cells, the discretization error \(\epsilon_{\text{disc}}\) is given by:

\[
\epsilon_{\text{disc}} = \frac{\beta(B_n - B_j) dT^2}{24 \cdot 2^n}
\]

(c) **Amplitude Estimation Error.** An error of \(\epsilon_{\text{amp}}\) is incurred by the Amplitude Estimation subroutine when repeating the state preparation and pricing procedure \(1/\epsilon_{\text{amp}}\) times.

A key observation that lays the very foundation of this paper is that the algorithm heavily depends on quantum arithmetic operations in its various aspects of its calculation in the discrete computational domain. It is therefore imperative, that these operations be probed, thereby paving the way for more efficient calculation methods in the quantum discrete arithmetic operation domain. These quantum arithmetic operations are further elaborated in the subsequent subsection. Since our approach of solving derivative pricing uses higher dimensional quantum systems, an additional error apart from these four has been inducted automatically, which is thoroughly described in Section [5].

### 2.1 Quantum Arithmetic Operations

This section highlights the background for common quantum arithmetic operations. These operations are used in resource estimation and error analysis in derivative pricing as discussed in previous section. In this section, resources are usually reported as Toffoli gate cost as we are working in the fault-tolerant setting. As per the state-of-the-art work, they also estimate the optimized T-depth of the circuits by decomposing Toffoli gate in a Clifford + T gate set [24]. They decomposed the Toffoli gates with a T-depth of one using four ancilla qubits as shown in Fig. 2(a) and 2(b), one can obtain a representation of the Toffoli gate of T-depth 1 and overall depth 7. They have also tried to parallelize the resulting circuits wherever possible to obtain the optimized depth.

The authors of [6] performed all the calculations in fixed-point arithmetic similarly to [2], which allows us to use the described quantum techniques for reversible function evaluation. An n-bit representation of a number \(x\) is

\[
x = \underbrace{x_{n-1} \cdots x_{n-p}}_{p} \cdot \underbrace{x_{n-p-1} \cdots x_0}_{n-p}
\]

where \(x_i \in 0, 1\) denotes the \(i\)-th bit of the binary representation of \(x\) and \(p\) denotes the number of bits to the left of the binary decimal point. The choice of \(n\) and \(p\) controls the error that one can allow in each calculation, we allow the same approximation for arithmetic error as in [6]. Let \(\text{Toffoli\_count}\) and \(T\_\text{depth}\) denote the number of Toffoli gates and the T-depth required to compute an arithmetic function as per the state-of-the-art work.

**Addition/Subtraction:** As per [6], [25], one can perform addition of two \(n\)-qubit registers in place with a Toffoli cost,

\[
\text{Toffoli\_count}_{\text{add}} = 10n - 3w(n) - 3w(n-1) - 3 \log_2 n - 3 \log_2 (n-1) - 7
\]

where \(w(n)\) denotes the number of ones in the binary expansion of \(n\), and a Toffoli depth,
The T-depth of this algorithm as reported by the authors is

\[ T_{\text{depth}}(n, z) = \left[ \frac{n}{z} \right] \cdot T_{\text{depth}}(n, z) + T_{\text{depth add}}. \]  

\[(T_{\text{add}} + 6) \text{ is the T-depth of a controlled addition discussed in the Addition/ Subtraction section.}\]

Square Root: In [6], they employ the square root algorithm from [2], which has the Toffoli count,

\[ T_{\text{offoli count square}}(n, p, k, M) = k \left( \frac{3}{2} n^2 + n(3p + \frac{7}{2}) - 3(p - 1)p - 1 \right) \]

\[ + \frac{n^2}{2} + 11n + 2M d \left( 4 \left[ \log_2 M \right] - 8 \right) + 4M n - 2. \]  

The T-depth for computing \( \arcsin(\sqrt{x}) \) of a number \( x \) represented in a register of size \( (n, p) \), calculated as akin to the exponential is

\[ T_{\text{depth square}}(n, p, z) = T_{\text{depth square}}(n) + T_{\text{depth square}}(n, z) + 8n + 6, \]  

where \( T_{\text{depth square}}(n) = 5n + 3 \) is the T-depth for the square root algorithm from [27]. Next we portray our approach of intermediate qutrit on quantum arithmetic operations.

### 3 Quantum Arithmetic Operations with Intermediate Qutrits

This section first describes the Toffoli decomposition with the help of intermediate qutrit that we have adapted from [7]. Thenceforth, we discuss about one of the quantum...
arithmetic operations i.e., quantum adder with intermediate qutrits [28], [29]. Later we have exhibited quantum multiplier using intermediate qutrits with an example implementation. We have simulated this multiplier circuit on Google Colab platform [30] and the code is available at https://github.com/LegacYFTw/NTU. These two operations, adder and multiplier, are the basic building blocks of other arithmetic operations like square root, exponential and arcsine, which are used in derivative pricing. Therefore, we vividly discuss about quantum adder and multiplier and their implementation further in this section.

3.1 Toffoli Decomposition with Intermediate Qutrits

In [7], [31], the authors showed that we may occupy the |2⟩ state temporarily during computation, hence temporarily ternary. Maintaining binary input and output allows this circuit construction to be inserted into any pre-existing qubit-only circuits. A Toffoli decomposition via qutrits has been portrayed in Fig. 3 [7], [31]. More specifically the goal is to carry out a NOT operation on the target qubit (third qubit) as long as the two control qubits, are both |1⟩. First, a |1⟩-controlled X₀, where +1 is used to denote that the target qubit is incremented by 1 (mod 3), is performed on the first and the second qubits. This upgrades the second qubit to |2⟩ if and only if the first and the second qubits were both |1⟩. Then, a |2⟩-controlled X gate is applied to the target qubit. Therefore, X is executed only when both the first and the second qubits were |1⟩, as expected. The controls are reinstated to their original states by a |1⟩-controlled X₀ gate, which reverses the effect of the first gate. That the |2⟩ state from ternary quantum systems can be used instead of ancilla to store temporary information, which is the most important aspect in this decomposition. Thus, to decompose Toffoli gate, 3 generalized ternary CNOT gates are sufficient with circuit depth 3. In fact, no T gate is required.

![Fig. 3. An example of Toffoli decomposition with intermediate qutrit, where input and output are qubits. The red controls activate on |1⟩ and the blue controls activate on |2⟩. The first gate temporarily elevates q₁ to |2⟩ if both q₀ and q₁ were |1⟩. X operation is then only performed if q₁ is |2⟩. The final gate acts as a mirror of first gate and restores q₀ and q₁ to their original state |7⟩.](image)

3.2 Quantum Adder with Intermediate Qutrits

We have now demonstrated quantum adder circuit from [25], [29] on two four-bit registers A and B with a carry-out bit using ancilla as shown in Fig. 4. The sum S is computed in-place on register B while A is untouched and the ancilla are restored to |0⟩. In [29], for the first time, the authors showed how this adder can be efficiently implemented with the help of intermediate qutrits with respect to gate count and circuit depth instead of conventional way of decomposition. The author achieves an advanced adder by decomposing all the Toffoli gate presented in Fig. 3 with the discussed intermediate qutrit method. This work provides us enough motivation to work upon others quantum operations with intermediate qutrit. Therefore, we present an example of implementing a multiplier with the adoption of intermediate qutrit for better understanding.

3.3 Quantum Multiplier with Intermediate Qutrits

Let’s put some light on quantum multiplier here in this subsection. We have considered multiplication of 5 × 3 as an example to illustrate our approach of designing quantum multiplier with intermediate qutrit. In Fig. 5(a), a multiplier circuit has been presented considering the mentioned example as per [26], in which all the qubits are initialized with |0⟩. In this circuit, first five qubits (q₀ - q₄) are the input qubits, where first two qubits (q₀ and q₁) represent the number 3 by applying two NOT gates on them and other three qubits (q₂, q₃ and q₄) represent the number 5 by applying two NOT gates on qubits q₂ and q₄. Next, we perform multiply operation on these qubits by applying Toffoli gates and stores the value on ancilla qubits (q₅ - q₁₀). Now, we carry out addition by applying CNOT gates on ancilla qubits. Finally, to get the output of 5 × 3, we shall measure the qubits (q₅, q₁₀, q₁₁ and q₁₂). Further, all the Toffoli gates presented in Fig. 5(a) are decomposed using intermediate qutrit approach as shown in Fig. 5(b) to achieve asymptotic improvement of the circuit. Our numerical analysis also yields 5 × 3 = 15 perfectly. Since the circuits of quantum adder and multiplier are verified, the focus shifts to resource required to perform all the discussed quantum operations using intermediate qutrits.

4 Resource Estimation and Analysis

As shown in the proposed approach, we have synthesized the quantum arithmetic operations via intermediate qutrits, in which, three generalized ternary CNOT gates are used to decompose the Toffoli gate without any ancilla. Hence, no T gate has been used in any proposed circuit synthesis. Therefore, T-depth becomes zero in our proposed approach. The overall circuit depth for Toffoli decomposition has now become three as compared to seven. Since there are no T gates in our proposed circuits, the resource analysis is based on the count of generalized ternary CNOT gate, which is a Clifford gate. It also needs to be noted that the additional
four ancilla qubits are not required to decompose Toffoli gate, otherwise the number of logical qubits remain same for the derivative pricing as per [3], as no other optimization with respect to logical qubits is presented in this article. Hence, from now on we shall only discuss about gate count or circuit depth of derivative pricing problem.

Addition/Subtraction: As per our proposed approach, one can perform addition of two $n$-qubit registers in place with a CNOT-cost,

$$CNOT_{\text{count}_{\text{add}}} = 30n - 9w(n) - 9w(n - 1) - 9\log_2 n - 9\log_2 (n - 1) - 21 \quad (23)$$

where $w(n)$ denotes the number of ones in the binary expansion of $n$. For subtraction, the CNOT count remains same as addition.

Multiplication: For multiplication if one can follow our proposed method, then the CNOT count becomes,

$$CNOT_{\text{count}_{\text{mul}}}(n, p) = \frac{9n^2}{2} + 9np + \frac{9}{2}n - 9p^2 + 9p. \quad (24)$$

This method can also be used for division, where the CNOT count remains same as multiplication.

Square Root: If one can employ our proposed approach for the square root algorithm, then the CNOT count becomes,

$$CNOT_{\text{count}_{\text{sqrt}}}(n, p) = \frac{3n^2}{2} + 9n - 12. \quad (25)$$

Exponential: As per our proposed approach, the total number of ternary CNOT is given by

$$CNOT_{\text{count}_{\text{exp}}}(n, p, k, M) = \frac{9}{2}n^2 k + 9npk + \frac{21}{2}nk - 9p^2 d$$
$$+ 9pk - 3d + 6Md(4\log_2 M - 8) + 12Mn. \quad (26)$$

Arcsine: The total ternary CNOT count for computing (arcsin $\sqrt{5}$) is

$$CNOT_{\text{count}_{\text{arcsin}}}(n, p, k, M) =$$
$$3k \left( \frac{3}{2}n^2 + n(3p + \frac{7}{2}) - 3(p - 1)p - 1 \right) + \frac{3n^2}{2}$$
$$+ 33n + 6Md(4\log_2 M - 8) + 12Mn - 26. \quad (27)$$

4.1 Comparative Analysis

As discussed, the Toffoli decomposition with intermediate qutrit is novel not only for its depth optimization from 7 to 3 or gate count optimization from 25 to 3 as compared to [24], but also it does not require any T gates. Hence to execute required quantum arithmetic operations like arcsine and square root for derivative pricing, T-depth and T-cost become 0 for the proposed methodology as discussed in this article. Since the T gates are more expensive than other gates in terms of space and time cost due to their increased tolerance to noise errors, our method of implementing derivative pricing problem is superior to the method proposed in [3].

As there are no T gates in our methodology, the resources for all the arithmetic operations for derivative pricing are presented in the form of ternary CNOT gates, which is thoroughly discussed earlier in this section. It can be easily observed that the overall CNOT-cost for derivative pricing is much more improved as compared to [6], since they are decomposed their Toffoli gate with 16 CNOT gates and as per our decomposition it requires only 3 CNOT gates. As a case study of derivative pricing problem in [6], the authors have taken into consideration a basket autocallable (Auto) with 3 underlyings, 5 payment dates and a knock-in put option with 20 barrier dates. They found that the benchmark use case that they examined for re-parameterization method of derivative pricing, required T-cost of 12 billion, T-depth of 54 million and overall circuit depth at least 378 million. In our case, both of these T-cost and T-depth become 0, while the ternary CNOT-cost and overall circuit depth are 162 million respectively. These results evidently show that our method outperforms the run-time of the state-of-the-art algorithm for derivative pricing quite comprehensively since gate count and circuit depth have been significantly improved. Although our circuit constructions have adopted higher dimensional gates, our used decomposition scales favorably in terms of asymptotically fewer gate errors and idle errors as our gate count and circuit depth is asymptotically lower, which is elaborated upon in next section.
5 Error Analysis of Toffoli Decomposition with Intermediate Qutrits

Decoherence, noisy gates, and other sorts of errors can occur in any finite-dimensional quantum system. It has been proven that using higher dimensional states other than binary systems causes the system to have more errors. The impact of these inaccuracies on the Toffoli gate decomposition has been investigated in this section. Although the introduction of qutrits increases error, the overall error probability of the decomposition is lower than that of the earlier decomposition [24] since the number of ancilla qubits, gate count, and depth are reduced.

5.1 Generic Error Model

The conventional quantum error or noise model is for gate and relaxation error [7], which can be expressed by the Kraus Operator formalism [1]. If the density matrix representation of a (pure) quantum state is \( \rho = |\Psi\rangle \langle \Psi| \), the evolution of this state for any channel is represented as the function \( E(\rho) \):

\[
E(\rho) = E(|\Psi\rangle \langle \Psi|) = \sum_i K_i \rho K_i^\dagger
\]

where \( K_i \) are called the Kraus Operators, and \( K_i^\dagger \) is the matrix conjugate-transpose of \( K_i \), \( i \). The Kraus operator formulation can also be used to represent the evolution of a state under a noise model. The Kraus operators, for example, are simply the Pauli matrices in the depolarization noise model.

5.1.1 Gate Errors

In a binary quantum system with only one-qubit and two-qubit gates, there are four possible error channels for a one-qubit gate, which can be expressed as products of the two Pauli matrices, a NOT gate, \( X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \) and a phase gate, \( Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \). The possible error channels are: (i) no-error \( X^0Z^0 = I \), (ii) the phase flip which is the product \( X^0Z^1 \), (iii) the bit flip which is \( X^1Z^0 \) and (iv) the phase-bit flip channel given by \( X^1Z^1 \). We can express this one-qubit gate error model in the Kraus operator formalism in the following manner:

\[
E(\rho) = \sum_{j=0}^{1} \sum_{k=0}^{1} p_{jk} (X^jZ^k)\rho (X^jZ^k)^\dagger
\]

where \( p_{jk} \) is the probability of the corresponding Kraus operator.

A noisy gate is modelled as an ideal gate followed by an unwanted Pauli operator [32]. In other words, a one-qubit error is followed by an unwanted Pauli \( Z \) with probability \( p_1 \), \( p_2 \), \( p_3 \), respectively; and a two-qubit gate is followed by an unwanted Pauli \( \{X, Y, Z\} \) with probability \( p_i \cdot p_j \), where \( i, j \in \{x, y, z\} \). For the sake of convenience, we represent the one-qubit and two-qubit gate error probabilities as \( p_1 \) and \( p_2 \) respectively.

In a binary system, there are 2 types each of unwanted X and Z Pauli errors that can follow a one-qubit gate. Therefore, there are \( 2^2 - 1 \) ways (without considering the identity error) in which an error can occur after a one-qubit gate. If \( p_1 \) is the probability of a one-qubit Pauli error, then the evolution of the system under noisy one-qubit operations can be represented as in Eq. 30.

\[
E(\sigma) = (1 - (2^2 - 1)p_1)\sigma + \sum_{j,k \in\{0,1\}^2} p_{jk} K_{jk}\sigma K_{jk}^\dagger
\]

where \( K_{jk} \) represents the various Pauli operators.

Similarly, for two-qubit gates, an unwanted Pauli operator can occur on each of the two qubits after the gate operation. Therefore, there are \( 2^4 - 1 \) ways (excluding the identity operation on both the qubits) in which a gate can be noisy. If \( p_2 \) is the probability of two-qubit gate errors, then the evolution of the system under noisy two-qubit operations is represented as in Eq. 31.

\[
E(\sigma) = (1 - (2^4 - 1)p_2)\sigma + \sum_{j,k,l,m \in\{0,1\}^2} p_{jklm} K_{jklm}\sigma K_{jklm}^\dagger
\]

where \( p_{jklm} = p_{jk} \cdot p_{lm} \). The probability that the density matrix remains error free is independent of whether the underlying depolarizing channel is symmetric or asymmetric. Rather, it depends on the total probability of error.

Our used decomposition here deals with two-qutrit gates only on ternary quantum systems. In general, for the decomposition of a Toffoli gate, our used method uses up to 3 dimension. Therefore, for a 3-dimensional system, the error in our system scales as \( O(3^4) \) as shown in Eq. 32.

\[
E(\sigma) = (1 - (3^4 - 1)p_2)\sigma + \sum_{j,k,l,m \in\{0,1,2,3\}^4} p_{jklm} K_{jklm}\sigma K_{jklm}^\dagger
\]

In Table 1, we show the decrease in the probability of no-error for two-qutrit gates due to the usage of higher dimensions for ternary systems.

TABLE 1

| Dimension \( d \) | Without our decomposition | With our decomposition |
|-------------------|--------------------------|-----------------------|
| \( 2 \)           | \( 1 - 12p_2 \)          | \( 1 - 81p_2 \)       |

5.1.2 Idle error:

In quantum devices, idle errors mainly focus on the relaxation from higher to lower energy levels. Amplitude damping is another name for this. This noise channel takes qutrits to lower states in an irreversible manner. For qubits, the only amplitude damping channel is from \( |1\rangle \) to \( |0\rangle \), and we denote this damping probability as \( \lambda_1 \). For qubits, the Kraus operators for amplitude damping are:

\[
K_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1 - \lambda_1} \end{pmatrix} \quad \text{and} \quad K_1 = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{\lambda_1} \end{pmatrix}
\]

For qutrits, we also model damping from \( |2\rangle \) to \( |0\rangle \), which occurs with probability \( \lambda_2 \). For qutrits, the Kraus operator for amplitude damping can be modeled as:
In each Kraus Operator $K_t$, the value of $\lambda_t \propto \exp(-t/T_{1_t})$, where $t$ is the duration of the computation, and $T_{1_t}$ are the relaxation time. We have qubit quantum devices, where $T_{1_t} \approx 100\mu s$ in some higher end IBM Quantum Devices [33]. However, due to the lack of qudit quantum computers, we do not have explicit values of other $T_{1_t}$'s except $30\mu s$ for qutrit ($T_{1_2}$) quantum devices [34]. Nevertheless, the length of time depends on the circuit depth. As a result, by reducing the circuit depth, idle error are reduced. Therefore, since depth has been optimized, the decoherence owing to our employed decomposition is significantly lower than the previous decomposition.

### 5.2 Analysis of Success Probability

In [7], the authors advocated the use of higher dimension for efficient decomposition of Toffoli gates. However, because there was no qudit hardware at the time, they assumed that the value of $T_{1_t}$ for ternary systems is the same as for binary systems in that article. However, as explained in the preceding article, currently we have the value of $T_{1_t}$ for ternary systems. As a result, in this section, we apply the method in [6, 24] and our method in this article to investigate the probability of success in the decomposed circuit of a Toffoli gate. It is worth noting that, although the decomposition of [6, 24] requires one-qubit and two-qubit gates, our method just necessitates two-qutrit ternary gates.

The complexity of decomposition of a Toffoli gate in terms of the number of gates and depth of the circuit for the method in [6, 24] and our used one are depicted here. For each Toffoli decomposition as shown in Fig. 6, one requires 7 one-qubit gates and 16 two-qubit gates with overall circuit depth 7, whether we need 3 two-qutrit gate with circuit depth 3 for our used decomposition.

Small errors in quantum circuit gates can be described as an ideal gate followed by an undesirable Pauli operator, as discussed in previous subsections. However, instead of comparing the probability of minor errors in the circuit, we compare the probability that the circuit remains error-free (probability of success) as per [35] for the decomposition in [6, 24] and our employed decomposition, without losing generality.

For any decomposition, the generalized formula for probability of success ($P_{success}$) described in [35, 36] is the product that the individual components does not fail. In other words,

$$P_{success} = \prod_{\text{gates}} (P_{success \ of \ gate})^{\# \ gates} \times e^{-\left(\text{depth}(T_{1_t})\right)},$$

where the first term’s product is the likelihood of all types of gates employed in the decomposition (one-qubit, two-qubit, two-qutrit), and the second term is the probability of no relaxation error. Due to the zero in the power, when a specific type of gate is not employed in a decomposition, the associated term has a value of 1. For example, our suggested decomposition does not require a one-qutrit gate, therefore that term’s contribution to the product is 1.

Current quantum devices are mostly binary, and the probabilities of one-qubit and two-qubit gates in the IBMQ Quantum Devices are in the range of $10^{-4}$ and $10^{-2}$ respectively [33]. Moreover, the time $T_{1_t}$ of most of the IBM Quantum Devices are in the range of $100\mu s$. However, in [34], the authors experimentally showed that the value of $T_{1_t}$ for each ternary gate is $30\mu s$, which we have also assumed for our study. We assume that the probability of error of each two-qubit and two-qutrit gate is $10^{-2}$, that of one-qubit gate is $10^{-4}$, and the time $T_{1_t}$ is $100\mu s$ and $T_{1_2}$ is $30\mu s$ for our simulation.

In Fig. 6, we exhibit the probability of success for the Toffoli gate decomposition using the method of [6, 24] (which we label as ‘conventional decomposition’ and our used method. We find that our proposed technique produces much fewer errors than the decomposition in [6, 24]. This is due to the fact that our decomposition has fewer gates and is shallower. Although our decomposition employs a few qutrit gates, which have a higher error probability due to the curse of dimensionality, our technique is superior due to the overall large reduction in gate count and depth. In fact, for Toffoli count 30 in a circuit, our used decomposition has a probability of success of $\approx 0.4$, whereas that of [6, 24] has a probability of success of $\approx 0.01$. Therefore, we obtain a percentage decrease in the probability of error by $\approx 40\%$ for Toffoli count 30. Thus, it can be concluded that the conventional decomposition methods attain $100\%$ error for Toffoli count 30 in a circuit, whereas our used decomposition method yields less erroneous results for Toffoli count 30 and above in a circuit. A comparative study of our Toffoli decomposition used in derivative pricing with some previous works [6, 24] is shown in Table 2. Our work outperforms them in terms of T-depth, overall circuit depth, overall gate count of the circuit and probability of error.

### 6 Conclusion

In this work, we have addressed basic quantum arithmetic operations and estimated the resources using intermediate qutrits which also demonstrate that our approach outperforms current techniques in terms of circuit depth and cir-
circuit robustness. Further, as an application, we have shown a novel approach of implementing derivative pricing problem by decomposing Toffoli gate into two-qutrit gates with optimized gate count and depth without using any ancilla qubit. We have given a comparative study on resources for derivative pricing to establish that our approach is better than the existing state-of-the-art one [6]. Finally, we have studied the effect of different error models on this decomposition technique. Our study shows that the few gates in higher dimensional quantum systems which are used in the used decomposition, are prone to more errors. Nevertheless, as we have obtained improved gate count and circuit depth, leading to low total error probability, the gates can operate with high fidelity as compared to state-of-the-art work. Through numerical simulation, as an instance, we have shown that intermediate qutrit-based Toffoli decomposition obtained a percentage increase in the probability of success by ∼ 40% for Toffoli count 30 in a circuit as compared to conventional way of Toffoli decomposition. These improved quantum arithmetic operations pave the way for more detailed time improvement estimate with intermediate qutrid to study the decomposition for other quantum circuits.

**References**

[1] M. A. Nielsen and I. Chuang, "Quantum computation and quantum information," 2002.

[2] T. Häner, M. Roetteler, and K. M. Svore, “Optimizing quantum circuits for arithmetic," 2018. [Online]. Available: https://arxiv.org/abs/1805.12445

[3] V. Rebizent, B. Gupt, and T. R. Bromley, "Quantum computational finance: Monte Carlo pricing of financial derivatives," Phys. Rev. A, vol. 98, p. 022321, Aug 2018. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.98.022321

[4] A. Martin, B. Candelas, A. Rodríguez-Rozas, J. D. Martín-Martín, G. Brassard, P. Høyer, M. Mosca, and A. Tapp, “Quantum amplitude amplification and estimation,” pp. 53–74, 2002. [Online]. Available: https://doi.org/10.1023/FQom.0000020521

[5] A. Montanaro, “Quantum speedup of monte carlo methods,” Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 471, no. 2181, p. 20150301, Sep 2015. [Online]. Available: https://doi.org/10.1098/rspa.2015.0301

[6] G. Crawford and B. Sen, Derivatives for decision makers. Wiley, 1996.

[7] Y. Suzuki, S. Uno, R. Raymond, T. Tanaka, T. Onodera, and N. Yamamoto, “Amplitude estimation without phase estimation,” Quantum Information Processing, vol. 9, no. 2, p. 75, Jan 2020. [Online]. Available: https://doi.org/10.1007/s11128-019-2565-2

[8] D. Grinko, J. Gacon, C. Zoufal, and S. Woerner, “Iterative quantum amplitude estimation,” npj Quantum Information, vol. 7, no. 1, p. 52, Mar 2021. [Online]. Available: https://doi.org/10.1038/s41534-021-00379-1

[9] T. Tanaka, Y. Suzuki, T. Onodera, and N. Yamamoto, “Amplitude estimation via maximum likelihood on noisy quantum computer,” Quantum Information Processing, vol. 20, no. 9, sep 2021. [Online]. Available: https://doi.org/10.1007/s11128-021-03215-9

[10] T. Giurgica-Tiron, I. Kerenidis, F. Labib, A. Prakash, and W. Zeng, “Low depth algorithms for quantum amplitude estimation,” 2020. [Online]. Available: https://arxiv.org/abs/2002.07534

[11] P. Selinger, “Quantum circuits of depth one,” Phys. Rev. A, vol. 87, p. 042302, Apr 2013. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.87.042302

[12] Y.-M. Di and H.-R. Wei, “Toward pricing financial derivatives with an IBM quantum computer,” Frontiers in Physics, vol. 8, Nov 2020. [Online]. Available: http://dx.doi.org/10.3389/fphy.2020.089504
[31] J. M. Baker, C. Duckering, P. C. Gokhale, N. C. Brown, K. R. Brown, and F. T. Chong, “Improved quantum circuits via intermediate qutrits,” *ACM Transactions on Quantum Computing*, vol. 1, no. 1, Oct 2020. [Online]. Available: https://doi.org/10.1145/3406509.

[32] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, “Surface codes: Towards practical large-scale quantum computation,” *Physical Review A*, vol. 86, no. 3, p. 032324, 2012.

[33] “IBM Quantum,” https://quantum-computing.ibm.com/, 2021.

[34] L. E. Fischer, D. Miller, F. Tacchino, P. K. Barkoutsos, D. J. Egger, and I. Tavernelli, “Ancilla-free implementation of generalized measurements for qubits embedded in a qudit space,” 2022. [Online]. Available: https://arxiv.org/abs/2203.07209.

[35] R. Majumdar, D. Madan, D. Bhoumik, D. Vinayagamurthy, S. Raghunathan, and S. Sur-Kolay, “Optimizing ansatz design in qaoa for max-cut,” *arXiv preprint arXiv:2106.02812*, 2021.

[36] A. Saha, R. Majumdar, D. Saha, A. Chakrabarti, and S. Sur-Kolay, “Asymptotically improved circuit for $d$-ary grover’s algorithm with advanced decomposition of $n$-qudit toffoli gate,” 2020. [Online]. Available: https://arxiv.org/abs/2012.04447.