

Article

# A Quantum Algorithm for Pricing Asian Options on Valuation Trees

Mark-Oliver Wolf , Roman Horsky and Jonas Koppe 

Department of Financial Mathematics, Fraunhofer ITWM, Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany

\* Correspondence: mark-oliver.wolf@itwm.fraunhofer.de

**Abstract:** We develop a novel quantum algorithm for approximating the price of a discrete floating-strike Asian option based on an underlying valuation tree. The paths of the tree are encoded in bit-representation into a qubit register, where quantum state preparation is used to load the corresponding distribution onto the states. We implement the expectation value of the option pricing formula as a composition of the price probabilities, the payout and an indicator function, mapping their respective values to amplitudes of additional qubits. Thus, the underlying no longer has to be discretized into the same bit values for different times, resulting in smaller quantum circuits. The algorithm may be used with quantum amplitude estimation, enabling a quadratic speed-up over classical Monte Carlo methods.

**Keywords:** quantum computing; option pricing; tree methods



**Citation:** Wolf, Mark-Oliver, Roman Horsky, and Jonas Koppe. 2022. A Quantum Algorithm for Pricing Asian Options on Valuation Trees. *Risks* 10: 221. <https://doi.org/10.3390/risks10120221>

Academic Editor: Hailiang Yang

Received: 20 September 2022

Accepted: 14 November 2022

Published: 22 November 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Due to the progress in developing quantum computers, the research in the field of industrial applications has intensified in recent years. One important branch of industry is the financial sector, in which a number of very complex and computer-intensive applications can be found, e.g., the evaluation of insurance portfolios or the pricing of structured financial products. Accordingly, the use of a quantum computer for both applications is investigated, in e.g., Egger et al. (2021). In the field of derivative pricing, the article by Rebentrost et al. (2018) is considered to be one of the first influential works. They present a way to price plain vanilla and Asian options in a Black–Scholes framework using quantum computing. In theory, quantum computers have the potential for a quadratic speed-up with respect to the number of computational steps compared to classical Monte Carlo methods used for option pricing. Because current quantum hardware is still small, many quantum algorithms are of hybrid character, meaning a mixture of classical computer applications and quantum computations. Therefore quantum speed-up must still be shown. Stamatopoulos et al. (2020) focus their research on the possibilities and performance of the currently available noisy intermediate-scale quantum (NISQ, c.f. Preskill (2018)) hardware to price multi-asset and path-dependent options. Especially latter ones are subject to further research, as they are harder to price than path-independent ones and therefore more promising for the speed-up that quantum computers promise. Likewise, in a Black–Scholes framework, Chakrabarti et al. (2021) present an algorithm which precomputes standard normal distributions into quantum states and uses affine transformations to obtain the asset's path-dependent return distributions. In addition to this new method, they also derive bounds on required resources for established quantum algorithms in finance to reach a practical quantum advantage. Another approach, followed by Miyamoto and Kubo (2021), is to solve partial derivative equations (PDE) for option prices by using the finite difference method. They define grid points for the logarithmic value of the underlying at maturity and calculate its value by using a discretized operator based on the PDE to describe the evolution. The quantum computer is used to calculate the resulting systems

of equations. Pricing multi-dimensional derivatives with the help of discretizing their price PDEs is a method [Kubo et al. \(2022\)](#) use as well. The novelty being that a variational quantum algorithm (VQA) is used to calculate the development of the underlying. This allows obtaining helpful results even for small-scaled quantum computers with only a few qubits available. To demonstrate the possibilities of the currently small quantum computers, [Woerner and Egger \(2019\)](#) made an application study using real hardware to calculate the risk measures of a financial portfolio. Since the resources of current quantum computers are quite limited, simplifications in the financial models are necessary. One such method is the principal component analysis where a quantum alternative exists as well. [Martin et al. \(2021\)](#) use a Heath–Jarrow–Morton framework for modeling forward rates and in doing so show how the model is adapted to the quantum computer. The application of a quantum gradient estimation algorithm to calculate financial indicators of sensitivity, so-called Greeks, is illustrated by [Stamatopoulos et al. \(2022\)](#). They present different approaches and investigate their complexity with respect to the requirements of quantum computers. We adopt a similar approach as in [Blank et al. \(2021\)](#). They consider a discrete stochastic process and develop a quantum algorithm to calculate its characteristic function. This derives the whole probability distribution for the stochastic trajectories resulting in a valuation tree approach for pricing derivatives. The work by [Vazquez and Woerner \(2021\)](#) also has to be mentioned, who show an efficient way to prepare quantum states when employing amplitude estimation and present their findings on a Heston model for option pricing using real hardware.

The presented research shows that quantum computing becomes more and more relevant for financial questions as the development of the hardware progresses. Currently, simulators of quantum computers and real quantum computers are used to make first experiences with this technology and learn how classically well understood problems can be solved with the help of the quantum technology. In our work, we aim at contributing to this development in this exact way. As a use case, we choose an Asian option, which puts some challenges to pricing due to its path-dependence. Asian options, sometimes equivalently called traded average price options (TAPOs), are derivatives that are commonly used in the energy and metal sector for hedging purposes. Using this as an example, we show how the classical approach of a valuation tree can be decomposed into three modules in order to be adapted to the requirements of a quantum computer. Each of the three calculation steps individually can already be evaluated on real quantum computers. As the hardware is still not powerful enough for the whole problem at once, our evaluations are performed with the help of a simulator for a quantum computer. Nevertheless, our results show that in the future, the whole algorithm will be applicable on a real quantum computer. The decomposition of the algorithm comprises the calculation of the underlying's probability distribution, the evaluation of the payoff and the encoding of an indicator function.

This new approach results in substantially less qubits used per time step. By encoding the underlying as probabilities and not in a bit representation, the derivative can be priced in a very efficient way. The paper is structured as follows. After presenting the mathematically formulated problem and the general ideas of our algorithm to solve this problem in Section 2, we give a quick overview on the underlying techniques used on a quantum computer in Section 3. Afterwards, in Section 4, we present our implementation in detail. We finish our research by illustrating the results of the entire algorithm for different problem sizes on a quantum computer simulator in Section 5. The paper closes with a discussion of our work in Section 6.

## 2. Problem and Algorithm

Let  $(S(t))_{0 \leq t \leq T}$  be an underlying, stochastic security price process in a financial model with arbitrary, but known, distribution. Assume the financial model to be free of arbitrage and that there exists some equivalent martingale measure (EMM)  $\mathbb{Q}$ . Additionally, let the first moment of  $S(t)$  exist (at least under  $\mathbb{Q}$ ). Then, the fair price of a discrete floating-strike

Asian option with maturity  $T$  and  $n$  time steps  $0 = t_0 \leq t_i < t_{i+1} \leq t_n = T, i \in \{1, \dots, n\}$  is given by (see, e.g., Korn et al. (2010))

$$C(t) = \mathbb{E}_{\mathbb{Q}} \left[ \frac{B(t)}{B(T)} \left( S(T) - \frac{1}{n} \sum_{i=1}^n S(t_i) \right)^+ \middle| \mathcal{F}_t \right], \tag{1}$$

where  $\mathbb{Q}$  is the risk-neutral EMM,  $(\mathcal{F}_t)_{0 \leq t \leq T}$  is a suitable filtration, and  $1/B(t)$  is the deterministic discount factor at time  $t$ . For simplicity, we assume a zero-interest situation,  $B(t) = 1$  for all  $t \geq 0$ . Additionally, as only the risk-neutral measure  $\mathbb{Q}$  is relevant for our computations, for clarity, we may assume that  $\mathbb{P} = \mathbb{Q}$  for some physical measure  $\mathbb{P}$ . To summarize, our objective is the computation of

$$C(0) = \mathbb{E} \left[ \left( S(T) - \frac{1}{n} \sum_{i=1}^n S(t_i) \right)^+ \right]. \tag{2}$$

Be aware that a fixed-strike Asian option can be priced with the upcoming algorithm as well as it is less complex using a constant strike  $K$  instead of the variable  $S(T)$ .

*General Idea*

Recall that the dynamics of the underlying security  $S(t)$  are known. We will now approximate this security by using a valuation tree with two distinct discretizations. The first discretization will be completed with respect to time, where we will only use the time steps  $t_i$  needed for the Asian option, using the index  $i$  to refer to this for the remainder. The second discretization has to be completed in the range of our underlying for each respective time step. Here, at time  $t_i$ , the underlying  $S(t_i)$  will only be able to equal a finite amount of values  $S^k(t_i), k \in \{0, \dots, N_i - 1\}$ , where  $N_i \in \mathbb{N}$  is the number of discretization points, i.e., nodes. Even though the discretization may change depending on  $t_i$ , we write  $k$  instead of  $k(t_i)$  for notational simplicity. Generally, the index  $k$  will be used to refer to all discretizations that do not refer to time. We combine different layers by defining transition probabilities  $\mathbb{P}(S^k(t_i) \mid S(t_{i-1}), \dots, S(t_0))$  to approximate the continuous underlying.

One example for a valuation tree would be the Cox–Ross–Rubinstein model by Cox et al. (1979), where a binomial tree is used to approximate a log-normally distributed underlying. It has to be stated that for our implementation, we are not only limited to binomial trees; all scenario trees, with discretely many edges, may be employed.

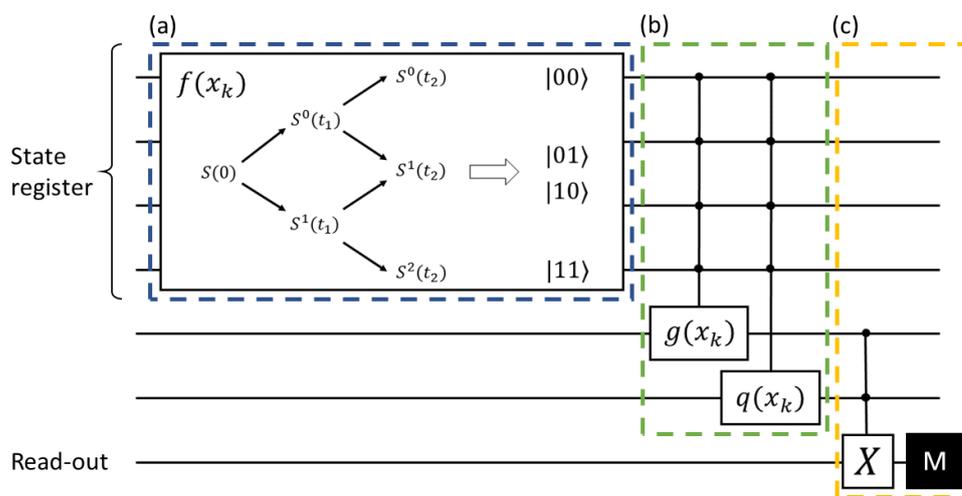
We denote each possible distinct path through the valuation tree with  $x_k = (S^{k_1}(t_1), \dots, S^{k_n}(t_n))$  and  $k_i \in \{0, \dots, N_i - 1\}$ . Consequently, there exist  $\prod_{i=1}^n N_i$  paths. Let  $N = \lceil \log_2(\prod_{i=1}^n N_i) \rceil$  be the number of qubits needed to represent all possible paths. If  $N \neq \log_2(\prod_{i=1}^n N_i)$ , we may set the upcoming probability of the respective additional paths  $f(x_k) = 0$  for  $\log_2(\prod_{i=1}^n N_i) < k < 2^N$ . This now enables us to approximate Equation (1) via

$$C(0) = \mathbb{E} \left[ \left( S(T) - \frac{1}{n} \sum_{i=1}^n S(t_i) \right)^+ \right] \approx \sum_{k=0}^{2^N-1} \underbrace{f(x_k)}_{\text{probability}} \underbrace{g(x_k)}_{\text{payoff}} \underbrace{q(x_k)}_{\text{positive part as condition}}, \tag{3}$$

where the different functions are accordingly defined as

$$\begin{aligned} f(x_k) &= \mathbb{P} \left( (S(t_0), \dots, S(t_n)) = x_k \right) = \prod_{i=1}^n \mathbb{P}(S^k(t_i) \mid S(t_{i-1}), \dots, S(t_0)), \\ g(x_k) &= S^{k_n}(T) - \frac{1}{n} \sum_{i=1}^n S^{k_i}(t_i), \\ q(x_k) &= \mathbf{1}_{\{g(x_k) > 0\}}. \end{aligned} \tag{4}$$

We present a visualization of the proposed algorithm in Figure 1. We start by computing the possible values of the underlying in our scenario on a classical computer. Then, an  $N$  qubit state register is allocated to represent all paths of the underlying that may possibly occur during an evolution period comprising  $n$  time steps, as shown in Figure 1a. The qubits' weights (amplitudes) are prepared to correspond to the respective asset probabilities denoted by the function  $f$ . The functions  $g$  and  $q$  for the payoff are then implemented separately as unitary transformations controlled by the  $N$  qubit state register in Figure 1b. In Figure 1c, they are recombined to calculate the scaled Asian option price, which can be read out by repeated measurements. A post-processing step ensures correct re-scaling. Using quantum parallelism, the option payoffs of all paths can be evaluated simultaneously.



**Figure 1.** A visualization of the upcoming algorithm as a quantum circuit model. The horizontal lines represent the used qubits. The black circles show the function's dependence on the respective qubits. (a) The given valuation tree is mapped to a qubit register, encoding in the qubit states the asset's path values  $x_k$ . The underlying's path probabilities  $f(x_k)$  are associated with the qubit's state probabilities. (b) Depending on the path, the payoff  $g(x_k)$  and indicator function  $q(x_k)$  are implemented on different qubits. (c) The functions are combined on a read-out qubit and finally measured.

### 3. Quantum Programming

#### 3.1. Qubits

A qubit  $|\varphi\rangle_q$  is a two-dimensional vector  $(a, b)^T \in \mathbb{C}^2$  with the condition that  $|a|^2 + |b|^2 = 1$ . We may write the qubit in Dirac notation as

$$|\varphi\rangle_q = a|0\rangle_q + b|1\rangle_q, \tag{5}$$

where the index denotes the specific qubit,  $|0\rangle = (1, 0)^T, |1\rangle = (0, 1)^T$  are the computational basis states and  $\varphi$  represents a combination of them. We call  $a$  and  $b$  the amplitudes of their respective states, and the probability of measuring the qubit in the respective state is given by the squared, absolute value of the amplitude.

A quantum register  $|\phi\rangle_s \in \otimes_{i=1}^N \mathbb{C}^2$  consists of  $N$  qubits and may be represented as the sum of its computational basis states

$$|\phi\rangle_s = \sum_{k=0}^{2^N-1} a_k |k\rangle. \tag{6}$$

Here,  $|k\rangle$  is the binary representation of integer  $k$ . For additional clarity, we sometimes write the number of qubits as an upper index, e.g., we write  $|0^{\otimes N}\rangle$  for an  $N$ -qubit register

with each qubit being in basis state  $|0\rangle$ . The condition on the amplitudes for a quantum register accordingly changes to  $\sum_{k=0}^{2^N-1} |a_k|^2 = 1$ .

### 3.2. Quantum Circuits

A quantum circuit is a model for quantum computations where a computation is a sequence of gates applied to a set of qubits akin to a bit-based circuit used in classical computing. Quantum gates are algorithmic steps to the qubits and mathematically expressed by applying unitary matrices to the vector representation of the qubits. In this paper, we will primarily make use of the Hadamard gate  $H$  used for creating an equal superposition. Moreover the  $X$ -gate which switches amplitudes, the  $\mathcal{R}_Y$ -gate and their conditional variants  $cU \in \{cX, c\mathcal{R}_Y\}$  are used respectively. They are defined as:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \mathcal{R}_Y(2\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}, \quad cU = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & & U \\ 0 & 0 & & \end{pmatrix}. \quad (7)$$

The controlled operations act on two qubits and are therefore represented by  $4 \times 4$  matrices. The included single-qubit transformation  $U$  is only applied to the target qubits if the control qubit is found in state 1.

### 3.3. Measurement

Determining the state of a qubit is accomplished by repeated measurements. In quantum computing, we generally refer to measurements with respect to the computational basis states, but theoretically, other measurement bases are possible. When a qubit  $|\cdot\rangle_t$  is measured, its probability of being in state  $|1\rangle$  is given by

$$P(U|0\rangle_t = |1\rangle) := |\langle 1|U|0\rangle_t|^2, \quad (8)$$

where we defined a new notation for easier readability. Here,  $U$  is the matrix corresponding to the used quantum circuit. It is now easy to see that applying the operator  $\mathcal{R}_Y(2\theta)$  for  $\theta = \arcsin(\sqrt{f})$  will generate the result of  $P(\mathcal{R}_Y(2\theta)|0\rangle_t = |1\rangle) = f$ . Because of this, we will restrict ourselves to the use of  $\mathcal{R}_Y$ -gate for ease of understanding. Circuit optimizations employing complex-valued rotations may be possible.

### 3.4. Arithmetics with Amplitudes

Almost every quantum circuit is built up of smaller subcircuits, each implementing a different part of the computation. It can generally be distinguished between two different approaches of data encoding when studying these subcircuits: an amplitude-based approach and a bit-based approach. The bit-based approach encodes data in multiple qubits, where after measurement, each qubit can be viewed as a bit (due to it collapsing to either 0 or 1), similar to classical computing. The amplitude-based approach encodes data as amplitudes, making the information available by measuring the number of times a certain state happens. By doing this, there is no restriction on some discretization that has to be chosen for the bit-encoding. Hence, we may encode arbitrary values of our underlying for different time steps and use arithmetics on an amplitude level to combine them precisely. Therefore, we are quickly going to state one of the main results used for the presented algorithm, the addition of amplitudes as presented in [Vazquez and Woerner \(2021\)](#).

For  $n$  operators  $\mathcal{F}_i$  generating the state

$$\mathcal{F}_i |x\rangle_s |0\rangle_t = |x\rangle_s \left( \sqrt{1-f_i} |0\rangle_t + \sqrt{f_i} |1\rangle_t \right), \quad i = 1, \dots, n, \quad (9)$$

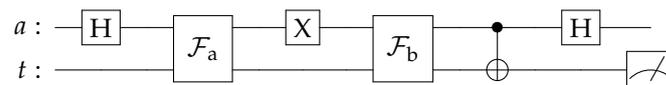
we are able to construct quantum circuits  $\mathcal{F}_+$  where the probability of our target qubit  $|\cdot\rangle_t$  being in state  $|1\rangle$  is given by

$$P(\mathcal{F}_+ |0\rangle_t = |1\rangle_t) = \frac{1}{n} \sum_{i=1}^n f_i. \tag{10}$$

We can also implement a subtraction of amplitudes by employing another  $cX$ -gate. In other words, for quantum circuits  $\mathcal{F}_a$  and  $\mathcal{F}_b$  implementing a qubit to be measured in state  $|1\rangle$  with a probability of  $f_a$  and  $f_b$ , respectively, we may construct  $\mathcal{F}_-$  with

$$P(\mathcal{F}_- |0\rangle_t = |1\rangle_t) = \frac{1}{2}(1 + f_a - f_b). \tag{11}$$

Note that we can retrieve the needed addition or subtraction by proper rescaling of the resulting probability. An example circuit for  $\mathcal{F}_-$  is given in Figure 2.



**Figure 2.** Visualization of the quantum circuit implementing the operator  $\mathcal{F}_-$  for calculating the difference of two functions  $f_a, f_b$  implemented by the operators  $\mathcal{F}_a, \mathcal{F}_b$ . Note that both functions  $f_a, f_b$  may also depend on some state register  $|x_k\rangle$  if needed. The last Hadamard application uncomputes the qubit  $|\cdot\rangle_a$ .

### 4. Quantum Algorithm

#### 4.1. Encoding the Paths

For the implementation of the algorithm on a quantum computer, we have to find a mapping of all possible paths through the valuation tree to a qubit register. The employed qubit register will be referred to as the state register for the remainder of this article, and it consists of  $N$  qubits. Assume that in our valuation tree, all nodes of the previous level are connected to the nodes of the current level. One example of such a mapping would be to represent the possible values of  $S^k(t_i)$  with a state  $|k\rangle_{t_i}$  (in binary representation) of a qubit register. Doing this for all time steps  $t_i$ , we may combine them to denote the entire state register by  $|s\rangle = \otimes_{i=1}^n |k_i\rangle_{t_i} = \sum_{k=0}^{2^N-1} \alpha_k |x_k\rangle$ , where  $|x_k\rangle$  are the computational basis states. Each  $|x_k\rangle$  can then be understood as a path taken through the approximating tree consisting of the individual  $S^{k_i}(t_i)$ .

In the following, we will present how each of the functions in Section 1 may be implemented as a quantum circuit with a single path  $x_k$  given as an input state. If properly constructed, we can eventually set our state register  $|s\rangle$  as a superposition over all the paths  $|x_k\rangle$  to obtain the simultaneous computation.

#### 4.2. Encoding the Probabilities

Assume we have an operator  $P$  that does the following transformation to our state register:

$$P |0^{\otimes N}\rangle = \sum_{k=0}^{2^N-1} \sqrt{f(x_k)} |x_k\rangle = \sum_{k_1, \dots, k_n} \prod_{i=1}^n \sqrt{\mathbb{P}(S^{k_i}(t_i) | S(t_{i-1}), \dots, S(t_0))} |k_1\rangle_{t_1} \cdots |k_n\rangle_{t_n}. \tag{12}$$

Then, the path probabilities are directly encoded in the amplitude of their respective state, and operators mapping from the state register have the needed probability. Constructing this operator  $P$  as a quantum circuit is generally referred to as quantum state preparation, and for arbitrary functions  $f$ , it requires an exponential number of gates (c.f. Shende et al. 2006; Plesch and Brukner 2011). Finding operators efficiently implementing  $P$  is a widely researched problem with various algorithms available, e.g., Grover and Rudolph (2002), Zoufal et al. (2019), Holmes and Matsuura (2020), and Zhang et al. (2021). Therefore, let us assume an appropriate operator to be given.

### 4.3. Encoding the Payoff

As previously explained, the algorithm uses the classically computed  $S(t_i)$  for the implementation of the option value. To encode them on a qubit amplitude, we need to ensure that all values are in  $[0, 1]$ . This can be completed by using an appropriate scaling technique. For this purpose, let  $M := \max_{t_i, k} S^k(t_i)$  and define the transformation

$$S_{\text{norm}}^k(t_i) := \frac{S^k(t_i)}{M} \in [0, 1]. \tag{13}$$

We first focus on implementing the mean of our underlying. Using uniformly controlled rotations, a technique introduced by Möttönen et al. (2004) and further developed by Shende et al. (2006), we may construct operators  $G_{t_i}$  implementing

$$G_{t_i} |k\rangle_{t_i} |0\rangle = |k\rangle_{t_i} \left( \sqrt{1 - S_{\text{norm}}^k(t_i)} |0\rangle + \sqrt{S_{\text{norm}}^k(t_i)} |1\rangle \right). \tag{14}$$

Employing Equation (10), we can construct an operator  $G_M$  with

$$G_M |x_k\rangle |0^{\otimes N_a}\rangle_a |0\rangle_g = |x_k\rangle \frac{1}{\sqrt{n}} \sum_{i=0}^n \left[ |i\rangle_a \left( \sqrt{1 - S_{\text{norm}}^k(t_i)} |0\rangle_g + \sqrt{S_{\text{norm}}^k(t_i)} |1\rangle_g \right) \right], \tag{15}$$

where  $N_a = \lceil \log_2(n) \rceil$  is the amount of additional ancillary qubits needed. These are qubits that are only used for auxiliary purposes and are not directly depending on the input. Often, they are reusable at a later point when employing an uncomputing process.

Furthermore, using Equation (11),  $G_M$ , and two additional qubits  $|0\rangle_b$  and  $|0\rangle_g$ , we are able to build an operator  $G$  with

$$G |x_k\rangle |0\rangle_b |0^{\otimes N_a}\rangle_a |0\rangle_g = |x_k\rangle \frac{1}{\sqrt{2}} \left\{ |0\rangle_b \frac{1}{\sqrt{n}} \sum_{i=1}^n |i\rangle_a \left( \sqrt{S_{\text{norm}}^k(t_i)} |0\rangle_g + \sqrt{1 - S_{\text{norm}}^k(t_i)} |1\rangle_g \right) + |1\rangle_b \frac{1}{\sqrt{n}} \sum_{i=1}^n |i\rangle_a \left( \sqrt{1 - S_{\text{norm}}^k(T)} |0\rangle_g + \sqrt{S_{\text{norm}}^k(T)} |1\rangle_g \right) \right\}. \tag{16}$$

Note that we constructed  $G$  in such a way that all the necessary information is loaded on state  $|1\rangle_g$ . Combining  $G$  with the probability loading operator  $P$  from (12), we obtain the following probability of measuring the success state  $|1\rangle_g$ :

$$\frac{1}{2} \sum_{k=0}^{2^N-1} f(x_k) \left( 1 + S_{\text{norm}}^k(T) - \frac{1}{n} \sum_{i=1}^n S_{\text{norm}}^k(t_i) \right) = \frac{1}{2} \underbrace{\sum_{k=0}^{2^N-1} f(x_k)}_{=1} + \frac{1}{2M} \sum_{k=0}^{2^N-1} f(x_k) g(x_k), \tag{17}$$

where  $N$  is the number of qubits in  $|x_k\rangle$ , and for the last equality, we used Equation (13). We may now retrieve  $g(x_k)$  by the appropriate re-scaling.

Let  $V$  be another operator implementing some amplitude  $v(x_k)$ . We want to multiply  $v(x_k)$  with the previously constructed  $g(x_k)$  by, for example, employing a Toffoli gate. This introduces an error term of  $v(x_k)/2$ . We can subtract this error term at the cost of another factor of  $1/2$  by employing an additional amplitude subtraction, which accordingly yields

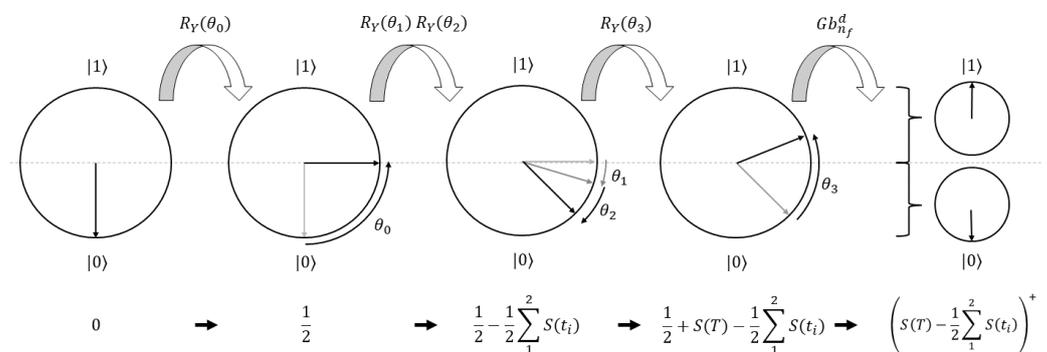
$$P(V |0\rangle_v G |0\rangle_g = |1\rangle_z |1\rangle_g) = \frac{1}{2} + \frac{1}{4M} \sum_{k=0}^{2^N-1} v(x_k) g(x_k). \tag{18}$$

We will use this method to combine  $g(x_k)$  with the positive part  $q(x_k)$  introduced in the next part.

#### 4.4. Encoding the Positive Part

Remember that all the constructed values are encoded in probabilities and not in qubit states. Because of this, accessing them is difficult and generally completed using measurements. This complicates a possible implementation of the positive part. Nevertheless, [Koppe and Wolf \(2022\)](#) showed how to construct a quantum circuit that implements the positive part depending on some input angle  $\theta$  of an  $\mathcal{R}_Y$ -gate. We may employ this technique by carefully encoding the needed option payoff as the final angle of a series of rotations. A visual representation of our encoding and how this constructs the needed input angle is shown in Figure 3. To be more precise, we construct a circuit  $Gb^D(\theta)$  with approximation level  $D$  implementing

$$P(Gb^D(\theta) |0^{\otimes 2^D-1}\rangle |0\rangle = |0^{\otimes 2^D-1}\rangle |1\rangle) = \frac{\sin^{2^{D+1}}(\theta)}{\sin^{2^{D+1}}(\theta) + \cos^{2^{D+1}}(\theta)} \approx \begin{cases} 0 & \text{if } \theta \leq \frac{\pi}{4}, \\ 1 & \text{if } \theta > \frac{\pi}{4}. \end{cases} \quad (19)$$



**Figure 3.** Representation of the quantum algorithm implementing the positive part of the value at maturity for  $n = 2$ . As the  $\mathcal{R}_Y$ -gate only rotates about the  $Y$ -axis, we can illustrate the employed rotations by showing the relevant 2-dimensional Bloch sphere of the used qubit. The used rotations are shown above, and the corresponding values are shown below. In the last step, the employed gearbox circuit sets a qubit in state  $|1\rangle$  for angles above  $\pi/4$  and in state  $|0\rangle$  for angles below  $\pi/4$ , ultimately implementing the needed function  $q(x_k)$  as shown in Equation (22).

The right-hand side of Equation (19) contains the limit value of a Fourier series

$$a_0 + \sum_{j=1}^{n_f} a_j \cos(2j\theta) \xrightarrow{n_f \rightarrow \infty} \frac{1}{2^{2^D-1} (\sin^{2^{D+1}}(\theta) + \cos^{2^{D+1}}(\theta))}. \quad (20)$$

For the evaluation of the probability, the order of the Fourier expansion  $n_f$  has to be decided for being able to implement a simultaneous computation. Due to the periodicity of the right-hand side, only cosine terms are needed. We denote the resulting circuit we employ by  $Gb_{n_f}^D(\theta) \xrightarrow{n_f \rightarrow \infty} Gb^D(\theta)$ . As the circuit  $Gb_{n_f}^D(\theta)$  employs  $\mathcal{R}_Y$ -gates, we may use the fact that  $\mathcal{R}_Y(\theta_1)\mathcal{R}_Y(\theta_2) = \mathcal{R}_Y(\theta_1 + \theta_2)$  to implement the angle  $\theta$  used for Equation (19) as a sum of individual angles  $\theta_i$  with

$$\theta_0 = \pi/4, \quad \theta_i = -\frac{1}{n} \tilde{S}_{\text{norm}}^k(t_i), i = 1, \dots, n, \quad \theta_{n+1} = \tilde{S}_{\text{norm}}^k(T), \quad (21)$$

where  $\tilde{S}_{\text{norm}}^k(t_i)$ , with  $|\tilde{S}_{\text{norm}}^k(T) - \sum_{i=1}^n \tilde{S}_{\text{norm}}^k(t_i)| < \pi/4$ , being a scaled-down version of  $S^k(t_i)$ . In other words, each asset value now corresponds to a rotation resulting in

$$\begin{aligned}
 &P\left(Gb_{n_f}^D\left(\sum_{i=0}^{n+1}\theta_i\right)|0^{\otimes 2^D-1}\rangle|0\rangle=|0^{\otimes 2^D-1}\rangle|1\rangle\right) \\
 &\approx \frac{1}{2^{2^D-1}}q(x_k)=\frac{1}{2^{2^D-1}}\cdot\begin{cases} 0 & \text{if } S(T)-\frac{1}{n}\sum_{i=1}^n S(t_i)\leq 0, \\ 1 & \text{if } S(T)-\frac{1}{n}\sum_{i=1}^n S(t_i)> 0. \end{cases}
 \end{aligned}
 \tag{22}$$

4.5. Concatenate Circuits and Resulting Computation

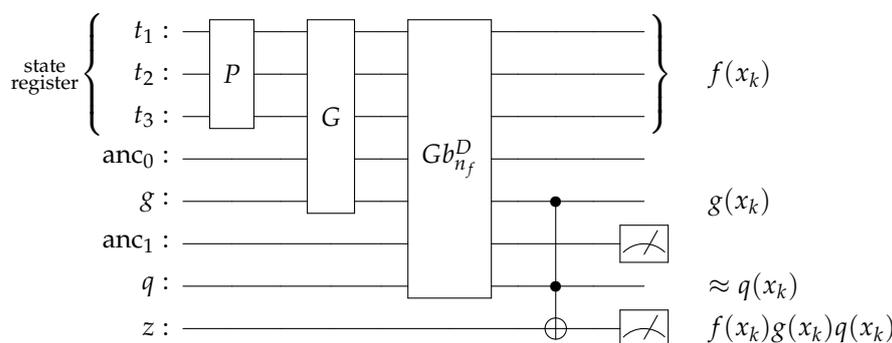
Let us now concatenate the above circuits as a new circuit Z combining their respective target qubits using a multi-controlled X-gate on a new target qubit we denote as  $|\cdot\rangle_z$ . The resulting probability is then given as

$$P(Z|0\rangle_z=|1\rangle_z)\approx\left(\frac{1}{2}+\frac{1}{2^{2^D-1}4M}\sum_{k=0}^{2^N-1}f(x_k)g(x_k)q(x_k)\right).
 \tag{23}$$

When executing the quantum circuit Z multiple times and counting the number of times state  $|1\rangle_z$  was measured, we obtain an estimator  $\hat{P}$  for the probability  $P(Z|0\rangle_z=|1\rangle_z)$  and may retrieve an estimator for the fair option price by rescaling the result as

$$2^{2^D-1}4M\left(\hat{P}-\frac{1}{2}\right)\approx\mathbf{E}\left[\left(S(T)-\frac{1}{n}\sum_{i=1}^n S(t_i)\right)^+\right].
 \tag{24}$$

The complete quantum circuit is presented in Figure 4.



**Figure 4.** A visual representation of the quantum circuit to price a floating-strike Asian option for  $n = 3$ . For  $|\cdot\rangle_g$  and  $|\cdot\rangle_q$ , the probability of measuring them in state  $|1\rangle$ , given the state register is in state  $|x_k\rangle$ , is shown at the right-hand side. We abbreviated the ancillary qubit registers needed for the circuit parts G and  $Gb_{n_f}^D$  with  $anc_0$  and  $anc_1$ , respectively. Note that, as described in Equation (22), the  $q(x_k)$  is approximated.

4.6. Amplitude Estimation and Quantum Speed-Up

The quantum speed-up for computing expectation values derives from the use of quantum amplitude estimation (AE) introduced by Brassard et al. (2000). Given an operator Z implementing the transformation

$$Z|0^{\otimes N}\rangle_a|0\rangle_z=\sqrt{1-z}|\phi_0\rangle_a|0\rangle_z+\sqrt{z}|\phi_1\rangle_a|1\rangle_z,
 \tag{25}$$

AE allows us to estimate the probability of the state  $|1\rangle_z$  being measured with a convergence speed of  $\mathcal{O}(1/N_s)$ , where  $N_s$  is the number of circuit executions. Montanaro (2015) presented that if this amplitude is carefully constructed to represent an expectation value, this estimator converges faster than classical Monte Carlo methods converging with  $\mathcal{O}(1/\sqrt{N_s})$ . By construction, our quantum circuit is compatible with AE; therefore, a quantum speed-up could be achieved. Further information on AE for derivative pricing can be found in Chakrabarti et al. (2021).

#### 4.7. Dividing the Circuit into Subcircuits to Reduce Depth

It is possible to divide our circuit into smaller subcircuits and recombine their results to be able to omit the scaling factor of 1/4 introduced by the amplitude arithmetics and to reduce depth, i.e., the number of gates employed in a single quantum circuit. This can be helpful as current quantum devices have a high rate of error per two-qubit gate. One such division can be made for  $g(x_k)$ . Let us define two new functions by

$$g(x_k) = S^k(T) - \frac{1}{n} \sum_{i=1}^n S^k(t_i) =: g_+(x_k) - g_-(x_k). \tag{26}$$

We construct new circuits where instead of  $g$  we use  $g_{\pm}$ , resulting in circuits  $Z_{\pm}$  no longer employing the amplitude subtraction. We may subtract their results on a classical computer to obtain

$$P(Z_+ |0\rangle_z = |1\rangle_z) - P(Z_- |0\rangle_z = |1\rangle_z) \doteq \frac{1}{2^{2^D-1} M} \mathbf{E} \left[ \left( S(T) - \frac{1}{n} \sum_{i=1}^n S(t_i) \right)^+ \right]. \tag{27}$$

Moreover, we may split the subcircuit implementing the Fourier approximation, where as a result, each term of the series corresponds to a new subcircuit. As this is a technical detail, we already assumed this for the computations performed above.

### 5. Results

#### Application to Binomial Model

We will apply the presented algorithm to the [Rendleman and Bartter \(1979\)](#) binomial model, which is a binomial model with equal probabilities for up and down jumps, and which satisfies the moment conditions for the application of Donsker’s theorem (c.f. [Korn et al. 2010](#), Theorem 4.17). Due to current hardware restrictions, we evaluate the algorithm on a quantum computer simulator. We employ the previous notation for our Asian option and choose time steps of length  $\Delta t = T/n$  with maturity set to  $T = 1$ . Our underlying starts at  $S(0) = 1$ , and for each time  $t_i = i \Delta t, i = 1, \dots, n$ , we assume our underlying to evolve under the risk-neutral measure  $\mathbb{Q}$  with

$$S(t_{i+1}) = \begin{cases} uS(t_i) & \text{with probability } 1/2, \\ dS(t_i) & \text{with probability } 1/2, \end{cases} \tag{28}$$

where

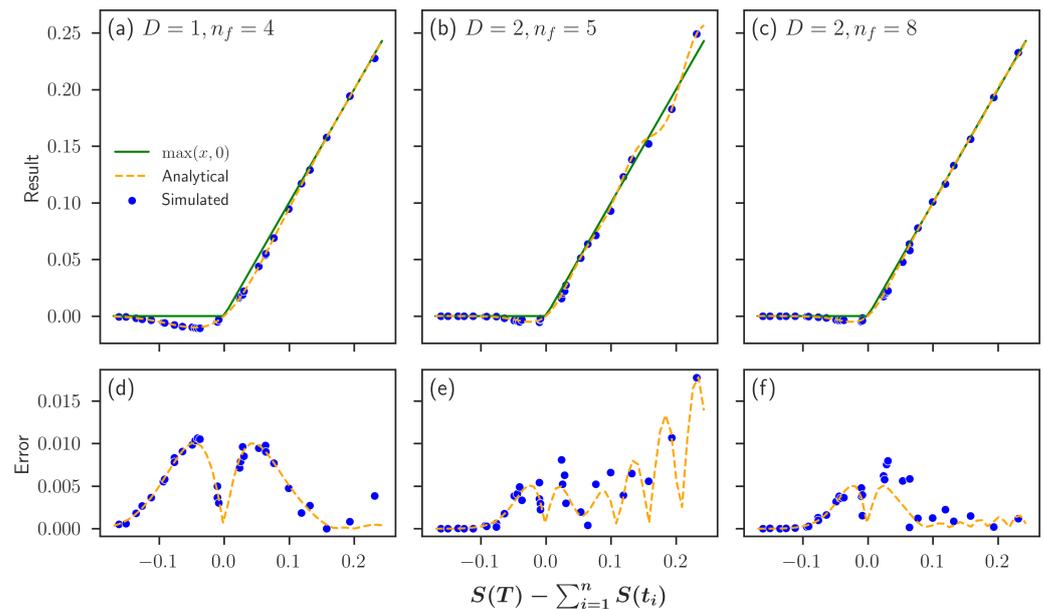
$$u = e^{(r-\sigma^2/2)\Delta t + \sigma\sqrt{\Delta t}}, \quad d = e^{(r-\sigma^2/2)\Delta t - \sigma\sqrt{\Delta t}}. \tag{29}$$

Each  $t_i$  used in the valuation tree will be included in our Asian option. As in (2), we assume a zero interest, i.e.,  $r = 0$ . We choose  $\sigma = 0.2$ . We simulated our quantum algorithm for  $n = 5$  time steps, approximation levels  $D = 1, 2$  for the indicator function to calculate the positive part, and we varied the number of terms  $n_f$  used in the Fourier expansion.

The most important decision when employing our algorithm is the qubit mapping of the valuation tree to the state register. In this case, we aimed for minimal qubit depth, introducing one qubit for each time step. Measuring this qubit in state  $|0\rangle$  or  $|1\rangle$ , respectively, corresponds to the upper or lower path taken. The ensuing state register consists of  $n$  qubits. Setting the state register in a uniform superposition with the help the Hadamard gates encodes the probability of  $p = 1/2$  directly, omitting the need for an additional circuit implementing  $P$ . Each of the possible values of our underlying  $S_{k_i}(t_i)$  can be uniquely mapped to a series of up or down decisions, enabling a straightforward implementation of the conditional payoff with  $g(x_k)$  and  $q(x_k)$ . In our implementation, we are not able to employ amplitude estimation due to circuit depth restrictions of the simulator.

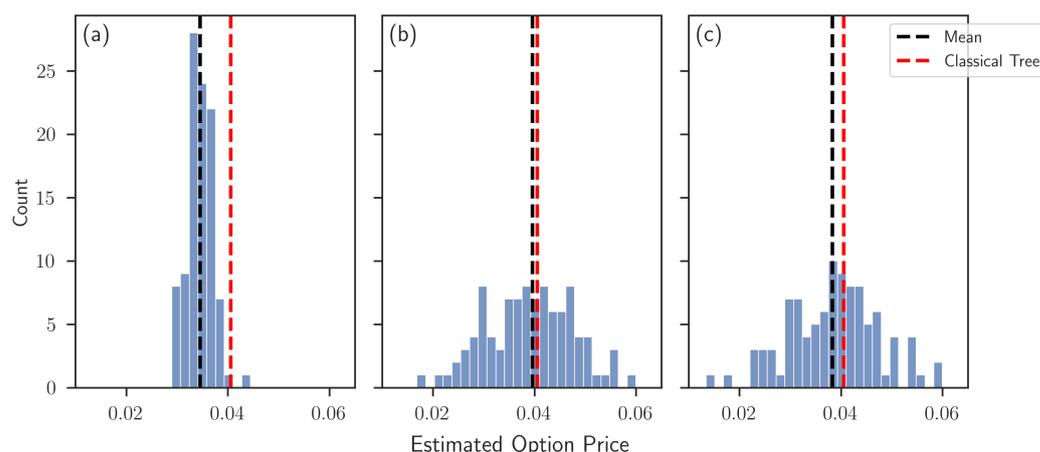
To begin with, we set our state register in each of its computational basis states corresponding to a single path taken to see the approximation for the corresponding

option value. In the upper row of Figure 5, the analytical probability as well as the circuit result for  $10^7$  shots are shown for  $n = 5$ . In the lower row, the corresponding errors are visualized. One circuit shot refers to one execution of the quantum circuit resulting in a single measurement. Observe that for approximation level  $D = 1$  of the positive part presented in Figure 5d, the error is comparatively large about 0, but it vanishes for outer values. As most evaluated values are about 0, decreasing the error about 0 is crucial. As expected, the error decreases significantly for approximation level  $D = 2$ . Nevertheless, the better approximation of the positive part  $q(x_k)$  also introduces the need for more terms  $n_f$  in the Fourier expansion, the influence of which can be seen when comparing Figure 5e,f.



**Figure 5.** Comparison of different approximation levels  $D$  and terms in the Fourier series  $n_f$  when evaluating an Asian option on a quantum computer simulator for different paths of the underlying valuation tree. In (a–c), we compare the circuit result (blue) for  $10^7$  executions to the analytical value (orange) we expect, and we try to approximate the option payoff (green). In (d–f), we present the respective absolute errors of the circuit and analytical results to the option payoff.

Subsequently, the state register  $|x_k\rangle$  is set to an equal superposition using Hadamard gates, enabling the simultaneous evaluation of each path. In Figure 6, results for the full pricing algorithm are shown for various parameter combinations. Each data point was generated by  $10^6$  circuit shots, and each histogram consists of 100 different data points. The respective mean in black (which consequently corresponds to a quantum circuit being executed  $10^8$  times) and the exact result of the classically evaluated binomial tree in red are depicted for each data set. Multiple observations can be made. In Figure 6a, approximation level  $D = 1$  results in a noticeable bias, c.f. Figure 5a. By increasing the approximation level to  $D = 2$  in Figure 6b,c, the bias is reduced. Figure 6b uses a Fourier expansion with  $n_f = 5$  terms, and as a result, the under-estimation for paths with payoff about 0 is leveled by outer payoffs, c.f. Figure 5b. Increasing the terms of the Fourier expansion to  $n_f = 8$  in Figure 6c decreases the outer errors as shown in Figure 5c. Consequently, the under-estimation for payoffs about 0 is no longer leveled, and the total bias increases again. Note that the over-estimation, hence, the smaller bias of Figure 5b, depends on the payoff range. When employing the algorithm for other valuation trees, additional under-estimations are possible.



**Figure 6.** Histograms of quantum simulator results for full pricing algorithm with state register in uniform superposition, i.e., simultaneous computation of all paths for  $n = 5$  time steps, gearbox approximation levels  $D$  and terms in the Fourier expansion  $n_f$ : (a)  $D = 1, n_f = 4$ , (b)  $D = 2, n_f = 5$ , and (c)  $D = 2, n_f = 8$ . 100 circuit results are presented, where each result consists of  $10^6$  circuit executions.

In Table 1, we present the  $cX$ -gate count and qubit amount needed for each of the three used parameter combinations, which are currently the main criteria for a quantum circuit to be implementable on real NISQ hardware. Current state-of-the-art quantum hardware can execute circuits with reliable results with up to 30  $cX$ -gate applications. Note that in our implementation, we split the main circuit into multiple subcircuits, each implementing a term for the Fourier series approximation and  $g_+(x_k)$  or  $g_-(x_k)$ , respectively. Because of this, the  $cX$ -gate count is the same for variants (b) and (c). The metric is dominated by the uniformly controlled rotations each requiring  $\mathcal{O}(2^n)$   $cX$ -gate applications as proven in Shende et al. (2006). The linear scaling in the amount of qubits needed can also be seen, increasing only in the qubits needed for the path encoding and the  $\lceil \log_2(n) \rceil$  ancillary qubits needed for the implementation of  $G$ .

**Table 1.** Required  $cX$ -gate counts and circuit width (number of used qubits) for a single subcircuit and number of subcircuits used in the presented algorithm with parameters (a)  $D = 1, n_f = 4$ , (b)  $D = 2, n_f = 5$  and (c)  $D = 2, n_f = 8$ , for a varying number of time steps  $n$ .

Parameters		$n$					Number of Subcircuits
		2	3	4	5	6	
(a)	$cX$	67	141	243	473	847	8
	Width	13	15	16	18	19	
(b)	$cX$	89	187	337	663	1229	10
	Width	17	19	20	22	23	
(c)	$cX$	89	187	337	663	1229	16
	Width	17	19	20	22	23	

## 6. Discussion

We developed a novel quantum circuit to efficiently price discrete floating-strike Asian options using quantum computing. The implementation uses a combination of amplitude-based arithmetics and the application of an amplitude-based unit step function to efficiently construct the fair price of the option as the amplitude of a quantum state. Additional quantum algorithms may be employed, resulting in the possibility of a quantum speed-up if larger, fault-tolerant quantum hardware becomes available in the future. This research further presents the applicability of quantum computing in the area of finance, risk management and insurance, and it should increase awareness of the upcoming changes

in computational methods provided by quantum technology. The algorithm has been simulated using the Qiskit simulator by Sajid et al. (2021), showing that the expected results can be produced. The method may be generalized to use for other derivative pricing algorithms that are sensitive to discretization errors. Depending on the implementation of the valuation tree, a comparatively small amount of qubits suffices for the construction. As there are quantum computers being developed with limited qubit availability (c.f. Pezzagna and Meijer 2021), this might prove to be a major advantage.

Subject to further research could be the application to different types of options such as the cliquet option (c.f. Bernard and Li 2013) used for example in life insurance contracts. As they are financed by surpluses of the general account, they allow both participation at the capital market and investing the premiums rather safely in the general account. Alternatively, the employment of different quantum circuits for the more optimized implementation of the various functions is of interest. A different application of our technique would be the combination of the presented approach of using an amplitude-based positive part with established quantum algorithms utilizing a bit-based encoding. This could reduce circuit depth and qubit amounts, bringing us one step further to quantum advantage.

**Author Contributions:** Conceptualization, M.-O.W.; methodology, M.-O.W. and J.K.; validation, M.-O.W.; formal analysis, M.-O.W. and R.H.; data curation, M.-O.W. and J.K.; writing, M.-O.W. and R.H. and J.K.; visualization, M.-O.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the project AnQuC-3 of the Competence Center Quantum Computing Rhineland-Palatinate (Germany).

**Data Availability Statement:** The Python code used for data generation is available upon request.

**Acknowledgments:** We would like to thank Ralf Korn for general advice and constructive discussions regarding financial mathematics. We further acknowledge Luminita Mihaila for fruitful discussions on quantum computing.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Bernard, Carole, and Wenbo V. Li. 2013. Pricing and hedging of cliquet options and locally capped contracts. *SIAM Journal on Financial Mathematics* 4: 353–71. [CrossRef]
- Blank, Carsten, Daniel K. Park, and Francesco Petruccione. 2021. Quantum-enhanced analysis of discrete stochastic processes. *NPJ Quantum Information* 7: 1–9. [CrossRef]
- Brassard, Gilles, Peter Hoyer, Michele Mosca, and Alain Tapp. 2000. Quantum amplitude amplification and estimation. *AMS Contemporary Mathematics Series* 305: 53–68. [CrossRef]
- Chakrabarti, Shouvanik, Rajiv Krishnakumar, Guglielmo Mazzola, Nikitas Stamatopoulos, Stefan Woerner, and William J. Zeng. 2021. A Threshold for Quantum Advantage in Derivative Pricing. *Quantum* 5: 463. [CrossRef]
- Cox, John C., Stephen A. Ross, and Mark Rubinstein. 1979. Option pricing: A simplified approach. *Journal of Financial Economics* 7: 229–63. [CrossRef]
- Egger, Daniel J., Jakub Mareček, and Stefan Woerner. 2021. Warm-starting quantum optimization. *Quantum* 5: 479. [CrossRef]
- Grover, Lov, and Terry Rudolph. 2002. Creating Superpositions that Correspond to Efficiently Integrable Probability Distributions. Available online: <https://arxiv.org/abs/quant-ph/0208112> (accessed on 13 November 2022).
- Holmes, Adam, and Anne Y. Matsuura. 2020. Efficient quantum circuits for accurate state preparation of smooth, differentiable functions. Paper presented at the 2020 IEEE International Conference on Quantum Computing and Engineering (QCE), Denver, CO, USA, October 12–16. pp. 169–79. [CrossRef]
- Koppe, Jonas, and Mark-Oliver Wolf. 2022. An amplitude-based implementation of the unit step function on a quantum computer. *arXiv* arXiv:2206.03053. <https://doi.org/10.48550/ARXIV.2206.03053>.
- Korn, Ralf, Elke Korn, and Gerald Kroisandt. 2010. *Monte Carlo Methods and Models in Finance and Insurance*. Boca Raton: CRC Press.
- Kubo, Kenji, Koichi Miyamoto, Kosuke Mitarai, and Keisuke Fujii. 2022. Pricing multi-asset derivatives by variational quantum algorithms. *arXiv* arXiv:2207.01277.
- Martin, Ana, Bruno Candelas, Ángel Rodríguez-Rozas, José D. Martín-Guerrero, Xi Chen, Lucas Lamata, Román Orús, Enrique Solano, and Mikel Sanz. 2021. Toward pricing financial derivatives with an ibm quantum computer. *Physical Review Research* 3: 013167. [CrossRef]

- Miyamoto, Koichi, and Kenji Kubo. 2021. Pricing multi-asset derivatives by finite difference method on a quantum computer. *arXiv* arXiv:2109.12896.
- Montanaro, Ashley. 2015. Quantum speedup of monte carlo methods. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science* 471: 20150301. [[CrossRef](#)]
- Möttönen, Mikko, Juha J. Vartiainen, Ville Bergholm, and Martti M. Salomaa. 2004. Quantum circuits for general multiqubit gates. *Physical Review Letters* 93: 130502. [[CrossRef](#)]
- Pezzagna, Sébastien, and Jan Meijer. 2021. Quantum computer based on color centers in diamond. *Applied Physics Reviews* 8: 011308. [[CrossRef](#)]
- Plesch, Martin, and Časlav Brukner. 2011. Quantum-state preparation with universal gate decompositions. *Physical Review A* 83: 032302. [[CrossRef](#)]
- Preskill, John. 2018. Quantum Computing in the NISQ era and beyond. *Quantum* 2: 79. [[CrossRef](#)]
- Rebentrost, Patrick, Brajesh Gupt, and Thomas R. Bromley. 2018. Quantum computational finance: Monte Carlo pricing of financial derivatives. *Physical Review A* 98: 022321. [[CrossRef](#)]
- Rendleman, Richard J., and Brit J. Bartter. 1979. Two-state option pricing. *The Journal of Finance* 34: 1093–110. [[CrossRef](#)]
- Sajid, Anis, Abby Mitchell, Héctor Abraham, AduOffei, Rochisha Agarwal, Gabriele Agliardi, Merav Aharoni, Vishnu Ajith, and Ismail Yunus Akhalwaya. 2021. Qiskit: An open-source framework for quantum computing. [[CrossRef](#)]
- Shende, Vivek V., Stephen S. Bullock, and Igor L. Markov. 2006. Synthesis of quantum-logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 25: 1000–10. [[CrossRef](#)]
- Stamatopoulos, Nikitas, Daniel J. Egger, Yue Sun, Christa Zoufal, Raban Iten, Ning Shen, and Stefan Woerner. 2020. Option pricing using quantum computers. *Quantum* 4: 291. [[CrossRef](#)]
- Stamatopoulos, Nikitas, Guglielmo Mazzola, Stefan Woerner, and William J. Zeng. 2022. Towards quantum advantage in financial market risk using quantum gradient algorithms. *Quantum* 6: 770. [[CrossRef](#)]
- Vazquez, Almudena Carrera, and Stefan Woerner. 2021. Efficient state preparation for quantum amplitude estimation. *Physical Review Applied* 15: 034027. [[CrossRef](#)]
- Woerner, Stefan, and Daniel J. Egger. 2019. Quantum risk analysis. *NPJ Quantum Information* 5: 15. [[CrossRef](#)]
- Zhang, Xiao-Ming, Man-Hong Yung, and Xiao Yuan. 2021. Low-depth quantum state preparation. *Physical Review Research* 3: 043200. [[CrossRef](#)]
- Zoufal, Christa, Aurélien Lucchi, and Stefan Woerner. 2019. Quantum generative adversarial networks for learning and loading random distributions. *NPJ Quantum Information* 5: 103. [[CrossRef](#)]